V.S.B ENGINEERING COLLEGE, KARUR – 639 111

DEPARTMENT OF INFORMATION TECHNOLOGY

CLASS / SEMESTER: IV YEAR/VII SEMESTER

QUESTION BANK

Subject Name: INFORMATION MANAGEMENT
Year/Sem: IV/VII

UNIT I

PART-A

1. What is a data model? List the types of data model used.

A database model is the theoretical foundation of a database and fundamentally determines in which manner data can be stored, organized, and manipulated in a database system.

It thereby defines the infrastructure offered by a particular database system. The most popular example of a database model is the relational model. Types of data model used

- Hierarchical model
- Network model
- Relational model
- Entity-relationship
- Object-relational model
- Object model

2. Give the levels of data abstraction?
   - Physical level
   - Logical level
   - View level
3. Distinguish between Hadoop 1.x and Hadoop 2.x
   - In Hadoop 1.x, Map Reduce is responsible for both processing and cluster management whereas in Hadoop 2.x processing is taken care of by other processing models and YARN is responsible for cluster management.
   - Hadoop 2.x scales better when compared to Hadoop 1.x with close to 10000 nodes per cluster.
   - Hadoop 1.x has single point of failure problem and whenever the Name Node fails it has to be recovered manually. However, in case of Hadoop 2.x Stand By Name Node overcomes the problem and whenever the Name Node fails it is configured for automatic recovery
4. Define data model?

A data model is a collection of conceptual tools for describing data, data relationships, data semantics and consistency constraints.

5. What is an entity relationship model?

The entity relationship model is a collection of basic objects called entities and relationship among those objects. An entity is a thing or object in the real world that is distinguishable from other objects.

6. What are attributes and relationship? Give examples.
   - An entity is represented by a set of attributes.
   - Attributes are descriptive properties possessed by each member of an entity set.
   - Example: possible attributes of customer entity are customer name, customer id, Customer Street, customer city.
   - A relationship is an association among several entities.
   - Example: A depositor relationship associates a customer with each account that he/she has.
7. Distinguish between HBase and Hive. **Nov/Dec 2016**

- HBase and Hive both are completely different Hadoop based technologies-
- Hive is a data warehouse infrastructure on top of Hadoop, whereas HBase is a NoSQL key value store that runs on top of Hadoop.
- Hive helps SQL savvy people to run Map Reduce jobs whereas HBase supports 4 primary operations put, get, scan and delete.
- HBase is ideal for real time querying of big data where Hive is an ideal choice for analytical

querying of data collected over period of time.

8. What is meant by normalization of data?
   It is a process of analyzing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties
   - Minimizing redundancy
   - Minimizing insertion
   - Deletion and updating anomalies
9. Define - Entity set and Relationship set.
   - Entity set: The set of all entities of the same type is termed as an entity set.
   - Relationship set: The set of all relationships of the same type is termed as a relationship set.
10. What are stored, derived, composite attributes?
    - Stored attributes: The attributes stored in a data base are called stored attributes.
    - Derived attributes: The attributes that are derived from the stored attributes are called derived attributes.
    - For example: The Age attribute derived from DOB attribute.
11. Define - null values.
    In some cases a particular entity may not have an applicable value for an attribute or if we do not know the value of an attribute for a particular entity. In these cases null value is used.
12. What is meant by the degree of relationship set?
    The degree of relationship type is the number of participating entity types.
13. Define - Weak and Strong Entity Sets
    - Weak entity set: entity set that do not have key attribute of their own are called weak entity sets.
    - Strong entity set: Entity set that has a primary key is termed a strong entity set.
14. What does the cardinality ratio specify?
    - Mapping cardinalities or cardinality ratios express the number of entities to which another entity can be associated.
    - Mapping cardinalities must be one of the following:
      • One to one
      • One to many
      • Many to one
      • Many to many
15. What is partitioning, shuffle and sort phase.
    Shuffle Phase: Once the first map tasks are completed, the nodes continue to perform several other map tasks and also exchange the intermediate outputs with the reducers as required. This process of moving the intermediate outputs of map tasks to the reducer is referred to as Shuffling.
    Sort Phase: Hadoop MapReduce automatically sorts the set of intermediate keys on a single node before they are given as input to the reducer.
    Partitioning Phase: The process that determines which intermediate keys and value will be received by each reducer instance is referred to as partitioning. The destination partition is same for any key irrespective of the mapper instance that generated it.
16. What is a candidate key and primary key? **Nov/Dec 2016**
    - Minimal super keys are called candidate keys.
    - Primary key is chosen by the database designer as the principal means of identifying an entity in the entity set.
17. What are the main components of a Hadoop Application?
    Hadoop applications have wide range of technologies that provide great advantage in solving complex business problems.

Core components of a Hadoop application are-

- Hadoop Common
- HDFS
- Hadoop MapReduce
- YARN
- Data Access Components are - Pig and Hive
- Data Storage Component is – Hbase
- Data Integration Components are - Apache Flume, Sqoop.
- Data Management and Monitoring Components are - Ambari, Oozie and Zookeeper.
- Data Serialization Components are - Thrift and Avr
- Data Intelligence Components are - Apache Mahout and Drill.

18. What is JDBC? List of JDBC drivers.

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is part of the Java Standard Edition platform, from Oracle Corporation.

- Type 1 - JDBC-ODBC Bridge Driver.
- Type 2 - Java Native Driver.
- Type 3 - Java Network Protocol Driver.
- Type 4 - Pure Java Driver.

19. What are three classes of statements using to execute queries in java?

- Statement
- Prepared Statement
- Callable Statement

20. What is stored procedure? **April/May 2017**

- In a database management system (DBM), a stored procedure is a set of Structured Query Language (SQL) statements with an assigned name that's stored in the database in compiled form so that it can be shared by a number of programs.
- The use of stored procedures can be helpful in controlling access to data, preserving data integrity and improving productivity.

21. What concept the Hadoop framework works? **April/May 2017** Hadoop Framework works on the following two core components-

- HDFS – Hadoop Distributed File System: It is the java based file system for scalable and reliable storage of large datasets. Data in HDFS is stored in the form of blocks and it operates on the Master Slave Architecture.
- Hadoop MapReduce: This is a java based programming paradigm of Hadoop framework that provides scalability across various Hadoop clusters.

PART B

1. Describe the importance of database modelling. Illustrate with a simple case study. **April/May 2017**

2. i) Explain the use of Java Data Base Connectivity(JDBC)

ii) Elaborate on the function of Map Reduce Framework **April/May 2017** 3. Explain how to transform enhanced ER diagrams into relations with suitable example **Nov/Dec2016** 4. With a neat diagram, explain the architecture of HDFS **Nov/Dec2016** 5. Explain Hadoop Eco systems.

PART B

1. Describe the importance of database modelling. Illustrate with a simple case study. **April/May 2017**

- Database
- Database modelling
- ER Model
  - Entity set
  - Attributes
- Types of Attributes
  i. Composite versus Simple Attribute
  ii. Single valued versus multivalued attribute
  iii. Stored versus derived attribute s
  iv. Null values
  v. Complex attributes
- Relationship between the entities
- Cardinality ratio
- Participation constraint
  i. Total/mandatory participation
  ii. Partial/optional participation
- ER to Relational Data Model
  - The Mapping
  - Type
  - Diagram
  - DDL
  - DML
  - Commands

Normalization

a. First Normal form
b. Second Normal form
c. Third Normal form
d. Boyce-codd Normal form

- Case Study-Library Management System

2. i) Explain the use of Java Data Base Connectivity(JDBC) **April/May 2017**

-JDBC API

-JDBC DRIVERS

-Types of JDBC Drivers

1.JDBC-ODBC Bridge Drivers

2.Java Native Drivers

3.Java Network Protocol Drivers

4.Pure Java Drivers

ii) Elaborate on the function of Map Reduce Framework **April/May 2017**

Map Reduce
-Reading the Data into MapReduce Program
-MapReduce Flow
-Mapping of physical and Logical division of data
-Diagram of MapReduce Flow
-Sample Working of the word count program

3. Explain how to transform enhanced ER diagrams into relations with suitable example

Nov/Dec2016 -Define ER Diagram
-entity
-Attribute
-Diagram
-ER Diagram to Relational Table

4. With a neat diagram, explain the architecture of HDFS **Nov/Dec2016** -Define HDFS

-HDFS Architecture
-Rack Awareness in HDFS
-Preparing HDFS Writes
-Reading Data from HDFS

5. Explain Hadoop Eco

systems. -Define Hadoop
-High level Hadoop Architecture
- High level Hadoop 1.xArchitecture
- High level Hadoop 1.xArchitecture
-Characteristics of Hadoop -
Components of Hadoop Echo system
-Diagram Hadoop ecosystem

UNIT – II

2 Mark questions and answers:

1. **What is firewall? (April / May – 2017)** A firewall is a device that filters all traffic between a protected network (internal network) and a less trustworthy (external network) network. It can be designed to operate as a filter at the level of IP packets.

2. **What are the goals of privacy laws in data protection? (April / May – 2017)**
   Information privacy or data protection laws prohibit the disclosure or misuse of information held by private individuals.

3. **What is session hijacking? (Nov / Dec – 2016)**
   Session hijacking is a process of stealing another user's identity and masquerading as a legitimate user. Every web server maintains a session with the help of cookie. Cookie is a piece of text send by the server to the client used for authentication and session maintenance. These cookies are stored at the client, attacker try to obtain these cookies, masquerading itself as a different person.

4. **Give an example for pseudonymity. (Nov / Dec – 2016)**
   Pseudonymity takes the most identifying fields within a database and replaces them with artificial identifiers, or pseudonyms. For example a name is replaced with a unique number. The purpose is to render the data record less identifying and therefore reduce concerns with data sharing and data retention.

5. **List the different types of virus.**
   A few types of virus are listed below:
   i) Transient virus ii) Resident virus iii) Document virus
   iv) Boot sector virus v) Macro virus vi) Polymorphic virus

6. **Write a note on the prevention of virus infection.**
   1. Use only commercial software acquired from reliable, well-established vendors.
   2. Test all new software on an isolated computer.
   3. Open attachments only when you know they are safer.
   4. Make a recoverable system image and store it safely.
   5. Make and retain backup copies of executable system files.
   6. Use virus detectors or scanners regularly and update them daily.

7. **What are the various non-malicious code?**
   Non-malicious code are errors due to human mistakes that go unnoticed while coding and do not cause severe damage to the system.
   i) Buffer overflows ii) Incomplete Mediation iii) Time-of-check to Time-of-Use errors.

8. **What are the control measures that can be used against threats?**
   Control measures that can be used against threats are
   i) Developmental control ii) Operating System Control iii) Administrative Control

9. **What is single sign-on? Give guidelines for selecting a password.**
   Single sign-on is a user authenticates once per session, and the system forwards that authenticated identity to all other processes that would require authentication.
   Guidelines:
   - Use character other than just A-Z.
   - Choose long password.
   - Avoid actual names.
   - Choose an unlikely password.

10. **What are the various memory and address protection scheme? Explain in brief.**
    Memory and address protection mechanisms can be built into the hardware that control the use of memory.

11. **What are the various security methods of operating system?**
    The various security methods of OS are separation, memory and address protection, control of access to general objects file protection mechanism, user authentication.

12. **What do you mean by stealth mode?**
    The stealth mode of IDS is to remain hidden in the network. In this mode IDS use two interfaces, one interface is used to monitor the network and the other is to raise alarm if any malicious found.

13. What is compliance?

Compliance is a snapshot of how your security program meets a specific set of security requirements at a given moment in time.

14. What is data privacy?

Data privacy, also called information **privacy**, is the aspect of information technology (IT) that deals with the ability an organization or individual has to determine what **data** in a computer system can be shared with third parties.

15. What is IDS?

An Intrusion Detection System is a device that keeps a check on the activities in a real time system and raises an alarm when it detects any malicious activity.

16. What are the advantages and disadvantages of IDS? Advantages:

i) It can detect the increasing number of threats by adding signature of the novel attack into the IDS.

ii) The cost of development of an IDS has significantly reduced.

i) The attacker tries to locate the IDS to disable it.

17. What is Denial of Service?

A DoS attack is an attempt to disrupt the services offered to legitimate users by rendering the computer resources as unavailable. DoS attack modes are

i) Resource exhaustion

ii) Modification or destruction of configuration files.

18. What is malicious code?

Malicious codes are generally imbibed into the program, with an intention of either modifying the contents or extracting the contents.

19. Define trapdoor.

Trapdoor is an undocumented entry point to a module. The main reason for the inclusion of trap doors by a developer is to test the module. Attackers can also use these trap door to maliciously access a program code.

20. What is network security?

Network Security is the process of taking physical and software preventative measures to protect the underlying **networking** infrastructure from unauthorized access, misuse, malfunction, modification, destruction, or improper disclosure, thereby creating a **secure** platform for computers, users and programs to perform.

16 mark questions:

1. Tabulate the types of malicious code with their characteristics. (April / May – 2017, Nov / Dec - 2016)
2. Explain the file protection mechanism in general purpose operating system. (April / May – 2017, Nov / Dec - 2016)
3. Outline the common components of Intrusion Detection Framework. Also compare the different types of IDS. (April / May – 2017)
4. Explain in detail about the various privacy principles and policies of data security. (Nov / Dec – 2016)
5. Explain in detail about firewall and the various types of firewall.

IT6701

Information Management

UNIT III

PART A

1. What are the key benefits of Master Data Management? (April/May 2017)

a) It provides a single version of truth.

b) It provides an increased consistency by reducing redundancy and data discrepancies.

c) It seperates master data from individual applications.

d) It improves operations and efficiency at low cost with increasing growth.

2. Define data governance ? (April/May 22017)

Data governance specifies the framework for decision rights and accountabilities to encourage a desirable behavior in the use of data. To promote a desirable, data governance develops and implements data policies, guidelines and standards that are consistent with the organization's mission, strategy, values, norms and culture.

3. List the advantages of client-side validation? (Nov/Dec 2016)

a) Allow for more interactivity by immediately responding to users' actions.

b) Execute quickly because they don't require a trip to the server.

c) May improve the usability of web sites for users whose browsers support scripts.

4. What is data scrubbing? (Nov/Dec 2016)

Data cleaning or Data scrubbing is the process of detecting and correcting corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data.

5. Define MDM?

Master Data Management (MDM) is a framework of processes and technologies aimed at creating and maintaining an authoritative, reliable, sustainable, accurate and secure data environment. It represent a "single and holistic version of the truth for master data and its relationships, and is an accepted benchmark used within an enterprise and across enterprises. It spans a diverse set of application system lines of business channels and user communities.

6. What are the types of MDM architectural dimensions?

There are three types of MDM architectural dimensions. There are

a) Design and deployment dimension
b) Use pattern dimension
c) Information scope or data domain dimension

7. What are the layers in MDM reference architecture?

There are five layers in MDM reference architecture. They are

a) Service abstraction layer
b) Data quality layer
c) Data rule layer
d) Data management layer
e) Business process layer

8. Define data quality management ?

The data quality management is a task for managing and maintaining good quality data by cleansing poor quality data using various tools that can be provided to different tools. The data cleaning tools are used to clean the poor quality data.

9. What are the different data quality management tools?

The different data quality management tools are

a) Data cleansing tool
b) Data parsing tools
c) Data profiling tools
d) Data matching tools
e) Data standardization tools
f) Data extract, transform and load (ETL) tools

10. Define data synchronization?

Data synchronization is an important task that has to be done in the master-slave environment. Data synchronization is a master-slave activity that needs to be done periodically when data contents at the master site changes as per business requirement. In MDM , the data hub is the master of some or all attributes of entities where synchronization flows from data hub towards other system components. The bidirectional synchronization in MDM data hub and legacy system are placed in a peer to peer relationship.

11. Define registry style?

The registry style of MDM data hub represents a registry of master entity identifiers that are created using identity attributes. The registry maintains identifying attributes. The identifying

attributes are used by entity resolution service to identify the master records. The data hub is responsible for creating and maintaining links with data source to obtain attributes.

12. What are the architectural styles in design and deployment dimension?

a) registry

b) external reference

c) reconciliation engine

d) transaction hub

13. Define collaborative master data management?

The collaborative MDM uses a process to create and maintain the master data associated with metadata. It allows users to author the master data objects. The collaborative process involves cleaning and updating operations to maintain the accurate master data.

14. What are the architectural implications of data domain or information scope dimension?

a) privacy and security concern put risk on the given data domain.

b) difficult to acquire and manage external references to entity.

c) complex design for entity resolution and identification.

15. Advantages of Sarbanes-Oxley Act?

a) reduction of financial statement fraud.

b) strengthening corporate governance.

c) reliability of financial information.

d) improving the liquidity.

e) model for private and non-profit companies.

16. Goals of data governance?

a) enable better decision making.

b) reduce operational friction.

c) train management and staffs to adopt common approaches to data issues.

d) reduce costs and increase effectiveness through coordination of efforts.

e) build standards, repeatable processes.

17. What are the three phases in data governance strategy?

a) initiate a data governance process

b) selection and implementation of data management and data delivery solutions

c) facilitate auditability and accountability

18. Define data rule layer?

The data rule layer includes key services driven by business defined rules for entity resolution, aggregation, synchronization, privacy and transformation. The different rules provided by this layer are synchronization rules, aggregation rules, visibility rules and transformation rules.

19. Define data quality layer?

Data quality layer is responsible for maintaining data quality using various services. The services of this layer are designed to validate the quality rules, resolve entity identification, and perform data standardization and reconciliation. The other services provided by this layer are data quality management, data transformation, guid management and data reporting.

20. Define IRM?

Risk management involves policies, procedures and practices for identifying, assessing, controlling, avoiding, minimizing and eliminating unacceptable risk. Most business entities adopt a common framework for risk management called Integrated Risk Management (IRM).

PART B

1. **Explain the enterprise architecture framework of MDM and highlight the key challenges of master data management. (April/May 2017)**
2. **Describe the functions of data governance framework and data synchronization. (April/May 2017)**
3. **Explain briefly about data quality management. (Nov/Dec 2016)**
4. **Explain privacy, regulatory requirements and compliance of master data management system in detail. (Nov/Dec 2016)**
5. **Explain MDM architectural dimensions.**

UNIT IV

1) List out the components of an information architecture system. (April/May 2017)

- organization systems
- Navigation systems
- Labeling systems
- Searching systems

2) Mention the use of labeling system in information architecture system. (April/May 2017)

The labeling system is used for representing thoughts and concepts on a website. The goal of labeling is to convey the meaningful information efficiently to the users without consuming much space. Labels are often used for representing organization and navigation systems.

3) What is information architecture? (Nov/Dec 2016)

A foundation discipline describing the theory, principles, guidelines, standards, conventions and factors for managing information as a resource.

The combination of organization, labeling and navigation schemes within an information system. The structural design of an information space to facilitate task completion and intuitive access to content.

4) Differences between library and web site. (Nov/Dec 2016)

A Web site is a related collection of World Wide Web (WWW) files that includes a beginning file called a home page. A company or an individual tells you how to get to their Web site by giving you the address of their home page. From the home page, you can get to all the other pages on their site.

A library is a collection of sources of information and similar resources, made accessible to a defined community for reference or borrowing.[1] It provides physical or digital access to material, and may be a physical building or room, or a virtual space, or both. .

5) What are the different phases of information architecture development?

- Research or analysis
- Strategic planning
- Conceptual design
- Implementation
- Administration

6) Define search system?

The search system is another important component of information architecture that allows an user to search for specific contents over a website. The search engine are the basic foundation of a search system. They are basically software applications running on web servers that perform search based on user queries.

7) What are the sources of labeling systems?

There are two sources in labeling system. They are

i) own site

ii) competitive site

8) What are the types of navigation system?

- Global navigation system
- Local navigation system
- Contextual navigation system
- Supplemental navigation system

9) What are the different types of label?

- Label as contextual links
- Label as heading
- Labels within navigation systems
- Iconic labels
- Labels as index terms

10) What are the main organization structure?

1. Hierarchial or top down structure
2. Database or bottom up structure
3. Hypertext structure

11) Write any four responsibilities of information architecture?

1. Collect information through various sources such as emails, focus groups
2. Organize huge amounts of information on large websites and intranets so that people can accurately find what they are looking for.
3. Understand user goals and needs.
4. Understand business and organization’s needs.

12) What are the three dimensions of information ecology?

The three dimensions of information ecology are

I. Content (includes content objective, volume of contents, documents, data types, governance and ownership)
II. Context (includes business goals, funding, policies, technology, constraints and resources)
III. Users (includes audience, task, needs, experience and information seeking behavior)

13) Define organization system.

The organization system is responsible for classifying the collected information in a correct manner for users. The information architect organizes the information such that people can search them easily, and they can find the right answer to their questions.

14) Define granularity of contents.

The granularity of contents is related to the organization contents at different levels. Various levels of granularities in information architecture include journals, articles, paragraphs and sentences. Granularity deals with articulating the contents hierarchically at different levels according to certain criteria.

15) What are the classification organization system? Two classification

I. Organization schemes
II. Organization structures

16) Define organization schemes.

Organization scheme are related to organizing the information in a correct manner by categorizing the contents and making relationship between each pieces. Organization schemes are mainly classified into three categories

I. Exact or objective organization schemes
II. Ambiguous or subjective organization schemes
III. Hybrid organization schemes

17) what are the classification of exact or objective organization schemes?

I. Alphabetical scheme
II. Chronological scheme
III. Geographical scheme

18) what are the classification of ambiguous or subjective organization schemes?

I. Topic scheme
II. Task scheme
III. Audience scheme
IV. Metaphor scheme

19) Define organization structure.

Organization structure plays an important role in designing websites. Iy helps architects to define relationships between pieces and content. A successful organization structure allows users to predict the information they want on a particular site.

20) What are advanced approaches related to navigation system?

- Personalization and customizations
- Visualization and social navigation

PART B

1) **Discuss the organization scheme used in information architecture system. (April/May 2017, Nov/Dec 2016)**
2) **Explain the role of navigation system in information architecture system. (April/May 2017, Nov/Dec 2016)**
3) **Explain the principles of information architecture.**
4) **Explain about labeling systems.**
5) **Discuss conceptual design and granularity of contents.**

UNIT V

1)What are the challenges in big data testing? (nov/dec 2016)

- Automation
- Virtualization
- Large dataset
- Testing across platforms
- Monitoring and diagnostic solution

2) What are the responsibilities of data administrator?

1. Designing the database
2. Security and authorization
3. Data availability and recovery from failures
4. Database tuning

3) Give some examples of sensitive data. (april/may 2017)

Sensitive information is defined as data that is protected against unwarranted disclosure. Access to sensitive information should be safeguarded.The most prevalent examples of sensitive information legislation include HIPAA, FERPA, and the NC Identity Theft Protect Act.

4) Write down the challenges with data administration. (april/may 2017)

- Creating the data repository
- Evolving nature of data consideration in analysis
- Enforcing the data policies and standards, especially those related to security.
- As the organization's needs are changing, efficient support should be provided to incorporate the changes and make provision for future scope

5) Define data retention policies.

Data retention policies are important legal regulations that deal with the complex issues of preserving business/commercial information for a pre-determined length of time on a pre-determined storage system. These policies define different retention periods, depending on the type of data. Along with the duration for maintaining the data, retention policies also describe the procedures for archiving the information, guidelines for destroying the information when the time limit exceeds and special mechanisms for handling the information when under litigation.

6) What are the three main objectives of data retention policy?

1. To main important records and documents for future use or reference.
2. To dispose of records or documents that are no longer needed.
3. To organize records so that they can be searched and accessed easily at a later date.

7) Define Internet Service Provider (ISP) License.

Internet Service Provider (ISP) licence permits restricted Internet Telephony for the ISP under the Internet Service License and is issued by the government under ISP guidelines. According to the ISP license, there are eight classes of records that the service providers should maintain for security purposes relating to customer information or transactions.

8) What are the three types of sensitive information

1. Personal information
2. Business information
3. Classified information

9) What is meant by handling of sensitive data?

Sensitive data needs to be handled with utmost care and with highest possible security measures. Given a dataset, one or more attribute values in the tuple/record can be sensitive and hence needs to be protected. But all the time, other attributes of the same tuple/record can be made available. Thus, the access policy needs to be defined at different granularity levels so that access of these values for the attributes can be made available.

10) Define access decisions.

The database administrator decides what data should be in the database and who should have access to it. These decisions are based on access policies that are defined in the organization. Multiple factors are considered in making these policies such as availability of data, acceptability of the access, authenticity of the user, etc.

11) What are the type of disclosures in sensitive data?

1. Displaying exact data
2. Displaying bounds
3. Displaying negative results

4. Displaying probable values

12) What are the stages in life cycle management costs.

1. Data creation
2. Backup storage against data loss
3. Archiving helps contain storage costs
4. Ensuring secure data destruction
5. Put secure IT asset disposition to work

13) Define data lifecycle management.

Data lifecycle management is the process of handling the flow of business information throughout its lifespan, from requirements through maintenance. The lifecycle crosses diverse applications, data stores and storage media. Data lifecycle management aims at automating data the processes involved in organizing data into separate tiers according to specified policies , and automating data migration from one tier to another based on those criteria.

14) What are the two-step process in testing of hadoop big data application.

1. Checking the functionality
2. Checking on the cluster

15) What are the challenges in testing big data application.

1. Automation
2. Virtualization
3. Large dataset
4. Testing across platforms
5. Monitoring and diagnostic solution

16) Why do enterprises have to retain information for long?

The enterprises retain or discover information for a number of reasons. The discovery could be coming from three angles. The first is for the business reason, to make information available from the operations perspective. The second is the compliance or legal aspect, where a certain legal case is field and you need to produce some piece of information in a court of law. The third requirement is with respect to storing personal information that depends on the needs of individuals.

17) What are the categories of protection and retention in documents?

1. Legal records 2) Final records 3) Permanent records
4) Accounting and corporate tax records 5) Workplace records 6) Employment, employee and payroll records
7) Bank records 8) Historic records 9) Temporary records

18) Define Unified Access Service License?

Unified Access Service License(UASL) was introduced by Do T through which an access provider can offer a fixed and/or mobile services using any technology under the same license.

19) Write the categories of records that the service providers are required in UASL?

1. Mobile numbers
2. Capture/Interception records
3. Site/Location
4. All call records

20) Define classified information.

Classified information pertains to a government body and is restricted according to the level of sensitivity(e.g., restricted, confidential, secret and top secret). Information is generally classified to protect security. Once the risk of harm has passed or decreased, classified information may be declassified and, possibly, made public.

Part B

1) **Explain the steps to test and deliver a big data application. (nov/dec 2016)**
2) **Explain the various phases of information lifecycle. (nov/dec 2016)**
3) **Outline the concept of data retention policies. (april/may 2017)**
4) **Write notes on information lifecycle management costs. (april/may 2017)**
5) **Write notes on**
    1. **confidential and sensitive data handling**
    2. **challenges with data administration**

Subject Code: CS6701

Subject Name: CRYPTOGRAPHY AND NETWORK SECURITY

Year/Sem: IV/VII

UNIT I
INTRODUCTION & NUMBER THEORY

1. Give the difference between active attack and passive attack.

Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: masquerade, replay, modification of messages, and denial of service. Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. Two types of passive attacks are release of message contents and traffic analysis.

2. Define the cryptanalysis and cryptography

Cryptology is the study of techniques for ensuring the secrecy and/or authenticity of information. The two main branches of cryptology are cryptography, which is the study of the design of such techniques; and cryptanalysis, which deals with the defeating such techniques, to recover information, or forging information that will be accepted as authentic.

3. Differentiate symmetric and asymmetric encryption.

Symmetric Encryption Asymmetric Encryption It is a form of cryptosystem in which encryption and decryption performed using the same key. It is a form of cryptosystem in which encryption and decryption performed using two keys. Eg: DES, AES Eg: RSA, ECC

4. What is brute force attack?

The attacker tries every possible key on a piece of cipher text until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.

5. Compare stream cipher with block cipher with example.

A stream cipher is one that encrypts a digital data stream one bit or one byte at a time. Examples of classical stream ciphers are the auto keyed Vigenère cipher and the Vernam cipher.

A block cipher is one in which a block of plaintext is treated as a whole and used to produce a cipher text block of equal length. Typically, a block size of 64 or 128 bits is used.

6. Compare Substitution and Transposition techniques.

A substitution techniques is one in which the letters of plaintext are replaced by other letter or by number or symbols. Eg: Caeser cipher. Transposition techniques means, different kind of mapping is achieved by performing some sort of permutation on the plaintext letters. Eg: DES, AES.

7. Define Diffusion & confusion.

Diffusion: It means each plaintext digits affect the value of many cipher text digits which is equivalent to each cipher text digit is affected by many plaintext digits. It can be achieved by performing permutation on the data. It is the relationship between the plaintext and cipher text.

Confusion: It can be achieved by substitution algorithm. It is the relationship between cipher text and key.

8. Explain Avalanche effect.

A desirable property of any encryption algorithm is that a small change in either the plaintext or the key produces a significant change in the cipher text. In particular, a change in one bit of the plaintext or one bit of the key should produce a change in many bits of the cipher text. If the change is small, this might provider a way to reduce the size of the plaintext or key space to be searched.

9. Give the five modes of operation of Block cipher.

- Electronic Codebook(ECB)
- Cipher Block Chaining(CBC)
- Cipher Feedback(CFB)
- Output Feedback(OFB)
- Counter(CTR)

10. Specify the four categories of security threats.

- Interruption
- Interception
- Modification
- Fabrication

11. Define integrity and non repudiation.

Integrity: Service that ensures that only authorized person able to modify the message.

Non repudiation: This service helps to prove that the person who denies the transaction is true or false.

12. Define cryptanalysis.

It is a process of attempting to discover the key or plaintext or both.

13. Define security mechanism.

It is process that is designed to detect prevent, recover from a security attack. Example: Encryption algorithm, Digital signature, Authentication protocols.

14. Define steganography.

Hiding the message into some cover media. It conceals the existence of a message.

15. Differentiate symmetric and asymmetric encryption?

Symmetric: It is a form of cryptosystem in which encryption and decryption performed using the same key. Eg: DES, AES

Asymmetric: It is a form of cryptosystem in which encryption and decryption performed using two keys. Eg: RSA, ECC

16. Specify the basic task for defining a security service.

A service that enhances the security of the data processing systems and the information transfer of an organization. The services are intended to counter security attack, and they make use of one or more security mechanism to provide the service.

17. Define network security.

This area covers the use of cryptographic algorithms in network protocols and network applications.

18. Define computer security.

This term refers to the security of computers against intruders and malicious software.

19. What are hill cipher merits and demerits?

Completely hides single letter and 2 letter frequency information.

20. List-out the types of attack in ceaser cipher.

- Brute force attack.
- Just try all the 25 possible keys.

21. What is meant by Denial of Service (DOS)?

The denial of service (DOS) is an active attack that prevents or inhibits the normal use or management of communications facilities. This attack may have a specific target, for example, an entity may suppress all messages directed to a particular destination. Another form of service denial is the disruption of an entire network either by disabling the network or by overloading it with messages so as to degrade performance.

22. What is Transposition Cipher?

In cryptography, a transposition cipher is a method of encryption by which the positions held by units of plain text are shifted according to a regular system, so that the cipher text constitutes a permutation of the plain text. The simplest such cipher is the rail fence technique.

23. What are the two problems with One-time pad?

The two problems with One-time pad are

- There is the practical problem of making large quantities of random keys.
- Even more daunting is the problem of key distribution and protection.

24. What is Abelian Group?

List the axioms should obey for Abelian Group. Let (G, *) be a group.

If a, b belongs to G and a * b = b * a, then the group is said to be Abelian or commutative group.

The following axioms are obeyed for Abelian Group.

(A1) Closure (A2) Associative (A3) Identity element (A4) Inverse element and (A5) Commutative

25. What are Rings?

A ring R, sometimes denoted by {R,+, },is a set of elements with two binary operations, called addition and multiplication, such that for all a, b, c in R the following axioms are obeyed.

(A1-A5) R is an Abelian group

(M1) Closure under multiplication

(M2) Associativity of multiplication

(M3) Distributive law

(M4) Commutative of multiplication

(M5) Multiplicative identity

(M6) No zero divisors

26. Define Fields.

A field F, sometimes denoted by {F,+, }, is a set of elements with two binary operations, called addition and multiplication, such that for all a, b, c in F the following axioms are obeyed.

(A1-A6) and (M1-M6) F is a Ring for Integral Domain (M7) Multiplicative inverse

27. List the three classes of Polynomial Arithmetic.

The three classes of Polynomial Arithmetic are Ordinary polynomial arithmetic, using the basic rules of algebra• Polynomial arithmetic in which the arithmetic on the coefficients is• performed modulo p; that is, the coefficients are in GF(p) Polynomial arithmetic in which the coefficients are in GF(p), and the• polynomials are defined modulo a polynomial m(x) whose highest power is some integer n.

28. State Euler's theorem.

Euler's theorem states that for every a and n that are relatively prime. a $\Phi(n) \equiv 1 \pmod{n}$ Where $\phi(n)$ is a totient function

29. State Fermat's theorem.

Fermat's theorem states the following. If p is prime and a is a positive integer not divisible by p, then $a^{p-1} \equiv 1 \pmod{p}$

UNIT-II

BLOCK CIPHERS & PUBLIC KEY CRYPTOGRAPHY

PART-A

1. Compare stream cipher with block cipher with example.

Stream cipher: Processes the input stream continuously and producing one element at a time. Example: caeser cipher.

Block cipher: Processes the input one block of elements at a time producing an output block for each input block. Example: DES.

2. Differentiate unconditionally secured and computationally secured.

An Encryption algorithm is unconditionally secured means, the condition is if the cipher text generated by the encryption scheme doesn't contain enough information to determine corresponding plaintext. Encryption is computationally secured means, the cost of breaking the cipher exceed the value of enough information.

Time required to break the cipher exceed the useful lifetime of information.

3. Define Diffusion & Confusion.

Diffusion:

It means each plaintext digits affect the values of many cipher text digits which is equivalent to each cipher text digit is affected by many plaintext digits. It can be achieved by performing permutation on the data. It is the relationship between the plaintext and cipher text.

Confusion:

It can be achieved by substitution algorithm. It is the relationship between cipher text and key.

4. What are the design parameters of Feistel cipher network?

- Block size
- Key size
- Number of Rounds
- Sub key generation algorithm
- Round function
- Fast software Encryption/Decryption
- Ease of analysis

5. Define Product cipher.

It means two or more basic cipher are combined and it produce the resultant cipher is called the product cipher.

6. State Avalanche effect.

A desirable property of any encryption algorithm is that a small change in either the plaintext or the key produces a significant change in the cipher text. In particular, a change in one bit of the plaintext or one bit of the key should produce a change in many bits of the cipher text.

7. Give the five modes of operation of Block cipher.

- Electronic Codebook(ECB)
- Cipher Block Chaining(CBC)
- Cipher Feedback(CFB)
- Output Feedback(OFB)

- Counter(CTR)

8. State advantages of counter mode.

- Hardware Efficiency
- Software Efficiency
- Preprocessing
- Random Access
- Simplicity
- Provable Security

9. Define Multiple Encryption.

It is a technique in which the encryption is used multiple times.
Eg: Double DES, Triple DES

10. Specify the design criteria of block cipher.

- Number of rounds
- Design of the function F
- Key scheduling

11. Define Reversible mapping.

Each plain text is maps with the unique cipher text. This transformation is called reversible mapping.

12. Specify the basic task for defining a security service.

A service that enhances the security of the data processing systems and the information transfer of an organization. The services are intended to counter security attack, and they make use of one or more security mechanism to provide the service.

13. What is the difference between link and end to end encryption?

Link Encryption	**End to End Encryption**
1.Message exposed in sending host and in intermediate nodes	1.Message encrypted in sending and intermediate nodes
2.Transperant to user	2.User applies encryption
3.Host maintains encryption facility	3.Users must determine algorithm
4.One facility for all users	4.Users selects encryption scheme
5.Can be done in hardware	5.Software implementations
6.Provides host authentication	6.Provides user authentication
7.Requires one key per(host-intermediate) Pair and (intermediate-intermediate) pair	7.Requires one key per user pair

14. What is traffic padding? What is its purpose?

Traffic padding produces cipher text output continuously, even in the absence of the plain text. A continuous random data stream is generated. When plain text is available, it is encrypted and transmitted.

When input plaintext is not present, random data are encrypted and transmittedwww.Thismakes.studentsfocusitimpossibletoforanattackertoditinguish.combetween true dataflow and padding and therefore impossible to deduce the amount of traffic.

15. List the evaluation criteria defined by NIST for AES.

The evaluation criteria for AES is as follows:

1. Security
2. Cost
3. Algorithm and implementation characteristics

16. What is Triple Encryption? How many keys are used in triple encryption?

Triple Encryption is a technique in which encryption algorithm is performed three times using three keys.

17. List the schemes for the distribution of public keys.

- Public announcement
- Publicly available directory
- Public key authority
- Public-key certificates

18. Drawback of 3-DES.

- The number of rounds in thrice as that of DES
- 3DES uses 64 bit block size
- To have higher efficiency and security a larger block size is needed.

19. List out the attacks to RSA.

- Brute force - Trying all possible private keys.
- Mathematical attacks - The approaches to factor the product of two prime numbers.
- Timing attack - Depends on the running time of the decryption algorithm.

20. Prove that 3 is a primitive root of 7.

That is, if is a primitive root of the prime number, then the numbers

$a \bmod p, a^2 \bmod p, \ldots a^{p-1} \bmod p$

3 mod 7, 9 mod 7, 27 mod 7, …. 656 mod 7

3, 2, 6,…..5.

21. Write any one technique of attacking RSA.

The most widely used public-key cryptosystem is RSA. The difficulty of attacking RSA is based on the difficulty of finding the prime factors of a composite number.

THE FACTORING PROBLEM: We can identify three approaches to attacking RSA mathematically.

1. Factor *n* into its two prime factors .This enables calculation of $\emptyset(n) = (p-1)x(q-1)$,which in turn enables determination of $d \equiv e^{-1} \pmod{\emptyset(n)}$.
2. Determine $\emptyset(n)$ directly, without first determining *p* and *q.* Again, this enables determination of $d \equiv e^{-1} \pmod{\emptyset(n)}$.
3. Determine *d* directly, without first determining $\emptyset(n)$.

22. What is differential cryptanalysis?

Differential cryptanalysis is a technique in which chosen plaintexts with particular XOR difference patterns are encrypted. The difference patterns of the resulting cipher text provide information that can be used to determine the encryption key.

23. What is linear cryptanalysis?

This attack is based on finding linear approximations to describe the transformations

performed in DES. This method can find a DES key given 2^{43} known plaintexts, as compared to 2^{47} chosen plaintexts for differential cryptanalysis. Although this is a minor improvement, because it may be easier to acquire known plaintext rather than chosen plaintext, it still leaves linear cryptanalysis infeasible as an attack on DES. So far, little work has been done by other groups to validate the linear cryptanalytic approach.

24. What are the requirements for the use of a public-key certificate scheme?

Four requirements can be placed on this particular scheme:

- Any participant can read a certificate to determine the name and public key of the certificate's owner
- Any participant can read a certificate to determine the name and public key of the certificate's owner
- Only the certificate authority can create and update certificates
  i. Any participant can verify the currency of the certificate

25. What are the different modes of operation in DES?

1. Double DES
2. Triple DES
3. Electronic Code Book
4. Counter mode
5. Cipher block chaining mode
6. Cipher Feedback mode

26. What are the CFB and OFB modes?

The Cipher Feedback (CFB) mode and the Output Feedback (OFB) mode are two standard modes of operation a block cipher.

In CFB mode the previous cipher text block is encrypted and the output produced is combined with the plaintext block using exclusive-or to produce the current cipher text block. OFB mode is similar to the CFB mode except that the quantity exclusive-oared with each plaintext block is generated independently of both the plaintext and cipher text.

27. What is DES?

Data Encryption Standard (DES) is a widely-used method of data encryption using a private (secret) key. DES applies a 56-bit key to each 64-bit block of data. The process can run in several modes and involves 16 rounds or operations.

28. Compare the symmetric and asymmetric key cryptography.

Symmetric Encryption uses a single secret key that needs to be shared among the people who needs to receive the message while Asymmetric encryption uses a pair of public key, and a private key to encrypt and decrypt messages when communicating.

10.Symmetric Encryption is an age old technique while asymmetric Encryption is relatively new.

11. Asymmetric Encryption was introduced to complement the inherent problem of the need to share the key in symmetric encryption model eliminating the need to share the key by using a pair of public-private keys.

29. What are the disadvantages of double DES?

The following are the disadvantages of double DES

1. Reduction to a single stage. \
2. Meet in the middle attacks.
3. Double DES is less secure than triple DES.
4. Double DES is within brute force attack.

UNIT-III

HASH FUNCTIONS AND DIGITAL SIGNATURES

PART-A

1. What is message authentication?

It is a procedure that verifies whether the received message comes from assigned source has not been altered. It uses message authentication codes, hash algorithms to authenticate the message.

2. Define the classes of message authentication function.

Message encryption: The entire cipher text would be used for authentication.

Message Authentication Code: It is a function of message and secret key produce a fixed length value.

Hash function: Some function that map a message of any length to fixed length which serves as authentication.

3. What are the requirements for message authentication?

The requirements for message authentication are

Disclosure: Release of message contents to any person or process not processing the appropriate cryptographic key

Traffic Analysis: Discovery of the pattern of traffic between parties. In a connection oriented application, the frequency and duration of connections could be determined. In either a connection oriented or connectionless environment, the number and length of messages between parties could be determined.

Masquerade: Insertion of messages into the network from a fraudulent source. This includes the creation of messages by an opponent that are purported to come from an authorized entity. Also included are fraudulent acknowledgements of message receipt or no receipt by someone other than the message recipient.

Content modification: Changes to the contents of a message , including insertion, deletion, transposition, and modification.

Sequence modification: Any modification to a sequence of messages between parties, including insertion, deletion, and modification.

4. What you meant by hash function?

Hash function accept a variable size message M as input and produces a fixed size hash code H(M) called as message digest as output. It is the variation on the message authentication code.

5. Differentiate MAC and Hash function?

MAC:

In Message Authentication Code, the secret key shared by sender and receiver. The MAC is appended to the message at the source at a time which the message is assumed or known to be correct.

Hash Function:

The hash value is appended to the message at the source at time when the message is assumed or known to be correct. The hash function itself not considered to be secret.

6. What are the requirements of the hash function?

It can be applied to a block of data of any size.

It produces a fixed length output.

H(x) is relatively easy to compute for any given x, making both hardware and software implementations practical.

7. What you meant by MAC?

MAC is Message Authentication Code. It is a function of message and secret key which produce a fixed length value called as MAC. MAC = Ck(M)

Where M = variable length message

K= secret key shared by sender and receiver.

CK(M) = fixed length authenticator.

8. Differentiate internal and external error control.

Internal error control:

In internal error control, an error detecting code also known as frame check sequence or checksum.

External error control:

In external error control, error detecting codes are appended after encryption.

9. What is the meet in the middle attack?

This is the cryptanalytic attack that attempts to find the value in each of the range and domain of the composition of two functions such that the forward mapping of one through the first function is the same as the inverse image of the other through the second function-quite literally meeting in the middle of the composed function.

10. What is the role of compression function in hash function?

The hash algorithm involves repeated use of a compression function f, that takes two inputs and produce a n-bit output. At the start of hashing the chaining variable has an initial value that is specified as part of the algorithm. The final value of the chaining variable is the hash value usually b>n; hence the term compression.

11. What is the difference between weak and strong collision resistance?

Weak collision resistance	Strong resistance collision
For any given block x, it is computationally infeasible to fine y≠x wit H(y)=H(x).	It is computationally infeasible to find any pair (x,y) such that H(x)=H(y).
It is proportional to 2n	It is proportional to 2 n/2

12. Distinguish between direct and arbitrated digital signature.

Direct digital signature	Arbitrated Digital Signature
1.The direct digital signature involves only the communicating parties.	1.The arbiter plays a sensitive and crucial role in this digital signature.
2.This may be formed by encrypting the entire message with the sender's private key.	2. Every signed message from a sender x to a receiver y goes first to an arbiter A, who subjects the message and its signature to a number of tests to check its origin and content.

13. What are the properties a digital signature should have?

It must verify the author and the data and time of signature. It must authenticate the contents at the time of signature. It must be verifiable by third parties to resolve disputes.

14. What requirements should a digital signature scheme should satisfy?

- The signature must be bit pattern that depends on the message being signed.
- The signature must use some information unique to the sender, to prevent both forgery and denial.
- It must be relatively easy to produce the digital signature.
- It must be relatively easy to recognize and verify the digital signature.
- It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message.
- It must be practical to retain a copy of the digital signature in storage.

15. What is meant by the Diffie-Hellman key exchange?

An element g is called a generator of a group G if every element in G can be expressed as the product of finitely many powers of g.

If $p \geq 1$ is an integer, then the numbers coprimeto p, taken modulo p, form a group with multiplication as its operation. It is written as $(Z/pZ)\times$or Zp^*.

16. How does Diffie-Hellman key exchange achieve security?

Diffie–Hellman key exchange is a specific method of exchanging cryptographic keys. It is one of the earliest practical examples of key exchange implemented within the field of cryptography. The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

17. What is weak collision resistance? What is the use of it?

For any given block x, It is computationally infeasible to find $Y \neq X$ with $H(Y) \neq H(X)$. It guarantees than an alternative message hashing to the same value as a given message cannot found. This prevents forgery when as encrypted hash code is used.

18. What is meant by EIGamal cryptosystem?

The ElGamal system is a public-key cryptosystem based on the discrete logarithm problem. It consists of both encryption and signature algorithms. The encryption algorithm is similar in nature to the Diffie-Hellman key agreement protocol.

19. What is meant by one-way property in hash function?

For any given code h, it is computationally infeasible to find X such that H(x) = h. A hash function, by itself, does not provide message authentication. A secret key must be used in some fashion with the hash function to produce authentication. A MAC, by definition, uses a secret key to calculate a code used for authentication.

20. List out the requirements of kerberos.

The requirements of Kerberos are as follows:

(1)Secure (2) Reliable (3) Transparent (4) Scalable

21. What is meant by life cycle of a key?

Keys have limited lifetimes for a number of reasons. The most important reason is protection against cryptanalysis. Each time the key is used, it generates a number of ciphertexts. Using a key repetitively allows an attacker to build up a store of ciphertext (and possibly plaintexts) which may prove sufficient for a successful cryptanalysis of the key value. If you suspect that an attacker may have obtained your key, then your key is considered compromised.

22. What is a hash function?

A hash function H is a transformation that takes a variable-size input m and returns a fixed-size string, which is called the hash value h (that is, h = H(m)). Hash functions with just this property have a variety of general computational uses, but when employed in cryptography the hash functions are usually chosen to have some additional properties.

23.What are the types of attacks addressed by message authentication?

There are four types of message authentication:

1. **Masquerade:** Insertion of messages into the network from a fraudulent source. This includes the creation of messages by an opponent that are purported to come from an authorized entity. Also included are fraudulent acknowledgments of message receipt or no receipt by someone other than the message recipient.

2. **Content modification:** Changes to the contents of a message, including insertion, deletion, transposition, and modification.

3. **Sequence modification:** Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.

4. **Timing modification:** Delay or replay of messages. In a connection-oriented application, an entire session or sequence of messages could be a replay of some previous valid session, or individual messages in the sequence could be delayed or replayed. In a connectionless application, an individual message (e.g., datagram) could be delayed or replayed.

24.What are two levels of functionality that comprise a message authentication or digital signature mechanism?

At the lower level, there must be some sort of function that produces an authenticator: a value to be used to authenticate a message. This lower-level function is then used as primitive in a higher-level authentication protocol that enables a receiver to verify the authenticity of a message.

25.What is the difference between an unconditionally secure cipher and a computationally secure cipher?

An encryption scheme is unconditionally secure if the ciphertext generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available. An encryption scheme is said to be computationally secure if:

(1) the cost of breaking the cipher exceeds the value of the encrypted information,

(2) the time required to break the cipher exceeds the useful lifetime of the information.

26. What is the difference between a message authentication code and a one-way hash function?

A hash function, by itself, does not provide message authentication. A secret key must be used in some fashion with the hash function to produce authentication. A MAC, by definition, uses a secret key to calculate a code used for authentication.

UNIT-IV
SECURITY PRACTICE & SYSTEM SECURITY

PART-A

1. Define Kerberos.

Kerberos is an authentication service developed as part of project Athena at MIT. The problem that Kerberos address is, assume an open distributed environment in which users at work stations wish to access services on servers distributed throughout the network.

2. What is Kerberos? What are the uses?

Kerberos is an authentication service developed as a part of project Athena at MIT.Kerberos provide a centralized authentication server whose functions is to authenticate servers.

3. What 4 requirements were defined by Kerberos?

Secure
Reliable
Transparent
Scalable

4. In the content of Kerberos, what is realm?

A full service Kerberos environment consisting of a Kerberos server, a no. of clients, no.of application server requires the following:

The Kerberos server must have user ID and hashed password of all participating users in its database.

The Kerberos server must share a secret key with each server. Such an environment is referred to as "Realm".

5. What is the purpose of X.509 standard?

X.509 defines framework for authentication services by the X.500 directory to its users.X.509 defines authentication protocols based on public key certificates.

6. List the 3 classes of intruder?

- Classes of Intruders
- Masquerader
- Misfeasor
- Clandestine user

7. Define virus. Specify the types of viruses?

A virus is a program that can infect other program by modifying them the modification includes a copy of the virus program, which can then go on to infect other program. Types:

- Parasitic virus
- Memory-resident virus
- Boot sector virus
- Stealth virus
- Polymorphic virus

8. What is application level gateway?

An application level gateway also called a proxy server; act as a relay of application-level traffic. The user contacts the gateway using a TCP/IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed.

9. List the design goals of firewalls?

- All traffic from inside to outside, and vise versa, must pass through the firewall.
- Only authorized traffic, as defined by the local security policy, will be allowed to pass.
- The firewall itself is immune to penetration.

10. What are the steps involved in SET Transaction?

- The customer opens an account
- The customer receives a certificate
- Merchants have their own certificate
- The customer places an order.
- The merchant is verified.
- The order and payment are sent.
- The merchant requests payment authorization.
- The merchant confirm the order.
- The merchant provides the goods or services.
- The merchant requests payment.

11. What is dual signature? What it is purpose?

The purpose of the dual signature is to link two messages that intended for two different recipients.

To avoid misplacement of orders.

12. What is the need for authentication applications?

- Security for E-mail
- Internet protocol security
- IP address security.

13. Specify the requirements for message authentication?

- Disclosure
- Traffic analysis
- Masquerade
- Content modification
- Sequence modification
- Timing modification
- Repudiation.

14. Specify the four categories of security threats?

- Interruption
- Interception
- Modification
- Fabrication

15. What do you mean by SET? What are the features of SET?

SET is an open encryption and security specification designed to protect credit card transaction on the Internet.

16. Write any 3 hash algorithm?

- MD5 algorithm x SHA-I
- RIPEMD-160 algorithm.

17. Define the classes of message authentication function.

- Message encryption

- Meassage authentication code
- Hash function.

18. List out the four phases of virus.

- Dormant phase
- Propagation phase
- Triggering phase
- Execution phase

19. What is worm?

A worm is a program that can relicate itself and send copies from computer to computer across network connections.

20. What is Bastion host?

Bastion host is a system identified by firewall administrator as critical strong point in network security.

21. What is a trusted software?

Trusted software a system that enhance the ability of a system to defend against intruders and malicious programs by implementing trusted system technology.

22. Four general techniques of firewall.

- Security control
- Direction control
- User control
- Behaviour control

Three types of firewall.

- Packet filter
- Application level gateway
- Circuit level gateway.

23. List approaches for intrusion detection.

- Statistical anamoly detection
- Rule based detection

24. What is intruder?

An intruder is an attacker who tries tog an unauthorized access to a system.

25. What is mean by SET? What are the features of SET?

Secure Electronic Transaction (SET) is an open encryption and security specification designed to protect credit card transaction on the internet.

Features are:

- Confidentiality of information
- Integrity of data
- Cardholder account authentication
- Merchant authentication

26. What is Zombie?

A Zombie is a program that securely takes over another internet-attached computer and then uses that computer to launch attacks are difficult to trace the Zombie's creator.

27 .Why does PGP generate a signature before applying compression?

The signature is generated before compression due to 2 reasons:

It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future.

28. Write the four SSL Protocols.

- SSL Handshake protocol
- SSL Change cipher spec. protocol
- SSL Alert Protocol
- SSL Record Protocol

29. What is meant by S/MIME?

S/MIME (Secure/Multipurpose Internet Mail Extensions) is a standard for public key encryptionand signingof MIMEdata. S/MIME is on an IETF standards trackand defined in a number of documents, most importantly RFCs (3369, 3370, 3850, 3851). S/MIME was originally developed by RSA Data SecurityInc. The original specification used the IETFMIME specification with the de facto industry standard PKCSsecure message format. Change control to S/MIME has since been vested in the IETF and the specification is now layered on cryptographic message syntax.

30. What are the services provided by IPSec?

The services provided by IPSec are authentication, confidentiality and key management authentication. It ensures the identity of an entity. Confidentiality is protection of data from unauthorized disclosure. Key management is generation, exchange, storage, safeguarding, etc. of keys in a public key cryptography.

31. What is meant by replay attack?

A replay attack (also known as playback attack) is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. This is carried out either by the originator or by an adversary who intercepts the data and retransmits it, possibly as part of a masquerade attack by IP packet substitution (such as stream cipher attack).

32. What is the difference between an SSL connection and SSL session?

Connection is a transport that provides a suitable type of service. For SSL, such connections are peer-topper relationships. The connections are transient. Every connection is associated with one session. Session: An SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

33. Why does ESP include a padding field?

The ciphertext needs to end on an eight octet boundary because the Authentication data field is properly aligned in the packet. This is what the protocol expects and if it doesn't follow the rules, it's considered to contain an error in the packet. It's like English or other languages. We expect sentences to end with a period so we know where one sentence ends and the other begins.

34. What is the problem that kerberos addresses?

The problem that Kerberos addresses is this: Assume an open distributed environment in which users at workstations wish to access services on servers distributed throughout the network. We would like for servers to be able to restrict access to authorized users and to be able to authenticate requests for service. In this environment a workstation cannot be trusted to identify its users correctly to network services.

35. What is meant by the function of a compression function in a hash function?

The hash function involves repeated use of a compression function. The motivation is that if the compression function is collision resistant, then the hash function is also collision resistant function. So a secure hash function can be produced.

36. How is signed data entity of S/MIME prepared?

Secure/Multipurpose Internet Mail Extension is a security enhancement to the MIME Internet e-mail format standard, based on technology from RSA data security. It is able to sign and/or encrypt messages.

37. What are the services provided by IPSec?

1. Access control
2. Connectionless integrity
3. Data origin authentication
4. Rejection of replayed packets

38. List out four requirements defined for Kerberos.

The four requirements defined for Kerberos are:

- Secure: A network eavesdropper should not be able to obtain the necessary information to impersonate a user. More generally Kerberos should be strong enough that a potential opponent does not find it to be the weak link.
- Reliable: For all services that relay on Kerberos for access control, lack of availability of the supported services. Hence, Kerberos should be highly reliable and should employ distributed server architecture, with one system able to back up another.
- Transparent: Ideally, the user should not be aware that authentication is taking place, beyond the requirement to enter a password.
- Scalable: The system should be capable of supporting large numbers of clients and servers. This suggests a modular, distributed architecture.

39. What are the entities that constitute a full-service kerberos environment?

A full service environment consists of a Kerberos server, a number of clients and a number of application servers.

40. What is the need of segmentation and reassembly function in PGP?

E-mail facilities often are restricted to a maximum message length. To accommodate this restriction, PGP automatically subdivides a message that is too large into segments that are small enough to send via e-mail. The segmentation is done after all of the other processing, including the radix-64 conversion. Thus, the session key component and signature component appear only once, at the beginning of the first segment.

UNIT-V

MAIL, IP & WEB SECURITY

PART-A

1. Define key Identifier.

PGP assigns a key ID to each public key that is very high probability unique with a user ID. It is also required for the PGP digital signature. The key ID associated with each public key consists of its least significant 64bits.

2. List the limitations of SMTP/RFC 822.

- SMTP cannot transmit executable files or binary objects.
- It cannot transmit text data containing national language characters.
- SMTP servers may reject mail message over certain size.
- SMTP gateways cause problems while transmitting ASCII and EBCDIC.
- SMTP gateways to X.400 E-mail network cannot handle non textual data included in X.400 messages.

3. Define S/MIME.

Secure/Multipurpose Internet Mail Extension(S/MIME) is a security enhancement to the MIME Internet E-mail format standard, based on technology from RSA Data Security.

4. What are the services provided by PGP services?

- Digital signature
- Message encryption
- Compression
- E-mail compatibility
- Segmentation

5. Why E-mail compatibility function in PGP needed?

Electronic mail systems only permit the use of blocks consisting of ASCII text. To accommodate this restriction PGP provides the service converting the row 8-bit binary stream to a stream of printable ASCII characters. The scheme used for this purpose is Radix-64 conversion.

6. Name any cryptographic keys used in PGP.

- One-time session conventional keys.
- Public keys.
- Private keys.
- Pass phrase based conventional keys.

7. Define S/MIME.

Secure / Multipurpose Internet Mail Extension(S/MIME) is a security enhancement to the MIME internet E-mail format standard, based on technology from RSA Data security.

8. What are the services provided by PGP services?

- Digital signature
- Compression
- Segmentation
- Message encryption
- E-mail compatibility

9. Name any cryptographic keys used in PGP.

- One time session conventional keys

- Public keys
- Private keys
- Pass phrase based conventional keys.
- What are the steps involved in SET transaction?
- The customer opens an account.
- The customer receives a certificate
- Merchants have their own certificate
- The customer places an order.
- The merchant requests payment authorization.
- The merchant confirm the order.
- The merchant requests payments.

10. List out the features of SET.

- Confidentiality
- Integrity of data
- Cardholder account authentication
- Merchant authentication

11. What is security association?

A security association (SA) is the establishment of shared security attributes between two network entities to support secure communication.

12. What does Internet key management in IPSec?

Internet key exchange (IKE) is a key management protocol standard used in conjunction with the Internet Protocol Security (IPSec) standard protocol. It provides security for Virtual Private Networks (VPNs) negotiations and network access to random hosts.

13. List out the IKE hybrid protocol dependence.

- ISAKMP - Internet Security Association and Key Management Protocols.
- Oakley

14. What does IKE hybrid protocol mean?

Internet Key Exchange (IKE) is a key management protocol standard used in conjunction with the internet protocol security (IPSec) standard protocol. It provides security for Virtual Private Networks (VPNs) negotiations and network access to random hosts.

15. What are the two security services provided by IPSec?

- Authentication Header (AH)
- Encapsulating Security Payload (ESP).
- What are the fields available in AH header?
- Next header
- Payload length
- Reserved
- Security parameter
- Sequence number Integrity check value

16. What is virtual private network?

VPN means virtual private network, a secure tunnel between two devices.

17. What is ESP?

ESP- encapsulating security payload provides authentication, integrity and confidentiality, which protect against data tempering

18. What is Behaviour-Blocking Software (BBS)?

BBS integrates with the OS of a host computer and monitors program behaviour in real time for malicious actions.

19. List password selection strategies.

- User education
- Reactive password checking
- Computer-generated password.
- Proactive password checking.

20. Define – Virus

Computer Viruses is defined as the malicious software programs that damage computer program entering into the computer without the permission of the users, and also run against the wishes of the users. They are replicated by themselves. Viruses are so dangerous and malicious that they can be automatically copied and pasted from memory to memory over and over.

Types of virus:

Boot sector Virus Macro virus
Multipartite Virus Stealth virus

21. What is application level gateway?

An application gateway or application level gateway (ALG) is a firewall proxy which provides network security. It filters incoming node traffic to certain specifications which mean that only transmitted network application data is filtered. Such network applications include File Transfer Protocol (FTP), Telnet, Real Time Streaming Protocol (RTSP) and Bit Torrent.

22. List out the design goals of firewalls.

1. All traffic from inside to outside, and vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local network except via the firewall.
2. Only authorized traffic, as defined by the local security policy, will be allowed to pass.
3. The firewall itself is immune to penetration. This implies the use of a hardened system with a secured operating system.

23. What are audit reports? Writ its two forms.

An information security audit is an audit on the level of information security in an organization. Within the broad scope of auditing information security there are multiple types of audits, multiple objectives for different audits, etc. Most commonly the controls being audited can be categorized to technical, physical and administrative. Auditing information security covers topics from auditing the physical security of data centers to auditing the logical security of databases and highlights key components to look for and different methods for auditing these areas.

24. Define − Password Protection

Password protection is defined as a security process that protects information accessible via computers that needs to be protected from certain users. Password protection allows only those with an authorized password to gain access to certain information.

25. Define − Malicious Program

Malicious software is defined as software written with the intent of causing some inconvenience to the user of the software. Malicious software in general terms is quite often called a virus however there are many other forms of malicious software. Some other types of malicious or potentially malicious software are worms, Trojan horses, spyware, and Pups.

26. What is meant by intruder?

A network is accessed by unauthorized user is called intrusion and the user is called as intruder.

Classes of intruders:
Masquerader Misfeasor
Clandestine user

27. What is meant by worm?

A computer worm is a self-replicating computer program that penetrates an operating system with the intent of spreading malicious code. Worms utilize networks to send copies of the original code to other computers, causing harm by consuming bandwidth or possibly deleting files or sending documents via email. Worms can also install backdoors on computers.

28. What is meant by Trojan horse?

In computers, a Trojan horse is a program in which malicious or harmful code is contained inside apparently harmless programming or data in such a way that it can get control and do its chosen form of damage, such as ruining the file allocation table on your hard disk. Trojan horse was a program that was supposed to find and destroy computer viruses. A Trojan horse may be widely redistributed as part of a computer virus.

29. What is meant by logic bomb?

A logic bomb is a malicious program timed to cause harm at a certain point in time, but is inactive up until that point. A set trigger, such as a preprogrammed date and time, activates a logic bomb. Once activated, a logic bomb implements a malicious code that causes harm to a computer. A logic bomb, also called slag code.

30. What are the steps in virus removal process?

Virus should be removed form the system by scanning process. The steps include in this process are,

- Backup your data
- Check to ensure that other factors aren't causing your problem
- Gather your antivirus tools
- Reboot in Safe Mode
- Run your scans
- Test your computer

31. What is meant by generic decryption technology?

A generic decryption technology can detect most complex polymorphic viruses with fast scanning speed.

32. What is meant by denial of service?

A denial of service is an attempt to prevent a genuine user of service from using it. A “denial-of-service" attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. Examples include,

- Attempts to "flood" a network, thereby preventing legitimate network traffic.
- Attempts to disrupt connections between two machines, thereby preventing access to a service.
- Attempts to prevent a particular individual from accessing a service.
- Attempts to disrupt service to a specific system or person.

PART B

UNIT I

1. Explain the followings:
   (a) Playfair cipher. (8)
   (b) Vernam cipher in detail. (8)
2. Explain simplified DES with example. (16)
3. Write short notes on i) Steganography(16)
4. Explain classical Encryption techniques in detail. (16)
5. Write short notes on
   (a) Security services(8)
   (b) Feistel cipher structure(8)
6. Explain the OSI security architecture. (16)
7. a. Explain various transposition ciphers in detail.(8)
   b. Explain the basic principle of rotor machine. (8)
8. Explain in detail about Feistel cipher with diagram. (16)
9. a.Explain classical encryption techniques with symmetric cipher model. (12)
   b. Explain steganography in detail. (4)
10. Convert "MEET ME" using Hill cipher with the key matrix Convert the cipher text back to plaintext.
11. Write short notes on block cipher modes of operation
12. (i) Discuss any four Substitution Technique and list their merits and demerits. (16)
13. Explain in detail about various types of attacks.
14. Explain in detail about various services provided by X.800.
15. Explain in detail about various Mechanisms provided by X.800.
16. Briefly explain the design principles of block cipher. (8)
17. Write short notes on
    (i)Fermat and Eluer's theorem (8)
    (ii)Chinese Remainder theorem (8)
18. Discuss with neat sketch a network security model. (8)

UNIT II

1. State and explain the principles of public key cryptography?
2. Explain Diffie Hellman key Exchange in detail with an example?
3. Explain the key management of public key encryption in detail?
4. Explain RSA algorithm in detail with an example?
5. Briefly explain the idea behind Elliptic Curve Cryptosystem?
6. Explain Data Encryption Standard (DES) in detail. (16)
7. List the evaluation criteria defined by NIST for AES. (16)
8. Using play fair cipher algorithm encrypts the message using the key "MONARCHY" and explains the poly alphabetic key. (16)
9. Explain 1.ceaser cipher 2. Mono alphabetic cipher 3.one time pad cipher (16)
10. Explain the Key Generation, Encryption and Decryption of DES algorithm in detail. (16)
11. Explain in detail the key generation in AES algorithm and its expansion format. (16)
12. a. Explain single round DES with neat sketch.(10)
    b. Explain Double &Triple DES with keys. (6)
13. Explain the block cipher modes of operation. (16)
14. Explain the key management of public key encryption in detail. (16)
15. Explain ECC - Diffie Hellman key Exchange with both keys in detail with an example. (16)
16. Write about elliptic curve architecture in detail and how they are useful for cryptography.(16)
17. a. Write about key distribution in detail. (10)

b. Explain the purpose of CRT. (6)

18. Explain the different methods used in random number generation. (16)
19. What are the requirements and applications of public key? Compare conventional with public key encryption. (16)
20. (i) Identify the possible threats for RSA algorithm and list their counter measures. (8)
    (ii) Perform decryption and encryption using RSA algorithm with p=3, q=11, e=7 and N=5.
21. (i) Draw the general structure of DES and explain the encryption decryption process. (10)
    (ii) Mention the strengths and weakness of DES algorithm. (6)
22. (i) Explain the generation sub key and S Box from the given 32-bit key by Blowfish. (8)
    (ii) In AES, hoe the encryption key is expanded to produce keys for the 10 rounds.(8)
23. (i) Describe about RC4 algorithm. (8)
    (ii) Explain the Miller-Rabin algorithm. (8)

UNIT III

1. Explain the classification of authentication function in detail.
2. Describe MD5 algorithm in detail. Compare its performance with SHA-1.
3. Describe SHA-1 algorithm in detail. Compare its performance with MD5 and RIPEMD-160 and discuss its advantages.
4. Describe RIPEMD-160 algorithm in detail. Compare its performance with
5. Describe HMAC algorithm in detail.
6. Write and explain the Digital Signature Algorithm.
7. Briefly explain Deffie Hellman key exchange with an example. (16)
8. Write and explain the digital signature algorithm. (8) (ii) Explain in detail Hash Functions.
9. Compare the Features of SHA-1 and MD5 algorithm. (8)
10. Discuss about the objectives of HMAC and it security features. (8)
11. How man in middle attack can be performed in Diffie Hellman algorithm.(4)
12. Explain in detail EIGamal Public key cryptosystem. (8)
13. Discuss clearly Secure Hash Algorithm(SHA) (8)
14. Describe the MD5 message digest algorithm with necessary block diagrams. (16)

UNIT IV

1. Explain in detail about KDC.
2. Explain the different ways of public key distribution in detail.
3. What is Kerberos? Explain how it provides authenticated service.
4. Explain the format of the X.509 certificate.
5. Explain the technical details of firewall and describe any three types of firewall with neat diagram.
6. Write short notes on Intrusion Detection.
7. Define virus. Explain in detail.
8. Describe trusted system in detail.
9. Explain the technical details of firewall and describe any three types of firewall with neat diagram.
10. Write short notes on Intrusion Detection.
11. Explain any two approaches for intrusion detection.
12. Explain firewalls and how they prevent intrusions.
13. Define intrusion detection and the different types of detection mechanisms, in detail.
14. Explain the types of Host based intrusion detection. List any two IDS software available.
15. What are the positive and negative effects of firewall?
16. Describe the familiar types of firewall configurations.
17. Explain Intrusion detection.
18. Explain the firewall design principles.

19. Name some viruses & explain it.
20. Describe about trusted systems.

UNIT V

1. Explain the operational description of PGP.
2. Write Short notes on S/MIME.
3. Explain the architecture of IP Security.
4. Write short notes on authentication header and ESP.
5. Explain in detail the operation of Secure Socket Layer in detail.
6. Explain Secure Electronic transaction with neat diagram.
7. Write brief note on E-mail Security.
8. Write brief note on IP Security.
9. Write brief note on Web Security.
10. Explain about PKI in detail.
11. Describe about SSL/TLS Protocol.
12. Explain in detail the operation of Internet Key Exchange with an example.

Subject Code: IT6702

Subject Name: Data Warehousing and Data Mining

Year/Sem: IV/VII

UNIT – I Part – A

1. Define data warehouse? [CO1-L1]

A data warehouse is a repository of multiple heterogeneous data sources organized under a unified schema at a single site to facilitate management decision making .(or)A data warehouse is a subject-oriented, time-variant and nonvolatile collection of data in support of management's decision-making process

2. What are operational databases? [CO1-L2]

Organizations maintain large database that are updated by daily transactions are called operational databases.

3. Define OLTP? [CO1-L1]

If an on-line operational database systems is used for efficient retrieval, efficient storage and management of large amounts of data, then the system is said to be on-line transaction processing.

4. Define OLAP? [CO1-L2]

Data warehouse systems serves users (or) knowledge workers in the role of data analysis and decision-making. Such systems can organize and present data in various formats. These systems are known as on-line analytical processing systems.

5. How a database design is represented in OLTP systems? [CO1-L1]

Entity-relation model

6. How a database design is represented in OLAP systems? [CO1-L1]

• Star schema • Snowflake schema • Fact constellation schema

7. Write short notes on multidimensional data model? [CO1-L2]

Data warehouses and OLTP tools are based on a multidimensional data model.This model is used for the design of corporate data warehouses and department data marts. This model contains a Star schema, Snowflake schema and Fact constellation schemas. The core of the multidimensional model is the data cube.

8. Define data cube? [CO1-L1]

It consists of a large set of facts (or) measures and a number of dimensions.

9. What are facts? [CO1-L1]

Facts are numerical measures. Facts can also be considered as quantities by which can analyze the relationship between dimensions.

10. What are dimensions? [CO1-L2]

Dimensions are the entities (or) perspectives with respect to an organization for keeping records and are hierarchical in nature.

11. Define dimension table? [CO1-L1]

A dimension table is used for describing the dimension. (e.g.) A dimension table for item may contain the attributes item_ name, brand and type.

12. Define fact table? [CO1-L1]

Fact table contains the name of facts (or) measures as well as keys to each of the related dimensional tables.

13. What are lattice of cuboids? [CO1-L1]

In data warehousing research literature, a cube can also be called as cuboids. For different (or) set of dimensions, we can construct a lattice of cuboids, each showing the data at different level. The lattice of cuboids is also referred to as data cube.

14. What is apex cuboid? [CO1-L1]

The 0-D cuboid which holds the highest level of summarization is called the apex cuboid. The apex cuboid is typically denoted by all.

15. List out the components of star schema? [CO1-L1]

- A large central table (fact table) containing the bulk of data with no redundancy.
- A set of smaller attendant tables (dimension tables), one for each dimension.

16. What is snowflake schema? [CO1-L2]

The snowflake schema is a variant of the star schema model, where some dimension tables are normalized thereby further splitting the tables in to additional tables.

17. List out the components of fact constellation schema? [CO1-L1]

This requires multiple fact tables to share dimension tables. This kind of schema can be viewed as a collection of stars and hence it is known as galaxy schema (or) fact constellation schema.

18. Point out the major difference between the star schema and the snowflake schema? [CO1-L2]

The dimension table of the snowflake schema model may be kept in normalized form to reduce redundancies. Such a table is easy to maintain and saves storage space.

19. Which is popular in the data warehouse design, star schema model (or) snowflake schema model? [CO1-L2]

Star schema model, because the snowflake structure can reduce the effectiveness and more joins will be needed to execute a query.

20. Define concept hierarchy? [CO1-L1]

A concept hierarchy defines a sequence of mappings from a set of low-level concepts to higher-level concepts.

PART – B

1) .What are the Data warehouse Architecture components? [CO1-H2]

2). How can you Building a Data warehouse? [CO1-H2]

3). How **can you Mapping the data warehouse architecture to Multiprocessor architecture? Explain. [CO1-H2]**

4). Write all the DBMS schemas for decision support. [CO1-H1]

5). Explain in detail Data Extraction, Cleanup, and Transformation Tools[CO1-H2]

6). what do you mean by Metadata? [CO1-H2]

UNIT II

1. Define schema hierarchy? [CO2-L1]

A concept hierarchy that is a total (or) partial order among attributes in a database schema is called a schema hierarchy.

2. List out the OLAP operations in multidimensional data model? [CO2-L1]

• Roll-up • Drill-down • Slice and dice • Pivot (or) rotate

3. What is roll-up operation? [CO2-L1]

The roll-up operation is also called drill-up operation which performs aggregation on a data cube either by climbing up a concept hierarchy for a dimension (or) by dimension reduction.

4. What is drill-down operation? [CO2-L2]

Drill-down is the reverse of roll-up operation. It navigates from less detailed data to more detailed data. Drill-down operation can be taken place by stepping down a concept hierarchy for a dimension

5. What is slice operation? [CO2-L1]

The slice operation performs a selection on one dimension of the cube resulting in a sub cube.

6. What is dice operation? [CO2-L1]

The dice operation defines a sub cube by performing a selection on two (or) more dimensions.

7. What is pivot operation? [CO2-L1]

This is a visualization operation that rotates the data axes in an alternative presentation of the data.

8. List out the views in the design of a data warehouse? [CO2-L1]

• Top-down view • Data source view • Data warehouse view • Business query view

9. What are the methods for developing large software systems? [CO2-L1]

- Waterfall method
- Spiral method

10. How the operation is performed in waterfall method? [CO2-L2]

The waterfall method performs a structured and systematic analysis at each step before proceeding to the next, which is like a waterfall falling from one step to the next.

11. List out the steps of the data warehouse design process? [CO2-L2]

- Choose a business process to model.
- Choose the grain of the business process
- Choose the dimensions that will apply to each fact table record.
- Choose the measures that will populate each fact table record.

12. Define ROLAP? [CO2-L1]

The ROLAP model is an extended relational DBMS that maps operations on multidimensional data to standard relational operations.

13. Define MOLAP? [CO2-L2]

The MOLAP model is a special purpose server that directly implements multidimensional data and operations.

14. Define HOLAP? [CO2-L2]

The hybrid OLAP approach combines ROLAP and MOLAP technology, benefiting from the greater scalability of ROLAP and the faster computation of MOLAP,(i.e.) a HOLAP server may allow large volumes of detail data to be stored in a relational database, while aggregations are kept in a separate MOLAP store.

15. What is enterprise warehouse? [CO2-L1]

An enterprise warehouse collects all the information's about subjects spanning the entire organization. It provides corporate-wide data integration, usually from one (or)more operational systems (or) external information providers. It contains detailed data as well as summarized data and can range in size from a few giga bytes to hundreds of giga bytes, tera bytes (or) beyond. An enterprise data warehouse may be implemented on traditional mainframes, UNIX super servers (or) parallel architecture platforms. It requires business modeling and may take years to design and build.

16. What is data mart? [CO2-L2]

Data mart is a database that contains a subset of data present in a data warehouse.Data marts are created to structure the data in a data warehouse according to issues such as hardware platforms and access control strategies. We can divide a data warehouse into data marts after the data warehouse has been created. Data marts are usually implemented on low-cost departmental servers that are UNIX (or) windows/NT based. The implementation cycle of the data mart is likely to be measured in weeks rather than months (or) years.

17. What are dependent and independent data marts? [CO2-L2]

Dependent data marts are sourced directly from enterprise data warehouses. Independent data marts are data captured from one (or) more operational systems (or) external information providers (or) data generated locally with in particular department (or) geographic area.

18. What is virtual warehouse? [CO2-L2]

A virtual warehouse is a set of views over operational databases. For efficient query processing, only some of the possible summary views may be materialized. A virtual warehouse is easy to build but requires excess capability on operational database servers.

19. Define indexing? [CO2-L2]

Indexing is a technique, which is used for efficient data retrieval (or) accessing data in a faster manner. When a table grows in volume, the indexes also increase in size requiring more storage.

20. Define metadata? [CO2-L2]

Metadata is used in data warehouse is used for describing data about data. (i.e.) meta data are the data that define warehouse objects. Metadata are created for the data names and definitions of the given warehouse.

Part - B

1. Define all the Reporting and query tools for data analysis:- [CO2-H2]

2. What are the need for applications:- [CO2-H2]

3. Write short notes on Online Analytical Processing [CO2-H1]

4. What are the needs for OLAP? [CO2-H2]

5. Explain Multidimensional Data Model with neat diagram. [CO2-H2]

7. Briefly explain about the Categories of OLAP Tools. [CO2-H2]

UNIT III

Part- A

1. Define Data mining? [CO3-L2]

It refers to extracting or "mining" knowledge from large amount of data. Data mining is a process of discovering interesting knowledge from large amounts of data stored either, in database, data warehouse, or other information repositories.

2. Give some alternative terms for data mining. [CO3-L2]

- Knowledge mining
- Knowledge extraction
- Data/pattern analysis.
- Data Archaeology
- Data dredging

3. What is KDD? [CO3-L1]

KDD-Knowledge Discovery in Databases.

4. What are the steps involved in KDD process? [CO3-L2]

- Data cleaning
- Data Mining
- Pattern Evaluation
- Knowledge Presentation
- Data Integration• Data Selection
- Data Transformation

5. What is the use of the knowledge base? [CO3-L1]

Knowledge base is domain knowledge that is used to guide search or evaluate the Interestingness of resulting pattern. Such knowledge can include concept hierarchies used to organize attribute /attribute values in to different levels of abstraction.

6 What is the purpose of Data mining Technique? [CO3-L1]

It provides a way to use various data mining tasks.

7. Define Predictive model? [CO3-L1]

It is used to predict the values of data by making use of known results from a different set of sample data.

8. Define descriptive model? [CO3-L1]

It is used to determine the patterns and relationships in a sample data. Data mining tasks that belongs to descriptive model: • Clustering • Summarization • Association rules • Sequence discovery

9. Define the term summarization? [CO3-L1]

The summarization of a large chunk of data contained in a web page or a document. Summarization = characterization=generalization

10. List out the advanced database systems? [CO3-L2]

- Extended-relational databases
- Object-oriented databases
- Deductive databases
- Spatial databases
- Temporal databases
- Multimedia databases
- Active databases• Scientific databases

- Knowledge databases

11. Define cluster analysis? [CO3-L2]

Cluster analyses data objects without consulting a known class label. The class labels are not present in the training data simply because they are not known to begin with.

12. Describe challenges to data mining regarding data mining methodology and user interaction issues? [CO3-L2]

- Mining different kinds of knowledge in databases
- Interactive mining of knowledge at multiple levels of abstraction
- Incorporation of background knowledge
- Data mining query languages and ad hoc data mining
- Presentation and visualization of data mining results
- Handling noisy or incomplete data
- Pattern evaluation

13. Describe challenges to data mining regarding performance issues? [CO3-L1]

- Efficiency and scalability of data mining algorithms
- Parallel, distributed, and incremental mining algorithms

14. Describe issues relating to the diversity of database types? [CO3-L1]

- Handling of relational and complex types of data
- Mining information from heterogeneous databases and global information systems

15. What is meant by pattern? [CO3-L2]

Pattern represents knowledge if it is easily understood by humans; valid on test data with some degree of certainty; and potentially useful, novel,or validates a hunch about which the used was curious. Measures of pattern interestingness, either objective or subjective, can be used to guide the discovery process.

16. How is a data warehouse different from a database? [CO3-L2]

Data warehouse is a repository of multiple heterogeneous data sources, organized under a unified schema at a single site in order to facilitate management decision- making. Database consists of a collection of interrelated data.

17 Define Association Rule Mining [CO3-L1]

.

Association rule mining searches for interesting relationships among items in a given data set.

18. When we can say the association rules are interesting? [CO3-L1]

Association rules are considered interesting if they satisfy both a minimum support threshold and a minimum confidence threshold. Users or domain experts can set such thresholds

. **19. Define support and confidence in Association rule mining. [CO3-L3]**

Support S is the percentage of transactions in D that contain AUB. Confidence c is the percentage of transactions in D containing A that also contain B. Support (A=>B)= P(AUB) Confidence (A=>B)=P(B/A)

20. How are association rules mined from large databases? [CO3-L1]

I step: Find all frequent item sets: II step: Generate strong association rules from frequent item sets

21. Describe the different classifications of Association rule mining? [CO3-L2]

- Based on types of values handled in the Rule
- Based on the dimensions of data involved
- Based on the levels of abstraction involved
- Based on various extensions

PART –B

1. Explain Data mining confluence of multiple disciplines. [CO3-H2]

2. Write down the Data mining functionalities. [CO3-H2]

3. Describe the Interestingness of patterns. [CO3-H1]

4. What are the Classification of Data Mining Systems? [CO3-H2]

5.Name all the Data Mining Task Primitives. [CO3-H1]

6.What are the Major Issues in Data Mining? [CO3-H1]

7. What is Data Preprocessing and Why preprocess the data?. Also explain Data cleaning ,Data integration and transformation, Data reduction, Discretization and concept hierarchy generation. [CO3-H3]

UNIT 4

PART – A

1. What is the purpose of Apriori Algorithm? [CO4-L2]

Apriori algorithm is an influential algorithm for mining frequent item sets for Boolean association rules. The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent item set properties.

2. Define anti-monotone property? [CO4-L1]

If a set cannot pass a test, all of its supersets will fail the same test as well

3. How to generate association rules from frequent item sets? [CO4-L2]

Association rules can be generated as follows For each frequent item set1, generate all non empty subsets of 1. For every non empty subsets s of 1, output the rule "S=>(1-s)"if Support count(1)=min_conf, Support_count(s) where min_conf is the minimum confidence threshold.

4. Give few techniques to improve the efficiency of Apriori algorithm? [CO4-L2]

- Hash based technique
- Transaction Reduction
- Portioning
- Sampling
- Dynamic item counting

5. What are the things suffering the performance of Apriori candidate generation technique? [CO4-L2]

- Need to generate a huge number of candidate sets

- Need to repeatedly scan the scan the database and check a large set of candidates by pattern matching

6. Describe the method of generating frequent item sets without candidate generation? [CO4-L2]

Frequent-pattern growth(or FP Growth) adopts divide-and-conquer strategy. Steps:

- Compress the database representing frequent items into a frequent pattern tree or FP tree,
- Divide the compressed database into a set of conditional database, • Mine each conditional database separately.

7. Mention few approaches to mining Multilevel Association Rules? [CO4-L2]

- Uniform minimum support for all levels(or uniform support) • Using reduced minimum support at lower levels(or reduced support) • Level-by-level independent • Level-cross filtering by single item
- Level-cross filtering by k-item set

8. What are multidimensional association rules? [CO4-L2]

Association rules that involve two or more dimensions or predicates

- **Inter dimension association rule:** Multidimensional association rule with no repeated predicate or dimension.
- **Hybrid-dimension association rule:** Multidimensional association rule with multiple occurrences of some predicates or dimensions.

9. Define constraint-Based Association Mining? [CO4-L1]

Mining is performed under the guidance of various kinds of constraints provided by the user. The constraints include the following • Knowledge type constraints • Data constraints • Dimension/level constraints • Interestingness constraints • Rule constraints.

10. Define the concept of classification? [CO4-L2]

Two step process • A model is built describing a predefined set of data classes or concepts. • The model is constructed by analyzing database tuples described by attributes.The model is used for classification.

11 What is Decision tree? [CO4-L1]

A decision tree is a flow chart like tree structures, where each internal node denotes a test on an attribute, each branch represents an outcome of the test,and leaf nodes represent classes or class distributions. The top most in a tree is the root node.

12. What is Attribute Selection Measure? [CO4-L1]

The information Gain measure is used to select the test attribute at each node in the decision tree. Such a measure is referred to as an attribute selection measure or a measure of the goodness of split.

13. Describe Tree pruning methods. [CO4-L1]

When a decision tree is built, many of the branches will reflect anomalies in the training data due to noise or outlier. Tree pruning methods address this problem of over fitting the data. Approaches:

- Pre pruning
- Post pruning

14. Define Pre Pruning[CO4-L1]

A tree is pruned by halting its construction early. Upon halting, the node becomes a leaf. The leaf may hold the most frequent class among the subset samples.

15. Define Post Pruning. [CO4-L1]

Post pruning removes branches from a “Fully grown” tree. A tree node is pruned by removing its branches. Eg: Cost Complexity Algorithm

16. What is meant by Pattern? [CO4-L1]

Pattern represents the knowledge.

17. Define the concept of prediction. [CO4-L2]

Prediction can be viewed as the construction and use of a model to assess the class of an unlabeled sample or to assess the value or value ranges of an attribute that a given sample is likely to have.

18 What is the use of Regression [CO4-L2]

Regression can be used to solve the classification problems but it can also be used for applications such as forecasting. Regression can be performed using many different types of techniques; in actually regression takes a set of data and fits the data to a formula.

19 What are the requirements of cluster analysis? [CO4-L2]

The basic requirements of cluster analysis are

- Dealing with different types of attributes.
- Dealing with noisy data.
- Constraints on clustering.
- Dealing with arbitrary shapes.
- High dimensionality
- Ordering of input data
- Interpretability and usability
- Determining input parameter and
- Scalability

20.What are the different types of data used for cluster analysis? [CO4-L1]

The different types of data used for cluster analysis are interval scaled, binary, nominal, ordinal and ratio scaled data.

PART -B

1. Describe the Mining Frequent Patterns and Associations & Correlations. [CO4-H2]

2. Write all the Mining Methods? [CO4-H3]

3. What are the Various Kinds of Mining Association Rules?

4.Explain in detail -Correlation Analysis (correlation - relationship)

5.write about Constraint Based Association Mining[CO4-H1]

6. Explain the basic concepts in Classification and Prediction[CO4-H2]

7. How can you Classify Decision Tree Induction (generation) [CO4-H1]

8. Explain Bayesian Classification with examples. [CO4-H3]
9.Rule Based Classification – describe [CO4-H1]

10. Why Classification by Back propagation happened? [CO4-H1]

11.Associative Classification Classification by Association Rule Analysis [CO4H1]

12.**Lazy Learners (or Learning from Your Neighbours) [CO4-H2]**

13. What are the following Classification Methods. [CO4-H2]

14.Explain the following Predictions (Numeric prediction / Regression) **[CO4-H2]**

UNIT V

1. Define Clustering? [CO5-L1]

Clustering is a process of grouping the physical or conceptual data object into clusters.

2. What do you mean by Cluster Analysis? [CO5-L1]

A cluster analysis is the process of analyzing the various clusters to organize the different objects into meaningful and descriptive objects.

3. What are the fields in which clustering techniques are used? [CO5-L2]

• Clustering is used in biology to develop new plants and animal taxonomies. • Clustering is used in business to enable marketers to develop new distinct groups of their customers and characterize the customer group on basis of purchasing. • Clustering is used in the identification of groups of automobiles Insurance policy customer. • Clustering is used in the identification of groups of house in a city on the basis of house type, their cost and geographical location.• Clustering is used to classify the document on the web for information discovery.

4.What are the requirements of cluster analysis? [CO5-L2]

The basic requirements of cluster analysis are • Dealing with different types of attributes. • Dealing with noisy data. • Constraints on clustering. • Dealing with arbitrary shapes. • High dimensionality • Ordering of input data • Interpretability and usability • Determining input parameter and • Scalability

5.What are the different types of data used for cluster analysis? [CO5-L2]

The different types of data used for cluster analysis are interval scaled, binary, nominal, ordinal and ratio scaled data.

6.What are interval scaled variables? [CO5-L1]

Interval scaled variables are continuous measurements of linear scale. For Example , height and weight, weather temperature or coordinates for any cluster. These measurements can be calculated using Euclidean distance or Minkowski distance

7. Define Binary variables? And what are the two types of binary variables? [CO5L2]

Binary variables are understood by two states 0 and 1, when state is 0, variable is absent and when state is 1, variable is present. There are two types of binary variables, symmetric and asymmetric binary variables. Symmetric variables are those variables that have same state values and weights. Asymmetric variables are those variables that have not same state values and weights.

8. Define nominal, ordinal and ratio scaled variables? [CO5-L1]

A nominal variable is a generalization of the binary variable. Nominal variable has more than two states, For example, a nominal variable, color consists of four states, red, green, yellow, or black. In Nominal variables the total number of states is N and it is denoted by letters, symbols or integers. An ordinal variable also has more than two states but all these states are ordered in a meaningful sequence. A ratio scaled variable makes positive measurements on a non-linear scale, such as exponential scale, using the formula AeBt or Ae-Bt Where A and B are constants.

9. **What do you mean by partitioning method? [CO5-L2]**

In partitioning method a partitioning algorithm arranges all the objects intovarious partitions, where the total number of partitions is less than the total number of objects. Here each partition represents a cluster. The two types of partitioning method are kmeans and k-medoids.

10. **Define CLARA and CLARANS? [CO5-L1]**

Clustering in LARge Applications is called as CLARA. The efficiency of CLARA depends upon the size of the representative data set. CLARA does not work properly if any representative data set from the selected representative data sets does not find best kmedoids. To recover this drawback a new algorithm, Clustering Large Applications based upon RANdomized search (CLARANS) is introduced. The CLARANS works like CLARA, the only difference between CLARA and CLARANS is the clustering process that is done after selecting the representative data sets.

11. **What is Hierarchical method? [CO5-L2]**

Hierarchical method groups all the objects into a tree of clusters that are arranged in a hierarchical order. This method works on bottom-up or top-down approaches.

12. **Differentiate Agglomerative and Divisive Hierarchical Clustering? [CO5-L2]**

Agglomerative Hierarchical clustering method works on the bottom-up approach.In Agglomerative hierarchical method, each object creates its own clusters. The single Clusters are merged to make larger clusters and the process of merging continues until all the singular clusters are merged into one big cluster that consists of all the objects. Divisive Hierarchical clustering method works on the top-down approach. In this method all the objects are arranged within a big singular cluster and the large cluster is continuously divided into smaller clusters until each cluster has a single object.

13. **What is CURE? [CO5-L1]**

Clustering Using Representatives is called as CURE. The clustering algorithms generally work on spherical and similar size clusters. CURE overcomes the problem of spherical and similar size cluster and is more robust with respect to outliers.

14. **Define Chameleon method? [CO5-L1]**

Chameleon is another hierarchical clustering method that uses dynamic modeling. Chameleon is introduced to recover the drawbacks of CURE method. In this method two clusters are merged, if the interconnectivity between two clusters is greater than the interconnectivity between the objects within a cluster.

15. **Define Density based method? [CO5-L1]**

Density based method deals with arbitrary shaped clusters. In density-based method, clusters are formed on the basis of the region where the density of the objects is high.

16. **What is a DBSCAN? [CO5-L2]**

Density Based Spatial Clustering of Application Noise is called as DBSCAN. DBSCAN is a density based clustering method that converts the high-density objects regions into clusters with arbitrary shapes and sizes. DBSCAN defines the cluster as a maximal set of density connected points.

17. **What do you mean by Grid Based Method? [CO5-L1]**

In this method objects are represented by the multi resolution grid data structure. All the objects are quantized into a finite number of cells and the collection of cells build the grid structure of objects. The clustering operations are performed on that grid structure. This method is widely used because its processing time is very fast and that is independent of number of objects.

18. **What is a STING? [CO5-L1]**

Statistical Information Grid is called as STING; it is a grid based multi resolution clustering method. In STING method, all the objects are contained into rectangular cells, these cells are kept into various levels of resolutions and these levels are arranged in a hierarchical structure.

19. **Define Wave Cluster? [CO5-L2]**

It is a grid based multi resolution clustering method. In this method all the objects are represented by a multidimensional grid structure and a wavelet transformation is applied for finding the dense region. Each grid cell contains the information of the group of objects that map into a cell. A wavelet transformation is a process of signaling that produces the signal of various frequency sub bands.

20. **What is Model based method? [CO5-L1]**

For optimizing a fit between a given data set and a mathematical model based methods are used. This method uses an assumption that the data are distributed by probability distributions.

21. **What is the use of Regression? [CO5-L2]**

Regression can be used to solve the classification problems but it can also be used for applications such as forecasting. Regression can be performed using many different types of techniques; in actually regression takes a set of data and fits the data to a formula

22. **What are the reasons for not using the linear regression model to estimate the output data? [CO5-L2]**

There are many reasons for that, One is that the data do not fit a linear model, It is possible however that the data generally do actually represent a linear model, but thelinear model generated is poor because noise or outliers exist in the data. Noise is erroneous data and outliers are data values that are exceptions to the usual and expected data.

23. **What do u mean by logistic regression? [CO5-L2]**

Instead of fitting a data into a straight line logistic regression uses a logistic curve. The formula for the univariate logistic curve is P= e (C0+C1X1) 1+e (C0+C1X1) The logistic curve gives a value between 0 and 1 so it can be interpreted as the probability of class membership.

24. **What is Time Series Analysis? [CO5-L1]**

A time series is a set of attribute values over a period of time. Time Series Analysis may be viewed as finding patterns in the data and predicting future values.

25What is Smoothing? [CO5-L1]

Smoothing is an approach that is used to remove the nonsystematic behaviors found in time series. It usually takes the form of finding moving averages of attribute values. It is used to filter out noise and outliers.

Part B

UNIT V CLUSTERING AND APPLICATIONS AND TRENDS IN DATA MINING

1. Briefly explain all the Cluster Analysis concepts with suitable examples [CO5H1]

2. What are the types of Data in cluster analysis? [CO5-H2]

3. Categorize the Major Clustering Methods in detail. [CO5-H2]

4. Which Hierarchical clustering methods is called agglomerative? [CO5-H2]

5. Density-Based Methods clustering – explain [CO5-H2]

6. Grid-Based Methods “STING” explain.

7. Model-Based Clustering Methods [CO5-H2]

8. Clustering High-Dimensional Data [CO5-H2]

9. Constraint-Based Cluster Analysis [CO5-H2]

10. Outlier Analysis [CO5-H2]

UNIT-V
University Questions
PART A

1. What are the requirements of clustering?
2. What are the applications of spatial data bases?
3. What is text mining?
4. Distinguish between classification and clustering.
5. Define a Spatial database.
7. What is the objective function of K-means algorithm?
8. Mention the advantages of Hierarchical clustering.
9. What is an outlier? Give example.
10. What is audio data mining?
11. List two application of data mining.

PART-B

1. BIRCH and CLARANS are two interesting clustering algorithms that perform effective clustering in large data sets.
(i) Outline how BIRCH performs clustering in large data sets. [10] (ii) Compare and outline the major differences of the two scalable clustering algorithms BIRCH and
CLARANS. [6]
2. Write a short note on web mining taxonomy. Explain the different activities of text mining.
3. Discuss and elaborate the current trends in data mining. [6+5+5]
4. Discuss spatial data bases and Text databases [16]
5. What is a multimedia database? Explain the methods of mining multimedia database? [16]
6. (a) Explain the following clustering methods in detail.
(a) BIRCH (b) CURE [16]
7. Discuss in detail about any four data mining applications. [16]
8. Write short notes on
(i) Partitioning methods [8] (ii) Outlier analysis [8]
9. Describe K means clustering with an example. [16]
10. Describe in detail about Hierarchical methods.

Subject Name :Grid and Cloud Computing

Year/Sem: IV/VII

Unit I - Introduction
Part - A

1. Define cloud computing

Cloud computing is the delivery of computing as a service rather than a product, hereby shared resources, software, and information are provided to computers and other devices as a utility.

2. What is Distributed computing

This is a field of computer science/engineering that studies distributed systems. A distributed system consists of multiple autonomous computers, each having its own private memory, communicating through a computer network. Information exchange in a distributed system is accomplished through message passing. A computer program that runs in a distributed system is known as a distributed program. The process of writing distributed programs is referred to as distributed programming.

3. Difference between distributed and parallel computing.

Distributed	Parallel
Each processor has its own private memory (distributed memory). Information is	All processors may have access to a shared memory to exchange information

exchanged by passing messages between the processors.	between processors.
It is loosely coupled.	It is tightly coupled.
An important goal and challenge of distributed systems is location transparency.	Large problems can often be divided into smaller ones, which are then solved concurrently ("in parallel").

4. What is mean by service oriented architecture?

In grids/web services, Java, and CORBA, an entity is, respectively, a service, a Java object, and a CORBA distributed object in a variety of languages. These architectures build on the traditional seven Open Systems Interconnection (OSI) layers that provide the base networking abstractions. On top of this we have a base software environment, which would be .NET or Apache Axis for web services, the Java Virtual Machine for Java, and a broker network for CORBA.

5. What is High Performance Computing(HPC).

supercomputer sites and large data centers must provide high-performance computing services to huge numbers of Internet users concurrently.Because of this high demand, the Linpack Benchmark for high-performance computing (HPC) applications is no longer optimal for measuring system performance. The emergence of computing clouds instead demands high-throughput computing (HTC) systems built with parallel and distributed computing technologies. We have to upgrade data centers using fast servers, storage systems, and high-bandwidth networks. The purpose is to advance network-based computing and web services with the emerging new technologies.

6. Define peer-to-peer network.

The P2P architecture offers a distributed model of networked systems. Every node acts as both a client and a server, providing part of the system resources. Peer machines are simply client computers connected to the Internet. All client machines act autonomously to join or leave the system freely. This implies that no master-slave relationship exists among the peers. No central coordination or central database is needed.

4. What are the Three New Computing Paradigms Radio-frequency identification (RFID), Global Positioning System (GPS), Internet of Things (IoT).

5. What is degree of parallelism and types

The degree of parallelism (DOP) is a metric which indicates how many operations can be or are being simultaneously executed by a computer. It is especially useful for describing the performance of parallel programs and multi-processor systems.

- Bit-level parallelism (BLP)
- Instruction-level parallelism (ILP)
- VLIW (very long instruction word)
- Data-level parallelism (DLP)
- Multicore processors and chip multiprocessors (CMPs)
- Job-level parallelism (JLP)

9. What is Cyber-Physical Systems

A cyber-physical system (CPS) is the result of interaction between computational processes and the physical world. A CPS integrates ―cyber‖ (heterogeneous, asynchronous) with ―physical‖ (concurrent and information-dense) objects.

10. Define multi core CPU.

Advanced CPUs or microprocessor chips assume a multi-core architecture with dual, quad, six, or more processing cores. These processors exploit parallelism at ILP and TLP levels. CPU has reached its limit in terms of exploiting massive DLP due to the aforementioned memory wall problem

11. Define GPU.

A GPU is a graphics coprocessor or accelerator on a computer's graphics card or video card. A GPU offloads the CPU from tedious graphics tasks in video editing applications. The GPU chips can process a minimum of 10 million polygons per second. GPU's have a throughput architecture that exploits massive parallelism by executing many concurrent threads.

12. Clusters of Cooperative Computers

A computing cluster consists of interconnected stand-alone computers which work cooperatively as a single integrated computing resource.

13. W hat is single-system image (SSI)

An ideal cluster should merge multiple system images into a single-system image (SSI). Cluster designers desire a cluster operating system or some middleware to support SSI at various levels, including the sharing of CPUs, memory, and I/O across all cluster nodes.

14. What is Grid Computing

Grid computing is the collection of computer resources from multiple locations to reach a common goal. The grid can be thought of as a distributed system with non-interactive workloads that involve a large number of files. Grid computing is distinguished from conventional high performance computing systems such as cluster computing in that grid computers have each node set to perform a different task/application.

15. What is Computational Grids

A computing grid offers an infrastructure that couples computers,software/middleware, special instruments, and people and sensors together. The grid is often constructed across LAN, WAN, or Internet backbone networks at a regional, national, or global scale. Enterprises or organizations present grids as integrated computing resources.

16. What is Overlay Networks and its types

Overlay is a virtual network formed by mapping each physical machine with its ID, logically, through a virtual mapping . When a new peer joins the system, its peer ID is added as a node in the overlay network.

Two types of overlay networks:

1.Unstructured 2. Structured.

7. Write the any three Grid Applications.

- Schedulers
- Resource Broker
- Load Balancing

8. Difference between grid and cloud computing

Grid computing	cloud computing
Grids enable access to shared computing power and storage capacity from your desktop	Clouds enable access to leasedcomputing power and storage capacity from your desktop
In computing centres distributed across different sites, countries and continents	The cloud providers private data centres which are often centralised in a few locations with excellent network connections and cheap electrical power.
Grids were designed to handle large sets of limited duration jobs that produce or use large quantities of data (e.g. the LHC)	Clouds best support long term services and longer running jobs (E.g. facebook.com)

19. What are the derivatives of grid computing?

There are 8 derivatives of grid computing. They are as follows:

a)Compute grid

- Data grid
- Science grid

12. Access grid
13. Knowledge grid
14. Cluster grid
15. Terra grid
16. Commodity grid.

20. What is grid infrastructure?

Grid infrastructure forms the core foundation for successful grid applications. This infrastructure is a complex combination of number of capabilities and resources identified for the specific problem and environment being addressed.

16. What are the Applications of High-Performance and High-Throughput Systems

1.**Science and engineering**- Scientific simulations, genomic analysis, etc.Earthquake prediction, global warming, weather forecasting, etc.

- **Business, education, services industry, and health care**- Telecommunication, content delivery, e-commerce, etc.Banking, stock exchanges, transaction processing, etc.Air traffic control, electric power grids, distance education, etc.Health care, hospital automation, telemedicine, etc
- **Internet and web services,and government applications**- Internet search, data centers, decision-making systems, etc.Traffic monitoring, worm containment, cyber security, etc.Digital government, online tax return processing, social networking, etc.

19. What is Utility computing?

It is a service provisioning model in which a service provider makes computing resources and infrastructure management available to the customer as needed, and charges them for specific usage rather than a flat rate

23. What is SLA?

A service-level agreement (SLA) is a part of a standardized service contract where a service is formally defined. Particular aspects of the service – scope, quality, responsibilities – are agreed between the service provider and the service user. A common feature of an SLA is a contracted delivery time (of the service or performance)

Part - B

1. Describe about Evolution of Distributed computing.

Distributed computing is a field of computer science that studies distributed systems.

A distributed system is a model in which components located on networked computers communicate and coordinate their actions by passing messages The components interact with each other in order to achieve a **common** goal. Three significant characteristics of distributed systems are: concurrency of components, lack of a global clock, and independent failure of components. Examples of distributed systems vary from SOA-based systems to massively multiplayer online games to peer-to-peer applications.

A computer program that runs in a distributed system is called a **distributed program**, and distributed programming is the process of writing such programs. There are many alternatives for the message passing mechanism, including pure HTTP, RPC-like connectors and message queues.

HISTORY

The use of concurrent processes that communicate by message-passing has its roots in operating system architectures studied in the 1960s. The first widespread distributed systems were local-area networks such as Ethernet, which was invented in the 1970s.

ARPANET, the predecessor of the Internet, was introduced in the late 1960s, and ARPANET e-mail was invented in the early 1970s. E-mail became the most successful application of ARPANET, and it is probably the earliest example of a large-scale distributed application. In addition to ARPANET, and its successor, the Internet, other early worldwide computer networks included Usenet and FidoNet from the 1980s, both of which were used to support distributed discussion systems.

The study of distributed computing became its own branch of computer science in the late 1970s and early 1980s. The first conference in the field, Symposium on Principles of Distributed Computing (PODC), dates back to 1982, and its European counterpart International Symposium on Distributed Computing (DISC) was first held in 1985.

20. Explain in detail about Scalable computing over the Internet

A parallel and distributed computing system uses multiple computers to solve large-scale problems over the Internet. Thus, distributed computing becomes data-intensive and network-centric. identifies the applications of modern computer systems that practice parallel and distributed computing. These large-scale Internet applications have significantly enhanced the quality of life and information services in society today.

The Age of Internet Computing

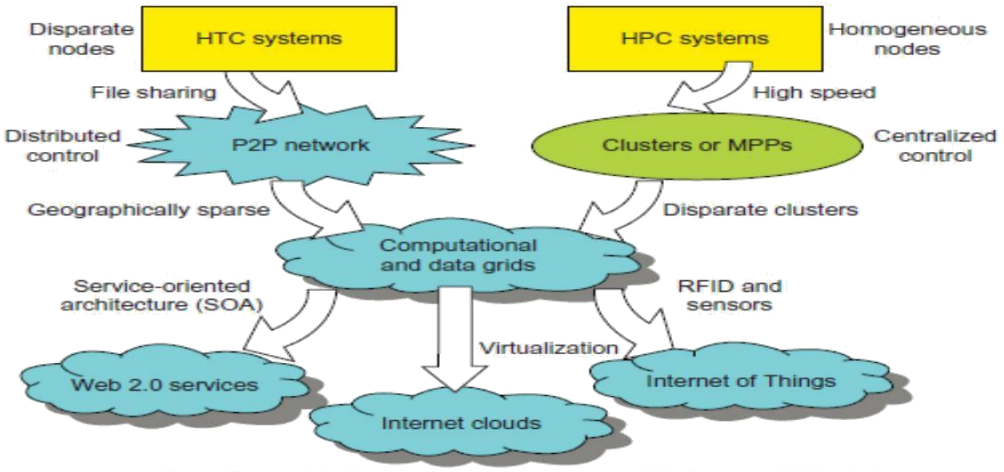
Billions of people use the Internet every day. As a result, supercomputer sites and large data centers must provide high-performance computing services to huge numbers of Internet users concurrently.Because of this high demand, the Linpack Benchmark for high-performance computing (HPC) applications is no longer optimal for measuring system performance. The emergence of computing clouds instead demands high-throughput computing (HTC) systems built with parallel and distributed computing technologies We have to upgrade data centers using fast servers, storage systems, and high-bandwidth networks. The purpose is to advance network-based computing and web services with the emerging new technologies.

The Platform Evolution

Computer technology has gone through five generations of development, with each generation lasting from 10 to 20 years. Successive generations are overlapped in about 10 years. For instance, from 1950 to 1970, a handful of mainframes, including the IBM 360 and CDC 6400, were built to satisfy the demands of large businesses and government organizations. From 1960 to 1980, lower-cost minicomputers such as the DEC PDP 11 and VAX Series became popular among small businesses and on college campuses.

- From 1970 to 1990, we saw widespread use of personal computers built with VLSI microprocessors.

- From 1980 to 2000, massive numbers of portable computers and pervasive devices appeared inboth wired and wireless applications.



Evolutionary trend toward parallel, distributed, and cloud computing with clusters, MPPs, P2P networks, grids, clouds, web services, and the Internet of Things.

High-Performance Computing

The speed of HPC systems has increased from Gflops in the early 1990s to now Pflops in 2010. This improvement was driven mainly by the demands from scientific, engineering, and manufacturing communities

High-Throughput Computing

The development of market-oriented high-end computing systems is undergoing a strategic change

from an HPC paradigm to an HTC paradigm. This HTC paradigm pays more attention to high-flux computing. The main application for high-flux computing is in Internet searches and web

services by millions or more users simultaneously. The performance goal thus shifts to measure high throughput or the number of tasks completed per unit of time.

Three New Computing Paradigms

radio-frequency identification (RFID), Global Positioning System (GPS), and sensor technologies has triggered the development of the Internet of Things (IoT).

Computing Paradigm Distinctions

In general distributed computing is the opposite of centralized computing. The field of parallel computing overlaps with distributed computing to a great extent, and cloud computing overlaps with distributed, centralized, and parallel computing.

Centralized computing this is a computing paradigm by which all computer resources are centralized in one physical system. All resources (processors, memory, and storage) are fully shared and tightly coupled within one integrated OS. Many data centers and supercomputers are centralized systems, but they are used in parallel, distributed, and cloud computing applications

 Parallel computing in parallel computing, all processors are either tightly coupled with centralized shared memory or loosely coupled with distributed memory. Some authors refer to this discipline as parallel processing. Interprocessor communication is accomplished through shared memory or via message passing. A computer system capable of parallel computing is commonly known as a parallel computer . Programs running in a parallel computer are called parallel programs. The process of writing parallel programs is often
referred to as parallel programming .

 Distributed computing This is a field of computer science/engineering that studies distributed systems. A distributed system consists of multiple autonomous computers, each having its own private memory, communicating through a computer network. Information exchange in a distributed system is accomplished through message passing. A computer program that runs in a distributed system is known as a distributed program. The process of writing distributed programs is referred to as distributed programming.

 Cloud computing An Internet cloud of resources can be either a centralized or a distributed computing system. The cloud applies parallel or distributed computing, or both. Clouds can be built with physical or virtualized resources over large data centers that are centralized or distributed. Some authors consider cloud computing to be a form of utility computing or service computing .

The high-tech community prefer the term concurrent computing or concurrent programming. parallel computing and distributing computing, although biased practitioners may interpret them differently.Ubiquitous computing refers to computing with pervasive devices at any place and time using wired or wireless communication. The Internet of Things (IoT) is a networked connection of everyday objects including computers, sensors, humans, etc. The IoT is supported by Internet clouds to achieve ubiquitous computing with any object at any place and time. Finally, the term Internet computing is even broader and covers all computing paradigms over the Internet.

Distributed System Families

Since the mid-1990s, technologies for building P2P networks and networks of clusters have been consolidated into many national projects designed to establish wide area computing infrastructures, known as computational grids or data grids.

Meeting these goals requires yielding the following design objectives:

Efficiency measures the utilization rate of resources in an execution model by exploiting massive parallelism in HPC. For HTC, efficiency is more closely related to job throughput, data access, storage, and power efficiency.

 Dependability measures the reliability and self-management from the chip to the system and application levels. The purpose is to provide high-throughput service with Quality of Service (QoS) assurance, even under failure conditions.

 Adaptation in the programming model measures the ability to support billions of job requests over massive data sets and virtualized cloud resources under various workload and service models.

3. **Flexibility** in application deployment measures the ability of distributed systems to run well in both HPC (science and engineering) and HTC (business) applications

Scalable Computing Trends and New Paradigms

Several predictable trends in technology are known to drive computing applications. In fact, designers and programmers want to predict the technological capabilities of future systems. For instance, Jim Gray's paper, ―Rules of Thumb in Data Engineering,‖ is an excellent example of how technology affects applications and vice versa. In addition, Moore's law indicates that processor speed doubles every 18 months. Although Moore's law has been proven valid over the last 30 years, it is difficult to say whether it will continue to be true in the future.

Degrees of Parallelism

when hardware was bulky and expensive, most computers were designed in a bit-serial fashion. In this scenario, bit-level parallelism (BLP) converts bit-serial processing to word-level processing gradually. users graduated from 4-bit microprocessors to 8-, 16-, 32-, and 64-bit CPUs. This led us to the next wave of improvement, known as instruction-levelparallelism (ILP), in which the processor executes multiple instructions simultaneously rather thanonly one instruction at a time. practiced ILP through pipelining, superscalar computing, VLIW (very long instruction word) architectures, and multithreading. ILP requires branch prediction, dynamic scheduling, speculation, and compiler support to work efficiently. Data-level parallelism (DLP) was made popular through SIMD (single instruction, multiple data) and vector machines using vector or array types of instructions. DLP requires even more hardware support and compiler assistance to work properly. Ever since the introduction of multicore processors and chip multiprocessors (CMPs), exploring task-level parallelism (TLP).A modern processor explores all of the aforementioned parallelism types. In fact, BLP, ILP, and As we move from parallel processing to distributed processing, increase in computing granularity to job-level parallelism (JLP). It is fair to say that coarse-grain parallelism is built on top of fine-grain parallelism.

Innovative Applications

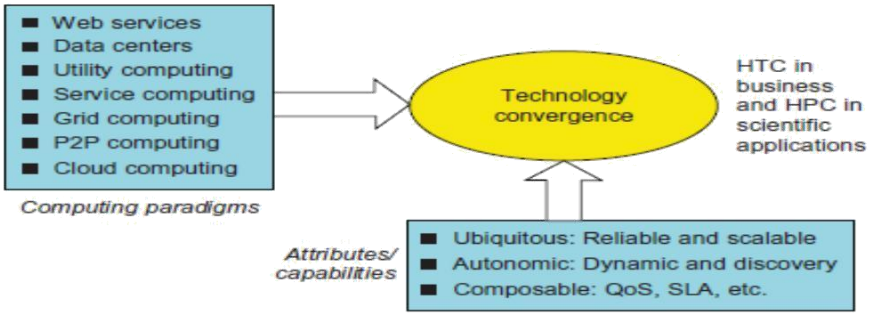
Applications of High-Performance and High-Throughput Systems

Domain	Specific Applications
Science and engineering	Scientific simulations, genomic analysis, etc. Earthquake prediction, global warming, weather forecasting, etc.
Business, education, services industry, and health care	Telecommunication, content delivery, e-commerce, etc. Banking, stock exchanges, transaction processing, etc. Air traffic control, electric power grids, distance education, etc. Health care, hospital automation, telemedicine, etc.
Internet and web services, and government applications	Internet search, data centers, decision-making systems, etc. Traffic monitoring, worm containment, cyber security, etc. Digital government, online tax return processing, social networking, etc.
Mission-critical applications	Military command and control, intelligent systems, crisis management, etc.

The Trend toward Utility Computing

All ubiquitous in daily life. Reliability and scalability are two major design objectives in these computing models.Second, they are aimed at autonomic operations that can be self-organized to support dynamic discovery.Finally, these paradigms are composable with QoS and SLAs (service-level agreements).

These paradigms and their attributes realize the computer utility vision.



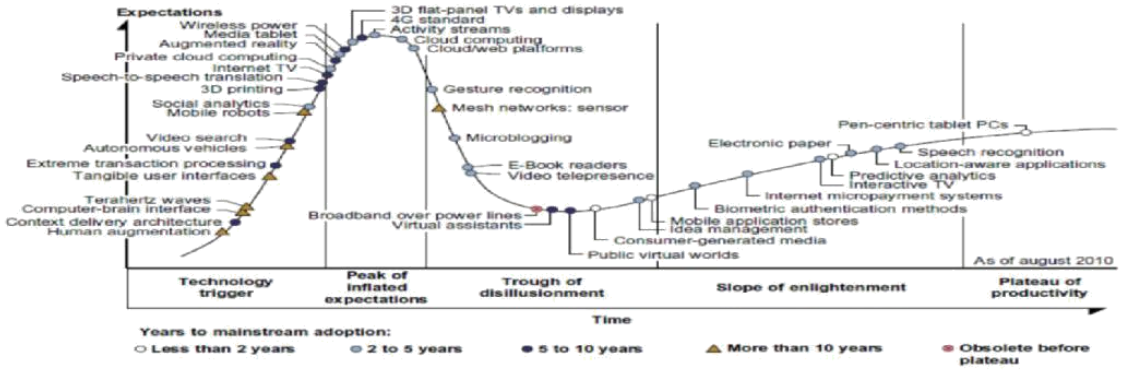
The vision of computer utilities in modern distributed computing systems.

The Hype Cycle of New Technologies

For example, at that time consumer-generated media was at the disillusionment stage, and it was predicted to take less than two years to reach its plateau of adoption. Internet micropayment systems were forecast to take two to five years to move from the enlightenment stage to maturity. It was believed that 3D printing would take five to 10 years to move from the rising expectation stage to mainstream adoption, and mesh network sensors were expected to take more than 10 years to move from the inflated expectation stage to a plateau of mainstream adoption.

The Internet of Things and Cyber-Physical Systems The Internet of Things

The traditional Internet connects machines to machines or web pages to web pages. The concept of the IoT was introduced in 1999 at MIT. The IoT refers to the networked interconnection of everyday objects, tools, devices, or computers. One can view the IoT as a wireless network of sensors that interconnect all things in our daily life.



Hype cycle for Emerging Technologies, 2010.

Cyber-Physical Systems

A cyber-physical system (CPS) is the result of interaction between computational processes and the physical world. A CPS integrates ―cyber‖ (heterogeneous, asynchronous) with ―physical‖ (concurrent and information-dense) objects. A CPS merges the ―3C‖ technologies of computation, communication, and control into an intelligent closed feedback system between the physical world and the information world, a concept which is actively explored in the United States. The IoT emphasizes various networking connections among physical objects, while the CPS emphasizes exploration of virtual reality (VR) applications in the physical world.

3. Explain in detail about Multicore CPUs and Multithreading Technologies

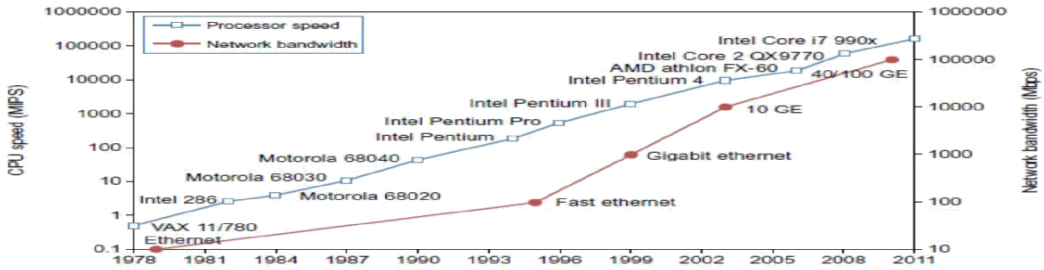
The growth of component and network technologies over the past 30 years. They are crucial to the development of HPC and HTC systems. processor speed is measured in millions of instructions per second (MIPS) and network bandwidth is measured in megabits per second (Mbps) or gigabits per second (Gbps). The unit GE refers to 1 Gbps Ethernet bandwidth

Advances in CPU Processors

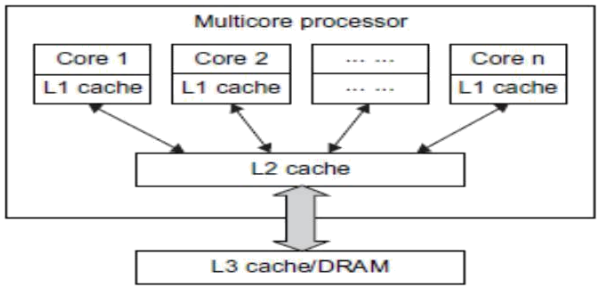
Advanced CPUs or microprocessor chips assume a multicore architecture with dual, quad, six, or more processing cores. These processors exploit parallelism at ILP and TLP levels. Processor speed growth is plotted in the upper curve in the diagram across generations of microprocessors or CMPs. We see growth from 1 MIPS for the VAX 780 in 1978 to 1,800 MIPS for the Intel Pentium 4 in 2002, up to a 22,000 MIPS peak for the Sun Niagara 2 in 2008. As the figure shows, Moore’s law has proven to be pretty accurate in this case. The clock rate for these processors increased from 10 MHz for the Intel 286 to 4 GHz for the Pentium 4 in 30 years.

The clock rate reached its limit on CMOS-based chips due to power limitations. At the time of this writing, very few CPU chips run with a clock rate exceeding 5 GHz. In other words, clock rate will not continue to improve unless chip technology matures. This limitation is attributed primarily to excessive heat generation with high frequency or high voltages. The ILP is highly exploited in modern CPU processors. ILP mechanisms include multiple-issue superscalar architecture, dynamic branch prediction, and speculative execution, among others. These ILP

techniques demand hardware and compiler support. In addition, DLP and TLP are highly explored in graphics processing units (GPUs) that adopt a many-core architecture with hundreds to thousands of simple cores



Improvement in processor and network technologies over 33 years.



Schematic of a modern multicore CPU chip using a hierarchy of caches, where L1 cache is private to each core, on-chip L2 cache is shared and L3 cache or DRAM Is off the chip.

Both multi-core CPU and many-core GPU processors can handle multiple instruction threads at different magnitudes today. the architecture of a typical multicore processor. Each core is essentially a processor with its own private cache (L1 cache). Multiple cores are housed in the same chip with an L2 cache that is shared by all cores. In the future, multiple CMPs could be built on the same CPU chip with even the L3 cache on the chip. Multicore and multithreaded CPUs are equipped with many high-end processors, including the Intel i7, Xeon, AMD Opteron, Sun Niagara, IBM Power 6, and X cell processors. Each core could be also multithreaded. For example, the Niagara II is built with eight cores with eight threads handled by each core. This implies that the maximum ILP and TLP that can be exploited in Niagara is 64 (8 × 8 = 64). In 2011, the Intel Core i7 990x has reported 159,000 MIPS execution rate as shown in the uppermost square

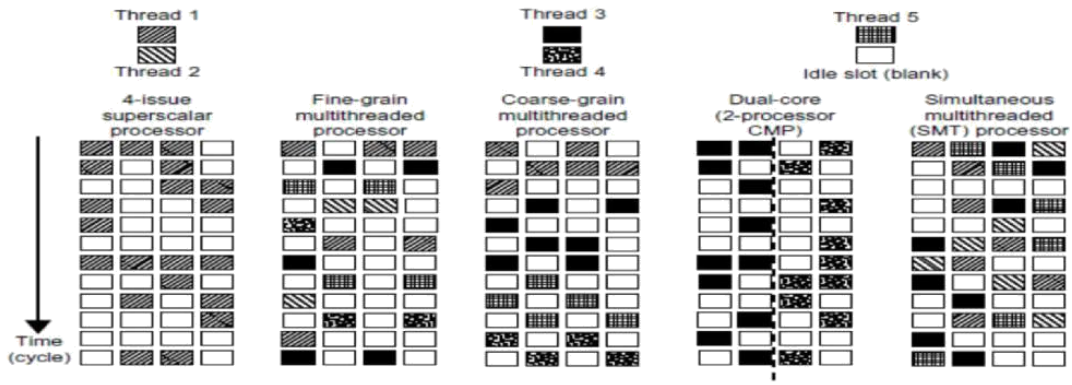
Multicore CPU and Many-Core GPU Architectures

Multicore CPUs may increase from the tens of cores to hundreds or more in the future. But the CPU has reached its limit in terms of exploiting massive DLP due to the aforementioned memory wall problem. This has triggered the development of many-core GPUs with hundreds or more thin cores. Both IA-32 and IA-64 instruction set architectures are built into commercial CPUs. Now, x-86 processors have been extended to serve HPC and HTC systems in some high-end server processors.

Many RISC processors have been replaced with multicore x-86 processors and many-core GPUs in the Top 500 systems. This trend indicates that x-86 upgrades will dominate in data centers and supercomputers. The GPU also has been applied in large clusters to build supercomputers in MPPs. In the future, the processor industry is also keen to develop asymmetric or heterogeneous chip multiprocessors that can house both fat CPU cores and thin GPU cores on the same chip

Multithreading Technology

The dispatch of five independent threads of instructions to four pipelined datapaths (functional units) in each of the following five processor categories from left to right: a



Five micro-architectures in modern CPU processors, that exploit ILP and TLP supported by multicore and multithreading technologies.

Four-issue superscalar processor, a fine-grain multithreaded processor, a coarse-grain multithreaded processor, a two-core CMP, and a simultaneous multithreaded (SMT) processor. The superscalar processor is single-threaded with four functional units. Each of the three multithreaded processors is four-way multithreaded over four functional data paths. In the dual-core processor, assume two processing cores, each a single-threaded two-way superscalar processor.

Instructions from different threads are distinguished by specific shading patterns for instructions from five independent threads. Typical instruction scheduling patterns are shown here. Only instructions from the same thread are executed in a superscalar processor. Fine-grain multithreading switches the execution of instructions from different threads per cycle. Course-grain multithreading executes many instructions from the same thread for quite a few cycles before switching to another thread. The multicore CMP executes instructions from different threads completely. The SMT allows simultaneous scheduling of instructions from different threads in the same cycle

These execution patterns closely mimic an ordinary program. The blank squares correspond to no available instructions for an instruction data path at a particular processor cycle. More blank cells imply lower scheduling efficiency. The maximum ILP or maximum TLP is difficult to achieve at each processor cycle. The point here is to demonstrate your understanding of typical instruction scheduling patterns in these five different micro-architectures in modern processors.

4. Explain in detail about GPU Computing to Exascale and Beyond

A GPU is a graphics coprocessor or accelerator mounted on a computer's graphics card or video card. A GPU offloads the CPU from tedious graphics tasks in video editing applications. The world's first GPU, the GeForce 256, was marketed by NVIDIA in 1999. These GPU chips can process a minimum of 10 million polygons per second, and are used in nearly every computer on the market today. Some GPU features were also integrated into certain CPUs. Traditional CPUs are structured with only a few cores. For example, the Xeon X5670 CPU has six cores. However, a modern GPU chip can be built with hundreds of processing cores.

GPUs have a throughput architecture that exploits massive parallelism by executing many concurrent threads slowly, instead of executing a single long thread in a conventional microprocessor very quickly. Lately, parallel GPUs or GPU clusters have been garnering a lot of attention against the use of CPUs with limited parallelism. General-purpose computing on GPUs, known as GPGPUs, have appeared in the HPC field. NVIDIA's CUDA model was for HPC using GPGPUs.

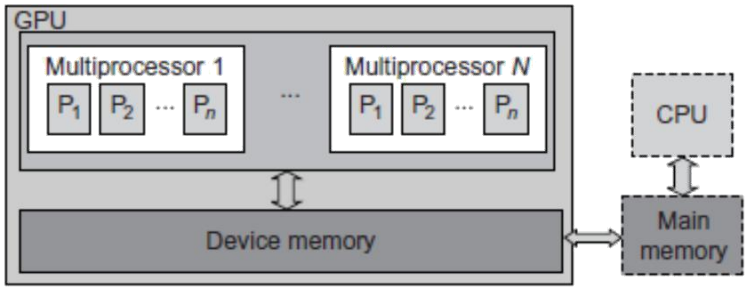
How GPUs Work

Early GPUs functioned as coprocessors attached to the CPU. Today, the NVIDIA GPU has been upgraded to 128 cores on a single chip. Furthermore, each core on a GPU can handle eight threads of instructions. This translates to having up to 1,024 threads executed concurrently on a single GPU. This is true massive parallelism, compared to only a few threads that can be

handled by a conventional CPU. The CPU is optimized for latency caches, while the GPU is optimized to deliver much higher throughput with explicit management of on-chip memory Modern GPUs are not restricted to accelerated graphics or video coding. They are used in HPC systems to power supercomputers with massive parallelism at multicore and multithreading levels. GPUs are designed to handle large numbers of floating-point operations in parallel. In a way, the GPU offloads the CPU from all data-intensive calculations, not just those that are related to video processing. Conventional GPUs are widely used in mobile phones, game consoles, embedded systems, PCs, and servers. The NVIDIA CUDA Tesla or Fermi is used in GPU clusters or in HPC systems for parallel processing of massive floating-pointing data.

GPU Programming Model

The interaction between a CPU and GPU in performing parallel execution of floating-point operations concurrently. The CPU is the conventional multicore processor with limited parallelism to exploit. The GPU has a many-core architecture that has hundreds of simple processing cores organized as multiprocessors. Each core can have one or more threads. Essentially, the CPU's floating-point kernel computation role is largely offloaded to the many-core GPU. The CPU instructs the GPU to perform massive data processing. The bandwidth must be matched between the on-board main memory and the on-chip GPU memory.



The use of a GPU along with a CPU for massively parallel execution in hundreds or thousands of processing cores.

In November 2010, three of the five fastest supercomputers in the world (the Tianhe-1a, Nebulae, and Tsubame) used large numbers of GPU chips to accelerate floating-point computations. the architecture of the Fermi GPU, a next-generation GPU from NVIDIA. This is a streaming multiprocessor (SM) module. Multiple SMs can be built on a single GPU chip. The Fermi chip has 16 SMs implemented with 3 billion transistors. Each SM comprises up to 512 streaming processors (SPs), known as CUDA cores. The Tesla GPUs used in the Tianhe-1a have a similar architecture, with 448 CUDA cores.

All functional units and CUDA cores are interconnected by an NoC (network on chip) to a large number of SRAM banks (L2 caches). Each SM has a 64 KB L1 cache. The 768 KB unified L2 cache is shared by all SMs and serves all load, store, and texture operations. Memory controllers are used to connect to 6 GB of off-chip DRAMs. The SM schedules threads in groups of 32 parallel threads called warps. In total, 256/512 FMA (fused multiply and add) operations can be done in parallel to produce 32/64-bit floating-point results. The 512 CUDA cores in an SM can work in parallel to deliver up to 515 Gflops of double-precision results, if fully utilized. With 16 SMs, a single GPU has a peak speed of 82.4 Tflops. Only 12 Fermi GPUs have the potential to reach the Pflops performance thousand-core GPUs may appear in Exascale (Eflops or 1018

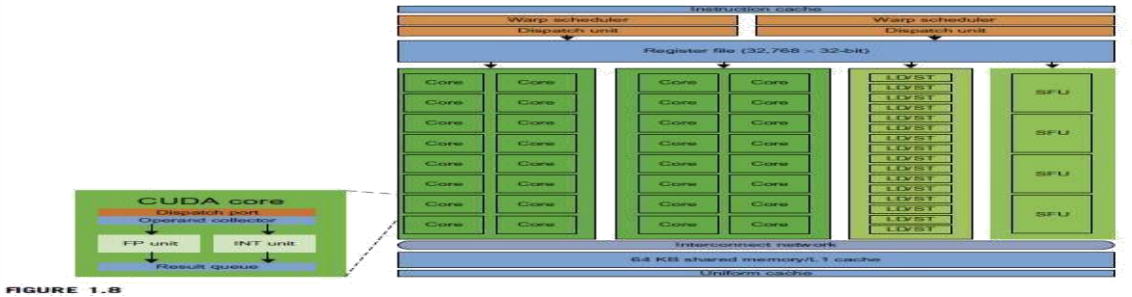


FIGURE 1.8

flops) systems. This reflects a trend toward building future MPPs with hybrid architectures of both types of processing chips. In a DARPA report published in September 2008, four challenges are identified for exascale computing: (1) energy and power, (2) memory and storage, (3) concurrency and locality, and (4) system resiliency

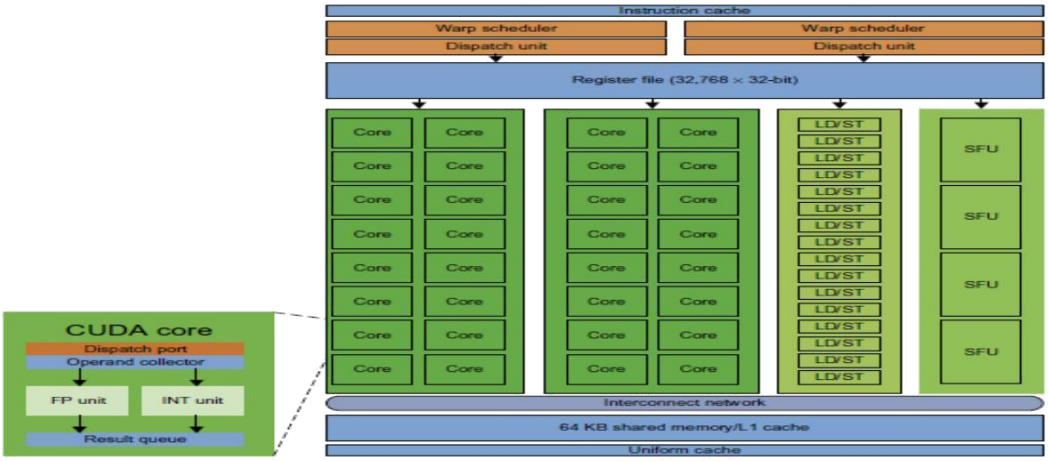
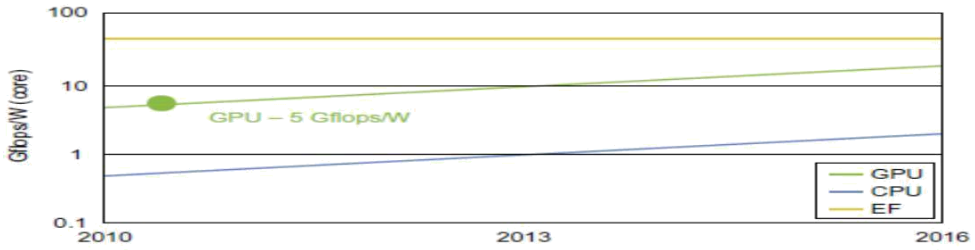


FIGURE 1.8

NVIDIA Fermi GPU built with 16 streaming multiprocessors (SMs) of 32 CUDA cores each; only one SM Is

Power Efficiency of the GPU

Bill Dally of Stanford University considers power and massive parallelism as the major benefits of GPUs over CPUs for the future. By extrapolating current technology and computer architecture, it was estimated that 60 Gflops/watt per core is needed to run an exaflops system Power constrains what we can put in a CPU or GPU chip. Dally has estimated that the CPU chip consumes about 2 nJ/instruction, while the GPU chip requires 200 pJ/instruction, which is 1/10 less than that of the CPU. The CPU is optimized for latency in caches and memory, while the GPU is optimized for throughput with explicit management of on-chip memory.



The GPU performance (middle line, measured 5 Gflops/W/core in 2011), compared with the lower CPU performance (lower line measured 0.8 Gflops/W/core in 2011) and the estimated 60 Gflops/W/core performance in 2011 for the Exascale (EF in upper curve) in the future.

This may limit the scaling of future supercomputers. However, the GPUs may close the gap with the CPUs. Data movement dominates power consumption. One needs to optimize the storage hierarchy and tailor the memory to the applications. We need to promote self-aware OS and runtime support and build locality-aware compilers and auto-tuners for GPU based MPPs. This implies that both power and software are the real challenges in future parallel and distributed computing system

2. i)Describe about Virtual Machines and Virtualization Middleware

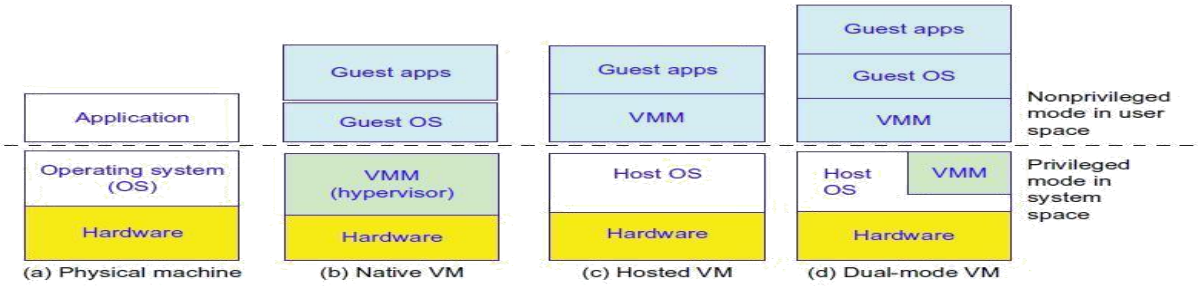
A conventional computer has a single OS image. This offers a rigid architecture that tightly couples application software to a specific hardware platform. Some software running well on one machine may not be executable on another platform with a different instruction set under a fixed OS. Virtual machines (VMs) offer novel solutions to underutilized resources, application inflexibility, software manageability, and security concerns in existing physical machines.

To build large clusters, grids, and clouds, we need to access large amounts of computing, storage, and networking resources in a virtualized manner.

In particular, a cloud of provisioned resources must rely on virtualization of processors, memory, and I/O facilities dynamically

Virtual Machines

The host machine is equipped with the physical hardware, as shown at the bottom of the figure. An example is an x-86 architecture desktop running its installed Windows OS, as shownin part of the figure. The VM can be provisioned for any hardware system. The VM is builtwith virtual resources managed by a guest OS to run a specific application. Between the VMs and the host platform, one needs to deploy a middleware layer called a virtual machine monitor (VMM).



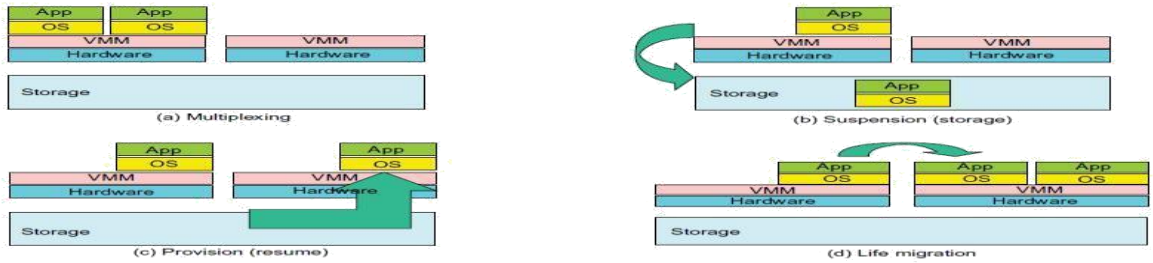
Three VM architectures in (b), (c), and (d), compared with the traditional physical machine shown in (a).

Shows a native VM installed with the use of a VMM called a hypervisor in privileged mode. For example, the hardware has x-86 architecture running the Windows system. The guest OS could be a Linux system and the hypervisor is the XEN system developed at Cambridge University. This hypervisor approach is also called bare-metal VM, because the hypervisor handles the bare hardware (CPU, memory, and I/O) directly. Another architecture is the host VM

The VM approach offers hardware independence of the OS and applications. The user application running on its dedicated OS could be bundled together as a virtual appliance that can be ported to any hardware platform. The VM could run on an OS different from that of the host computer.

VM Primitive Operations

The VMM provides the VM abstraction to the guest OS. With full virtualization, the VMM exports a VM abstraction identical to the physical machine so that a standard OS such as Windows 2000 or Linux can run just as it would on the physical hardware



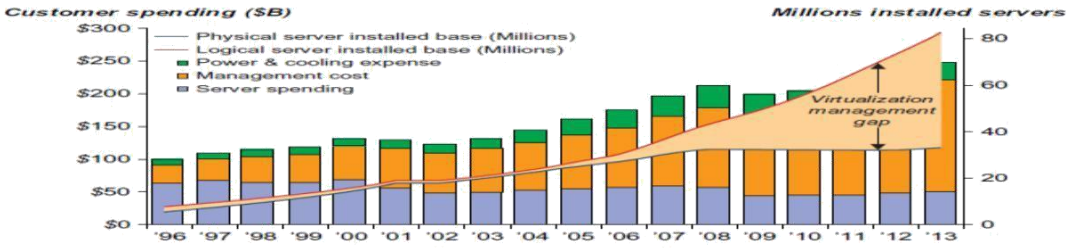
VM multiplexing, suspension, provision, and migration in a distributed computing environment.

These VM operations enable a VM to be provisioned to any available hardware platform. They also enable flexibility in porting distributed application executions. Furthermore, the VM approach will significantly enhance the utilization of server resources

Virtual Infrastructures

Physical resources for compute, storage, and networking at the bottom of are mapped to the needy applications embedded in various VMs at the top. Hardware and software are then separated. Virtual infrastructure is what connects resources to distributed applications. It is a

dynamic mapping of system resources to specific applications. The result is decreased costs and increased efficiency and responsiveness.



Growth and cost breakdown of data centers over the years.

5. ii) Explain in detail about Data Center Virtualization for Cloud Computing .

Basic architecture and design considerations of data centers. Cloud architecture is built with commodity hardware and network devices. Almost all cloud platforms choose the popular x86 processors. Low-cost terabyte disks and Gigabit Ethernet are used to build data centers. Data center design emphasizes the performance/price ratio over speed performance alone. In other words, storage and energy efficiency are more important than shear speed performance.

Data Center Growth and Cost Breakdown

A large data center may be built with thousands of servers. Smaller data centers are typically built with hundreds of servers. The cost to build and maintain data center servers has increased over the years. Typically only 30 percent of data center costs goes toward purchasing IT equipment (such as servers and disks), 33 percent is attributed to the chiller, 18 percent to the uninterruptible power supply (UPS), 9 percent to computer room air conditioning (CRAC), and the remaining 7 percent to power distribution, lighting, and transformer costs. Thus, about 60 percent of the cost to run a data center is allocated to management and maintenance The server purchase cost did not increase much with time. The cost of electricity and cooling did increase from 5 percent to 14 percent in 15 years.

Low-Cost Design Philosophy

High-end switches or routers may be too cost-prohibitive for building data centers. Thus, using high-bandwidth networks may not fit the economics of cloud computing. using commodity x86 servers is more desired over expensive mainframes. The software layer handles network traffic balancing, fault tolerance, and expandability. Currently, nearly all cloud computing data centers use Ethernet as their fundamental network technology

Convergence of Technologies

cloud computing is enabled by the convergence of technologies in four areas: (1) hardware virtualization and multi-core chips, (2) utility and grid computing, (3) SOA, Web 2.0, and WS mashups, and (4) atonomic computing and data center automation. Hardware virtualization and multicore chips enable the existence of dynamic configurations in the cloud. Utility and grid computing technologies lay the necessary foundation for computing clouds Recent advances in SOA, Web 2.0, and mashups of platforms are pushing the cloud another step forward. Finally, achievements in autonomic computing and automated data center operations contribute to the rise of cloud computing.

Jim Gray once posted the following question: ―Science faces a data deluge. How to manage and analyze information?‖ This implies that science and our society face the same challenge of data deluge. Data comes from sensors, lab experiments, simulations, individual archives, and the web in all scales and formats. Preservation, movement, and access of massive data sets require generic tools supporting high-performance, scalable file systems, databases, algorithms, workflows, and visualization

On January 11, 2007, the Computer Science and Telecommunication Board (CSTB) recommended fostering tools for data capture, data creation, and data analysis. A cycle of interaction exists among four technical areas. First, cloud technology is driven by a surge of

interest in data deluge. Also, cloud computing impacts e-science greatly, which explores multicore and parallel computing technologies.

By linking computer science and technologies with scientists, a spectrum of e-science or e-research applications in biology, chemistry, physics, the social sciences, and the humanities has generated new insights from interdisciplinary activities

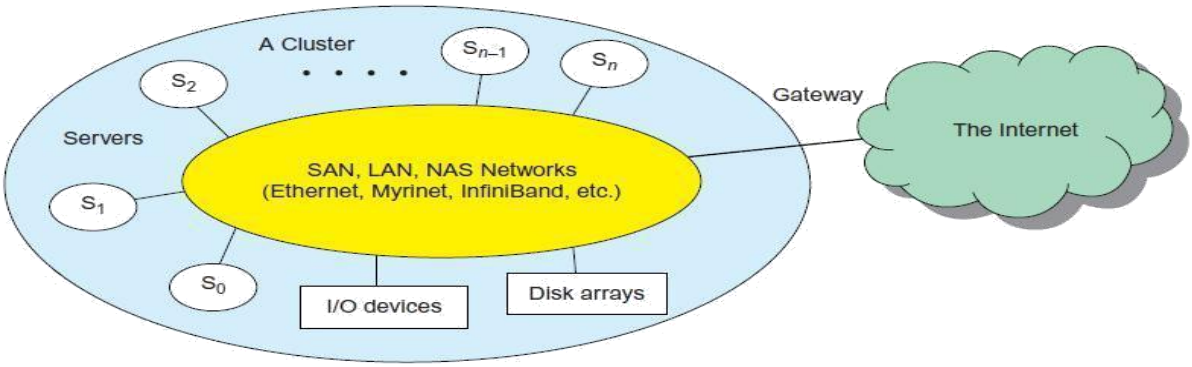
Iterative MapReduce extends MapReduce to support a broader range of data mining algorithms commonly used in scientific applications. The cloud runs on an extremely large cluster of commodity computers. Internal to each cluster node, multithreading is practiced with a large number of cores in many-core GPU clusters

Explain in detail about clusters of cooperative computers

A computing cluster consists of interconnected stand-alone computers which work cooperatively as a single integrated computing resource. In the past, clustered computer systems have demonstrated impressive results in handling heavy workloads with large data sets

Cluster Architecture

Architecture of a typical server cluster built around a low-latency, highbandwidth interconnection network. This network can be as simple as a SAN (e.g., Myrinet) or a LAN (e.g., Ethernet). To build a larger cluster with more nodes, the interconnection network can be built with multiple levels of Gigabit Ethernet, Myrinet, or InfiniBand switches. Through hierarchical construction using a SAN, LAN, or WAN, one can build scalable clusters with an increasing number of nodes. The cluster is connected to the Internet via a virtual private network (VPN) gateway. The gateway IP address locates the cluster. The system image of a computer is decided by the way the OS manages the shared cluster resources. Most clusters have loosely coupled node computers. All resources of a server node are managed by their own OS. Thus, most clusters have multiple system images as a result of having many autonomous nodes under different OS control.



A cluster of servers interconnected by a high-bandwidth SAN or LAN with shared I/O devices and disk arrays; the cluster acts as a single computer attached to the Internet.

Single-System Image

An ideal cluster should merge multiple system images into a single-system image (SSI). Cluster designers desire a cluster operating system or some middleware to support SSI at various levels, including the sharing of CPUs, memory, and I/O across all cluster nodes.

An SSI is an illusion created by software or hardware that presents a collection of resources as one integrated, powerful resource. SSI makes the cluster appear like a single machine to the user. A cluster with multiple system images is nothing but a collection of independent computers.

Hardware, Software, and Middleware Support

Cluster design principles for both small and large clusters. Clusters exploring massive parallelism are commonly known as MPPs. Almost all HPC clusters in the Top 500 list are also MPPs. The building blocks are computer nodes (PCs, workstations, servers, or SMP), special

communication software such as PVM or MPI, and a network interface card in each computer node. Most clusters run under the Linux OS.

Special cluster middleware supports are needed to create SSI or high availability (HA). Both sequential and parallel applications can run on the cluster, and special parallel environments are needed to facilitate use of the cluster resources. For example, distributed memory has multiple images. Users may want all distributed memory to be shared by all servers by forming distributed
shared memory (DSM).

Major Cluster Design Issues

A cluster-wide OS for complete resource sharing is not available yet. Middleware or OS extensions were developed at the user space to achieve SSI at selected functional levels. Without this middleware, cluster nodes cannot work together effectively to achieve cooperative computing.

Table 1.3 Critical Cluster Design Issues and Feasible Implementations

Features	Functional Characterization	Feasible Implementations
Availability and Support	Hardware and software support for sustained HA in cluster	Failover, failback, check pointing, rollback recovery, nonstop OS, etc.
Hardware Fault Tolerance	Automated failure management to eliminate all single points of failure	Component redundancy, hot swapping, RAID, multiple power supplies, etc.
Single System Image (SSI)	Achieving SSI at functional level with hardware and software support, middleware, or OS extensions	Hardware mechanisms or middleware support to achieve DSM at coherent cache level
Efficient Communications	To reduce message-passing system overhead and hide latencies	Fast message passing, active messages, enhanced MPI library, etc.
Cluster-wide Job Management	Using a global job management system with better scheduling and monitoring	Application of single-job management systems such as LSF, Codine, etc.
Dynamic Load Balancing	Balancing the workload of all processing nodes along with failure recovery	Workload monitoring, process migration, job replication and gang scheduling, etc.
Scalability and Programmability	Adding more servers to a cluster or adding more clusters to a grid as the workload or data set increases	Use of scalable interconnect, performance monitoring, distributed execution environment, and better

e. Explain in detail about Grid computing Infrastructures

Users have experienced a natural growth path from Internet to web and grid computing services. Internet services such as the Telnet command enables a local computer to connect to a remote computer.

Web service such as HTTP enables remote access of remote web pages. Grid computing is envisioned to allow close interaction among applications running on distant computers simultaneously.

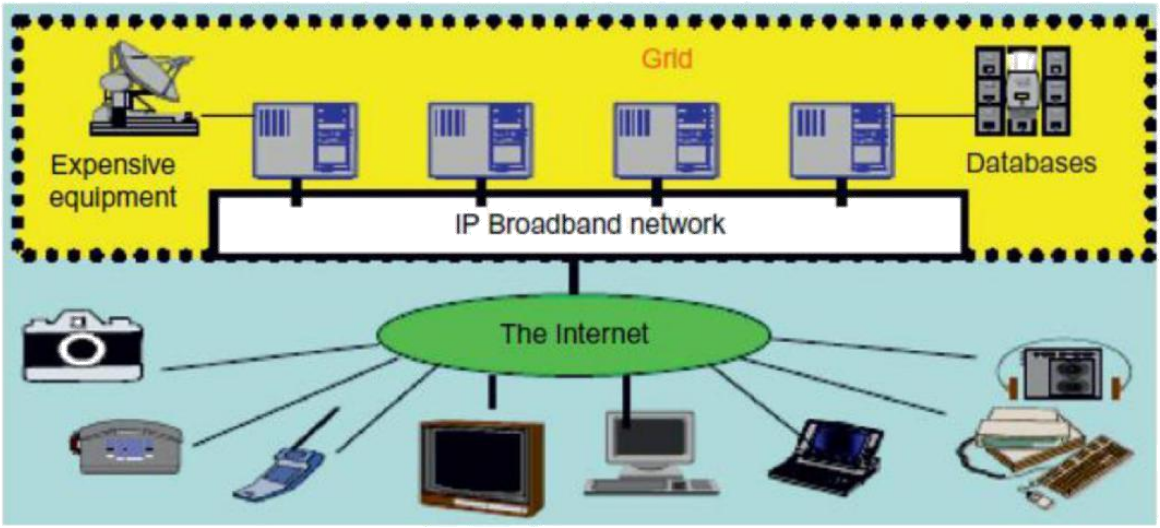
Computational Grids

Like an electric utility power grid, a computing grid offers an infrastructure that couples computers, software/middleware, special instruments, and people and sensors together. The grid is often constructed
across LAN, WAN, or Internet backbone networks at a regional, national, or global scale
Enterprises or organizations present grids as integrated computing resources. They can also be viewed as virtual platforms to support virtual organizations. The computers used in a grid are primarily workstations, servers, clusters, and supercomputers. Personal computers, laptops, and PDAs can be used as access devices to a grid system

Special instruments may be involved such as using the radio telescope in SETI@Home search of life in the galaxy and the austrophysics@Swineburne for pulsars. At the server end, the grid is a network.

Grid Families

Grid technology demands new distributed computing models, software/middleware support, network protocols, and hardware infrastructures. National grid projects are followed by industrial grid platform development by IBM, Microsoft, Sun, HP, Dell, Cisco, EMC, Platform Computing, and others. New grid service providers (GSPs) and new grid applications have emerged rapidly, similar to the growth of Internet and web services in the past two decades.



Computational grid or data grid providing computing utility, data, and information services through resource sharing and cooperation among participating organizations.

Table 1.4 Two Grid Computing Infrastructures and Representative Systems

Design Issues	Computational and Data Grids	P2P Grids
Grid Applications Reported	Distributed supercomputing, National Grid initiatives, etc.	Open grid with P2P flexibility, all resources from client machines
Representative Systems	TeraGrid built in US, ChinaGrid in China, and the e-Science grid built in UK	JXTA, FightAid@home, SETI@home
Development Lessons Learned	Restricted user groups, middleware bugs, protocols to acquire resources	Unreliable user-contributed resources, limited to a few apps

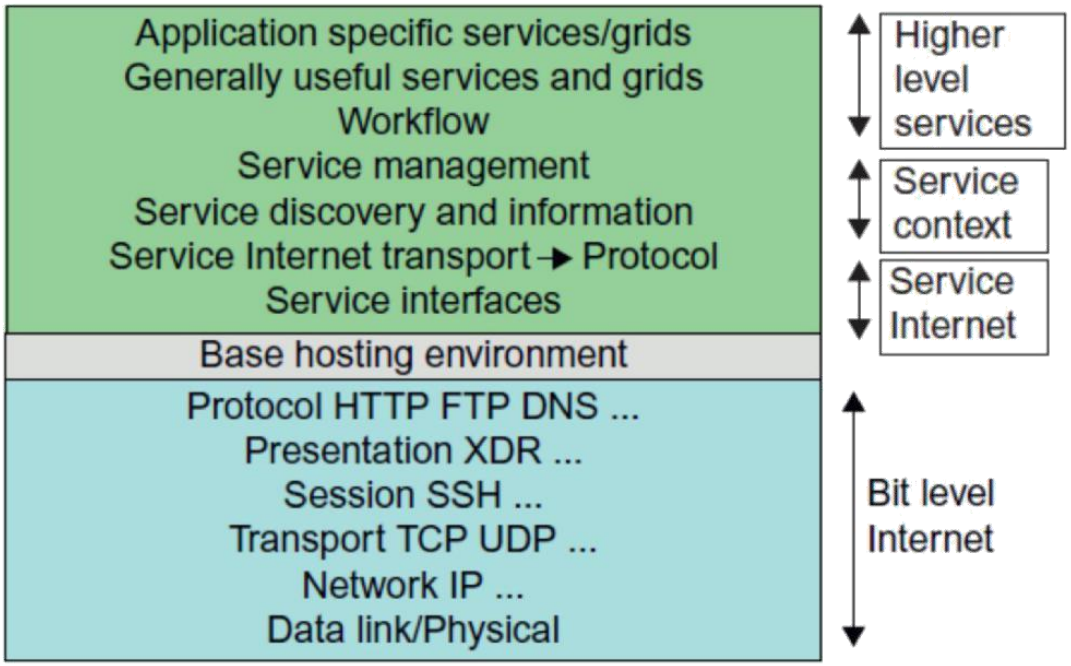
Explain in detail about service oriented architecture

In grids/web services, Java, and CORBA, an entity is, respectively, a service, a Java object, and a CORBA distributed object in a variety of languages. These architectures build on the traditional seven Open Systems Interconnection (OSI) layers that provide the base networking abstractions.

Layered Architecture for Web Services and Grids

The entity interfaces correspond to the Web Services Description Language (WSDL), Java method, and CORBA interface definition language (IDL) specifications in these example distributed systems. These interfaces are linked with customized, high-level communication systems: SOAP, RMI, and IIOP in the three examples. These communication systems support features including particular message patterns (such as Remote Procedure Call or RPC), fault recovery, and specialized routing the features in the Web Services Reliable Messaging (WSRM) framework mimic the OSI layer capability (as in TCP fault tolerance) modified to match the different abstractions (such as messages versus packets, virtualized addressing) at the entity levels. Security is a critical capability that either uses or reimplements the capabilities seen in concepts such as Internet Protocol Security (IPsec) and secure sockets in the OSI layers.

JNDI (Jini and Java Naming and Directory Interface) illustrating different approaches within the Java distributed object model. The CORBA Trading Service, UDDI (Universal Description, Discovery, and Integration), LDAP (Lightweight Directory Access Protocol), and ebXML (Electronic Business using eXtensible Markup Language) are other examples of discovery and information services described



Layered achitecture for web services and the grids.

Web Services and Tools

Loose coupling and support of heterogeneous implementations make services more attractive than distributed objects. corresponds to two choices of service architecture: web services or REST systems (these are further discussed in . Both web services and REST systems have very distinct approaches to building reliable interoperable systems. In web services, one aims to fully specify all aspects of the service and its environment.

In CORBA and Java, the distributed entities are linked with RPCs, and the simplest way to build composite applications is to view the entities as objects and use the traditional ways of linking them together. For Java, this could be as simple as writing a Java program with method calls replaced by Remote Method Invocation (RMI), while CORBA supports a similar model with a syntax reflecting the C++ style of its entity (object) interfaces.

The Evolution of SOA

service-oriented architecture (SOA) has evolved over the years. SOA applies to building grids, clouds, grids of clouds, clouds of grids, clouds of clouds (also known as interclouds), and systems of systems in general. A large number of sensors provide data-collection services, denoted in the figure as SS (sensor service). A sensor can be a ZigBee device, a Bluetooth device, a WiFi access point, a personal computer, a GPA, or a wireless phone, among other things. Raw data is collected by sensor services.

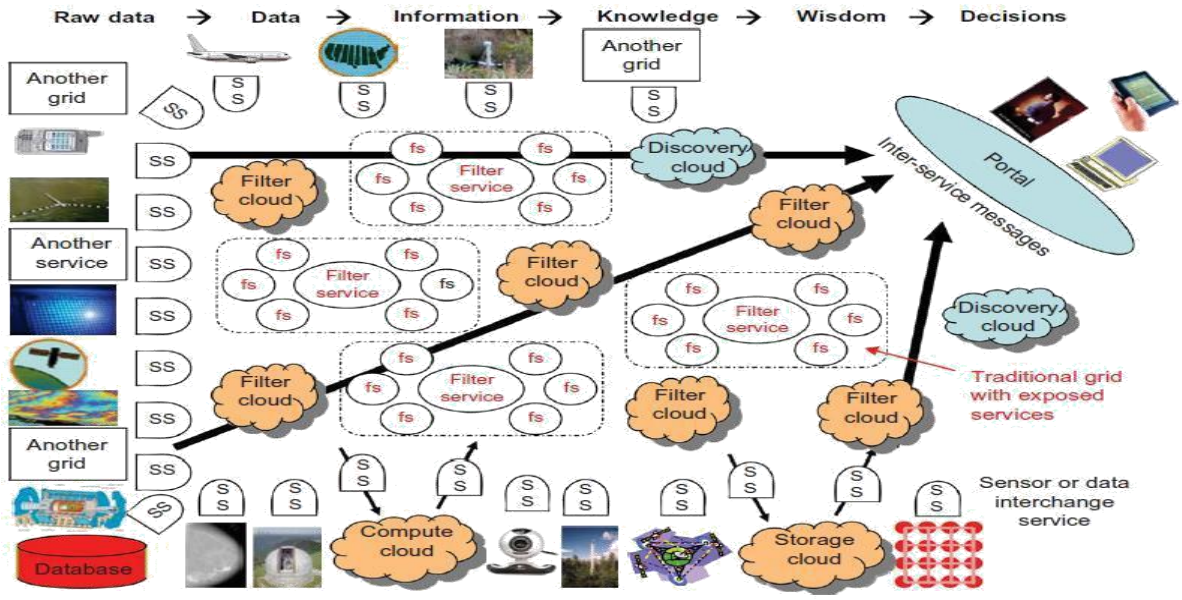
The evolution of SOA: grids of clouds and grids, where ―SS‖ refers to a sensor service and ―fs‖ to a filter or transforming service Most distributed systems require a web interface or portal. For raw data collected by a large number of sensors to be transformed into useful information or knowledge, the data stream may go through a sequence of compute, storage, filter, and discovery clouds. Finally, the inter-service messages converge at the portal, which is accessed by all users

Grids versus Clouds

The boundary between grids and clouds are getting blurred in recent years. For web services, workflow technologies are used to coordinate or orchestrate services with certain specifications used to define critical business process models such as two-phase transactions

In general, a grid system applies static resources, while a cloud emphasizes elastic resources. For some researchers, the differences between grids and clouds are limited only in dynamic resource allocation based on virtualization and autonomic computing. Thus one may end up building with

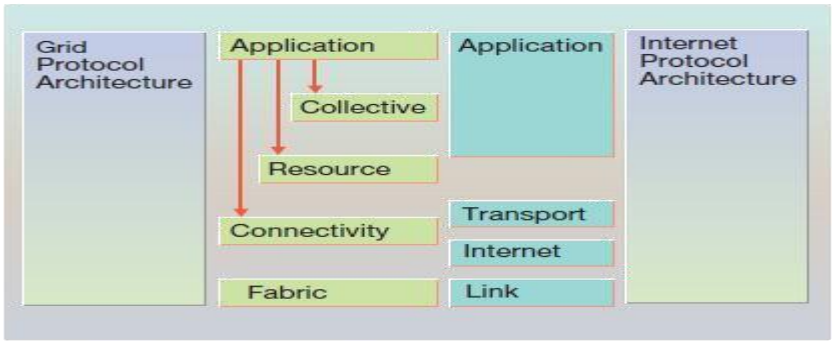
a system of systems: such as a cloud of clouds, a grid of clouds, or a cloud of grids, or inter-clouds as a basic SOA architecture.



9. Explain in detail about Grid Architecture and standards

New architecture model and technology has been developed for the establishment and management of cross-organizational resource sharing. This new architecture, called *grid architecture*, identifies the basic components of a grid system. The grid architecture defines the purpose and functions of its components, while indicating how

these components interact with one another.7 The main focus of the architecture is on interoperability among resource providers and users in order to establish the sharing relationships. This interoperability, in turn, necessitates common protocols at each layer of the architectural model, which leads to the definition of a grid protocol architecture as shown in Figure.



Reprinted with permission of Ian Foster

This protocol architecture defines common mechanisms, interfaces, schema, and protocols at each layer, by which users and resources can negotiate, establish, manage, and share resources. Figure 1 shows the component layers of the grid architecture and the capabilities of each layer. Each layer shares the behavior of the underlying component layers. The following describes the core features of each of these component layers, starting from the bottom of the stack and moving upward.

Fabric layer—The fabric layer defines the interface to local resources, which may be shared. This includes computational resources, data storage, networks, catalogs, software modules, and other system resources.

4. *Connectivity layer*—The connectivity layer defines the basic communication and authentication protocols required for grid-specific networkingservice transactions.

6. *Resource layer*—This layer uses the communication and security protocols (defined by the connectivity
layer) to control secure negotiation, initiation, monitoring, accounting, and payment for the sharing of functions of individual resources. The resource layer calls the fabric layer functions to access and control local resources. This layer only handles individual resources, ignoring global states and atomic actions across the resource collection
pool, which are the responsibility of the collective layer.

7. *Collective layer*—While the resource layer manages an individual resource, the collective layer is responsible
for all global resource management and interaction with collections of resources. This protocol layer implements a wide variety of sharing behaviors using a small number of resource-layer and connectivity-layer protocols.

8. *Application layer*—The application layer enables the use of resources in a grid environment through various collaboration and resource access protocols.

Thus far, our discussions have focused on the grid problem in the context of a virtual organization and the proposed grid computing architecture as a suggested solution to this problem. This architecture is designed for controlled resource sharing with improved interoperability among participants. In contrast, emerging architectures help the earlier-defined grid architecture quickly adapt to a wider (and strategically important) technology domain.

10. Explain in detail about Memory, Storage, and Wide-Area Networking

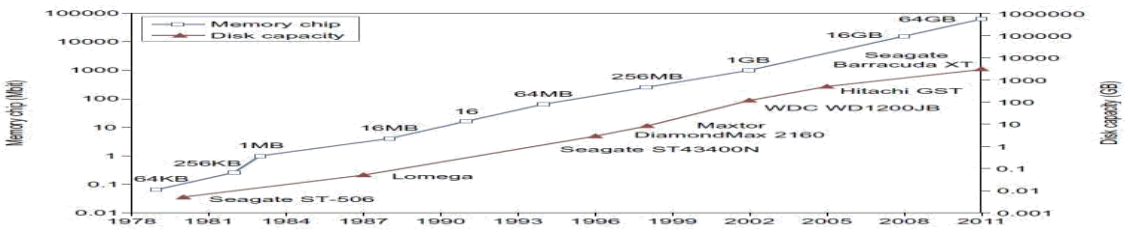
Memory Technology

Plots the growth of DRAM chip capacity from 16 KB in 1976 to 64 GB in 2011. This shows that memory chips have experienced a 4x increase in capacity every three years. Memory access time did not improve much in the past. In fact, the memory wall problem is getting worse as the processor gets faster. For hard drives, capacity increased from 260 MB in 1981 to 250 GB in 2004

The Seagate Barracuda XT hard drive reached 3 TB in 2011. This represents an approximately 10x increase in capacity every eight years. The capacity increase of disk arrays will be even greater in the years to come. Faster processor speed and larger memory capacity result in a wider gap between processors and memory

Disks and Storage Technology

Beyond 2011, disks or disk arrays have exceeded 3 TB in capacity. The lower curve in the disk storage growth in 7 orders of magnitude in 33 years. The rapid growth of flash memory and solid-state drives (SSDs) also impacts the future of HPC and HTC systems. The mortality rate of SSD is not bad at all. A typical SSD can handle 300,000 to 1 million write cycles per



Improvement in memory and disk technologies over 33 years. The Seagate Barracuda XT disk has a capacity

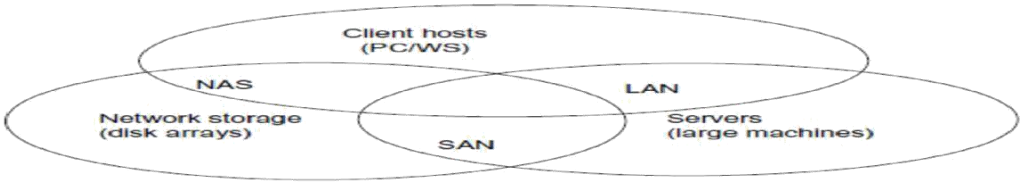
System-Area Interconnects

The nodes in small clusters are mostly interconnected by an Ethernet switch or a local area network (LAN). a LAN typically is used to connect client hosts to big servers. A storage area network (SAN) connects servers to network storage such as disk arrays. Network attached storage (NAS) connects client hosts directly to the disk arrays.

All three types of networks often appear in a large cluster built with commercial network components. If no large distributed storage is shared, a small cluster could be built with a multiport Gigabit Ethernet switch plus copper cables to link the end machines.

Wide-Area Networking

An increase factor of two per year on network performance was reported, which is faster than Moore's law on CPU speed doubling every 18 months. The implication is that more computers will be used concurrently in the future. High-bandwidth networking increases the capability of building massively distributed systems.



Three interconnection networks for connecting servers, client hosts, and storage devices; the LAN connects client hosts and servers, the SAN connects servers with disk arrays, and the NAS connects clients with large storage systems in the network environment.

Unit – 2 – Grid Services

Part – A

1. List the OGSA grid service interfaces?

Port Type	Operation
Grid service	Find service data, Termination time and Destroy
Notification source	Subscribe to notification topic
Notification sink	Deliver notification
Registry	Register service and Unregister service
Factory	Create service
Handle map	Find by handle

2. Define Endpoint References in WSRF

The WSRF service addressing mechanism is defined in the WS-addressing standard and uses a term called an endpoint reference (EPR), which is an XML document that contains various information about the service and resource. Specifically, the endpoint reference includes both the service address (URI) and resource identification called a key.

3. What are the specifications of WSRF

WSRF is actually a collection of four specifications (standards):

1. WS-ResourceProperties — specifies how resource properties are defined and accessed
2. WS-ResourceLifetime — specifies mechanisms to manage resource lifetimes
3. WS-ServiceGroup — specifies how to group services or WS-Resources together
4. WS-BaseFaults — specifies how to report faults

8. Define Globus 4 information services

Globus 4 information services collectively is called the Monitoring and Discovering System (MDS4 in GT 4) and consists of a set of three WSRF information components:

- Index service
- Trigger service • WebMDS

from which a framework can be constructed for collecting and using information. The three components are part of the full GT4 package.

5. Define WebMDS.

WebMDS (Web Monitoring and Discovering System) is a servlet that provides a Web-based interface to display XML-based information such as resource property information, and as such can be a front-end to index services.

6. Write about the strategies of replication

The strategies of replication can be classified into method types: dynamic and static. For the static method, the locations and number of replicas are determined in advance and will not be modified. Dynamic strategies can adjust locations and number of data replicas according to changes in conditions.

7. Define data grid? List the Grid Data Access Models

A data grid is a set of structured services that provides multiple services like the ability to access alter and transfer very large amounts of geographically separated data, especially for research and collaboration purposes.

1. Monadic model 2. Hierarchical model 3. Federation model 4. Hybrid model

8. Define grid data access Federation model

This model is better suited for designing a data grid with multiple sources of data supplies. Sometimes this model is also known as a mesh model. The data sources are distributed to many different locations. Although the data is shared, the data items are still owned and controlled by their original owners. According to predefined access policies, only authenticated users are authorized to request data from any data source.

9. Write about Parallel Data Transfer

parallel data transfer opens multiple data streams for passing subdivided segments of a file simultaneously. Although the speed of each stream is the same as in sequential streaming, the total time to move data in all streams can be significantly reduced compared to FTP transfer.

10. Define Striped Data Transfer

Striped data transfer, a data object is partitioned into a number of sections, and each section is placed in an individual site in a data grid. When a user requests this piece of data, a data stream is created for each site, and all the sections of data objects are transferred simultaneously.

11. Write about Monadic access model

This is a centralized data repository model. All the data is saved in a central data repository. When users want to access some data they have to submit requests directly to the central repository. No data is replicated for preserving data locality. This model is the simplest to implement for a small grid.

12. Explain grid data access Hierarchical model

This is suitable for building a large data grid which has only one large data access directory. The data may be transferred from the source to a second-level center. Then some data in the regional center is transferred to the third-level center. After being forwarded several times, specific data objects are accessed directly by users.

13. List the basic functionality requirements of grid service

- Discovery and brokering
- Metering and accounting
- Data sharing
- Deployment
- Virtual organizations
- Monitoring
- Policy

11. What are the security requirements of grid service

- Multiple security infrastructures
- Perimeter security solutions
- Authentication, Authorization, and Accounting
- Encryption
- Application and Network-Level Firewalls
- Certification

17. List the System Properties Requirements of grid service

- Fault tolerance
- Disaster recovery
- Self-healing capabilities
- Legacy application management
- Agreement-based interaction
- Grouping/aggregation of services

2 What are the objectives of OGSA?

18. Manage resources across distributed heterogeneous platforms
19. Support QoS-oriented Service Level Agreements (SLAs).
20. Provide a common base for autonomic management

- Define open, published interfaces and protocols for the interoperability of diverse resources. **17 .**

Define grid service instance

A grid service instance is a (potentially taransient) service that conforms to a set of conventions, expressed as WSDL interfaces, extensions, and behaviors, for such purposes as lifetime management, discovery of characteristics, and notification.

18. Define grid service handle (GSH)

A grid service handle (GSH) can be thought of as a permanent network pointer to a particular grid service instance. The GSH does not provide sufficient information to allow a client to access the service instance; the client needs to ―resolve‖ a GSH into a grid service reference (GSR).

19. Define grid service reference (GSR).

The GSR contains all the necessary information to access the service instance. The GSR is not a ―permanent‖ network pointer to the grid service instance because a GSR may become invalid for various reasons; for example, the grid service instance may be moved to a different server.

20. What is meant by grid service description

A grid service description describes how a client interacts with service instances. This description is independent of any particular instance. Within a WSDL document, the grid service description is embodied in the most derived of the instance, along with its associated portTypes bindings, messages, and types definitions.

iii) List the XML lifetime declaration properties

The three lifetime declaration properties are

ogsi:goodFrom
ogsi:goodUntil
ogsi:availableUntil

iv) Define Naming by Attributes in semantic name space

Attribute naming schemes associate various metadata with services and support retrieval via queries on attribute values. A registry implementing such a scheme allows service providers to publish the existence and properties of the services that they provide, so that service consumers can discover them.

23. Define naming by path in semantic name space

Path naming or directory schemes (as used, for example, in file systems) represent an alternative approach to attribute schemes for organizing services into a hierarchical name space that can be navigated.

Part – B

1. Explain in detail about Open Grid Services Architecture

The OGSA is an open source grid service standard jointly developed by academia and the IT industry under coordination of a working group in the Global Grid Forum (GGF). The standard was specifically developed for the emerging grid and cloud service communities. The OGSA is extended from web service concepts and technologies. The standard defines a common framework that allows businesses to build grid platforms across enterprises and business partners. The intent is to define the standards required for both open source and commercial software to support a global grid infrastructure

OGSA Framework

The OGSA was built on two basic software technologies: the Globus Toolkit widely adopted as a grid technology solution for scientific and technical computing, and web services (WS 2.0) as a popular standards-based framework for business and network applications. The OGSA is intended to support the creation, termination, management, and invocation of stateful, transient grid services via standard interfaces and conventions

OGSA Interfaces

The OGSA is centered on grid services. These services demand special well-defined application interfaces.

These interfaces provide resource discovery, dynamic service creation, lifetime management, notification, and manageability. These properties have significant implications regarding how a grid service is named, discovered, and managed

Port Type	Operation	Brief Description
Grid service	Find service data	Query a grid service instance, including the handle, reference, primary key, home handle map, interface information, and service-specific information. Extensible support for various query languages.
	Termination time	Set (and get) termination time for grid service instance.
	Destroy	Terminate grid service instance.
Notification source	Subscribe to notification topic	Subscribe to notifications of service events. Allow delivery via third-party messaging services.
Notification sink	Deliver notification	Carry out asynchronous delivery of notification messages.
Registry	Register service	Conduct soft-state registration of Grid Service Handles (GSHs).
	Unregister service	Unregister a GSH.
Factory	Create service	Create a new grid service instance.
Handle map	Find by handle	Return the Grid Service Reference (GSR) associated with the GSH.

Grid Service Handle

A GSH is a globally unique name that distinguishes a specific grid service instance from all others. The status of a grid service instance could be that it exists now or that it will exist in the future.

These instances carry no protocol or instance-specific addresses or supported protocol bindings. Instead, these information items are encapsulated along with all other instance-specific information. In order to interact with a specific service instance, a single abstraction is defined as a GSR.

Grid Service Migration

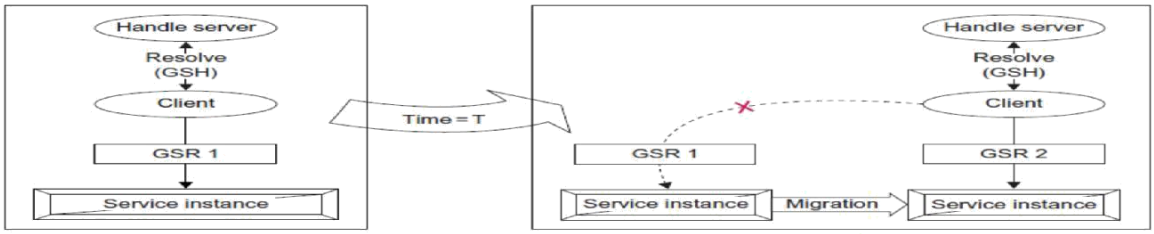
This is a mechanism for creating new services and specifying assertions regarding the lifetime of a service. The OGSA model defines a standard interface, known as a factor, to implement this reference. This creates a requested grid service with a specified interface and returns the GSH and initial GSR for the new service instance.

If the time period expires without having received a reaffirmed interest from a client, the service instance can be terminated on its own and release the associated resources accordingly

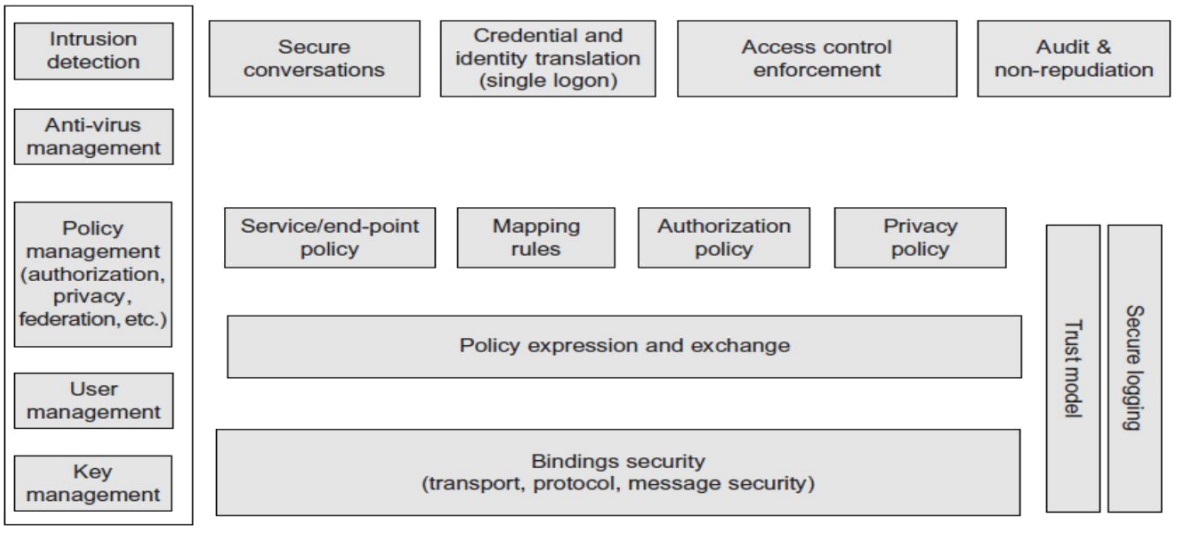
OGSA Security Models

The grid works in a heterogeneous distributed environment, which is essentially open to the general public. We must be able to detect intrusions or stop viruses from spreading by implementing secure conversations, single logon, access control, and auditing for nonrepudiation.

At the security policy and user levels, we want to apply a service or endpoint policy, resource mapping rules, authorized access of critical resources, and privacy protection. At the Public Key Infrastructure (PKI) service level, the OGSA demands security binding with the security protocol stack and bridging of certificate authorities (CAs), use of multiple trusted intermediaries, and so on.



A GSH resolving to a different GSR for a migrated service instance before (shown on the left) and after (on the right) the migration at time T.



The OGSA security model implemented at various protection levels.

6. Describe in detail about basic functionality requirements and System Properties Requirements

Basic functionality requirements

Discovery and brokering. Mechanisms are required for discovering and/or allocating services, data, and resources with desired properties. For example, clients need to discover network services before they are used, service brokers need to discover hardware and software availability, and service brokers must identify codes and platforms suitable for execution requested by the client

Metering and accounting. Applications and schemas for metering, auditing, and billing for IT infrastructure and management use cases. The metering function records the usage and duration, especially metering the usage of licenses. The auditing function audits usage and application profiles on machines, and the billing function bills the user based on metering.

Data sharing. Data sharing and data management are common as well as important grid applications. chanisms are required for accessing and managing data archives, for caching data and managing its consistency, and for indexing and discovering data and metadata.

Deployment. Data is deployed to the hosting environment that will execute the job (or made available in or via a high-performance infrastructure). Also, applications (executable) are migrated to the computer that will execute them

Virtual organizations (VOs). The need to support collaborative VOs introduces a need for mechanisms to support VO creation and management, including group membership services [58]. For the commercial data center use case [55], the grid creates a VO in a data center that provides IT resources to the job upon the customer's job request.

Monitoring. A global, cross-organizational view of resources and assets for project and fiscal planning, troubleshooting, and other purposes. The users want to monitor their applications running on the grid. Also, the resource or service owners need to surface certain states so that the user of those resources or services may manage the usage using the state information

Policy. An error and event policy guides self-controlling management, including failover and provisioning. It is important to be able to represent policy at multiple stages in hierarchical systems, with the goal of automating the enforcement of policies that might otherwise be implemented as organizational processes or managed manually

System Properties Requirements

Fault tolerance. Support is required for failover, load redistribution, and other techniques used to achieve fault tolerance. Fault tolerance is particularly important for long running queries that can potentially return large amounts of data, for dynamic scientific applications, and for commercial data center applications.

Disaster recovery. Disaster recovery is a critical capability for complex distributed grid infrastructures. For distributed systems, failure must be considered one of the natural behaviors and disaster recovery mechanisms must be considered an essential component of the design.

Self-healing capabilities of resources, services and systems are required. Significant manual effort should not be required to monitor, diagnose, and repair faults.

Legacy application management. Legacy applications are those that cannot be changed, but they are too aluable to give up or to complex to rewrite. Grid infrastructure has to be built around them so that they can continue to be used

Administration. Be able to ―codify‖ and ―automate‖ the normal practices used to administer the environment. The goal is that systems should be able to selforganize and self-describe to manage low-level configuration details based on higher-level configurations and management policies specified by administrators.

Agreement-based interaction. Some initiatives require agreement-based interactions capable of specifying and enacting agreements between clients and servers (not necessarily human) and then composing those agreements into higher-level end-user structures ***Grouping/aggregation of services***. The ability to instantiate (compose) services using some set of existing services is a key requirement. There are two main types of composition techniques: selection and aggregation. Selection involves choosing to use a particular service among many services with the same operational interface

3) Explain the following functionality requirements a) Security requirements

Security requirements

Grids also introduce a rich set of security requirements; some of these requirements are:

Multiple security infrastructures. Distributed operation implies a need to interoperate with and manage multiple security infrastructures. For example,for a commercial data center application, isolation of customers in the same commercial data center is a crucial requirement; the grid should provide not only access control but also performance isolation.

Perimeter security solutions. Many use cases require applications to be deployed on the other side of firewalls from the intended user clients. Intergrid collaboration often requires crossing institutional firewalls.

Authentication, Authorization, and Accounting. Obtaining application programs and deploying them into a grid system may require authentication/authorization. In the commercial data center use case, the commercial data center authenticates the customer and authorizes the submitted request when the customer submits a job request.

Encryption. The IT infrastructure and management use case requires encrypting of the communications, at least of the payload

Application and Network-Level Firewalls. This is a long-standing problem; it is made particularly difficult by the many different policies one is dealing with and the particularly harsh restrictions at international sites.

Certification. A trusted party certifies that a particular service has certain semantic behavior. For example, a company could establish a policy of only using e-commerce services certified by Yahoo

b) Resource Management Requirements

Resource management is another multilevel requirement, encompassing SLA negotiation, provisioning, and scheduling for a variety of resource types and activities

Provisioning. Computer processors, applications, licenses, storage, networks, and instruments are all grid resources that require provisioning. OGSA needs a framework that allows resource provisioning to be done in a uniform, consistent manner.

Resource virtualization. Dynamic provisioning implies a need for resource virtualization mechanisms that allow resources to be transitioned flexibly to different tasks as required; for example, when bringing more Web servers on line as demand exceeds a threshold..

Optimization of resource usage while meeting cost targets (i.e., dealing with finite resources). Mechanisms to manage conflicting demands from various organizations, groups, projects, and users and implement a fair sharing of resources and access to the grid

Transport management. For applications that require some form of real-time scheduling, it can be important to be able to schedule or provision bandwidth dynamically for data transfers or in support of the other data sharing applications. In many (if not all) commercial applications, reliable transport management is essential to obtain the end-to-end QoS required by the application

Management and monitoring. Support for the management and monitoring of resource usage and the detection of SLA or contract violations by all relevant parties. Also, conflict management is necessary;

Processor scavenging is an important tool that allows an enterprise or VO to use to aggregate computing power that would otherwise go to waste

Scheduling of service tasks. Long recognized as an important capability for any information processing system, scheduling becomes extremely important and difficult for distributed grid systems.

Load balancing. In many applications, it is necessary to make sure make sure deadlines are met or resources are used uniformly. These are both forms of load balancing that must be made possible by the underlying infrastructure

Advanced reservation. This functionality may be required in order to execute the application on reserved resources.

Notification and messaging. Notification and messaging are critical in most dynamic scientific problems.

Logging. It may be desirable to log processes such as obtaining/deploying application programs because, for example, the information might be used for accounting. This functionality is represented as ―metering and accounting‖

Workflow management. Many applications can be wrapped in scripts or processes that require licenses and other resources from multiple sources. Applications coordinate using the file system based on events

Pricing. Mechanisms for determining how to render appropriate bills to users of a grid.

4. Describe in detail about Practical view of OGSA/OGSI

OGSA aims at addressing standardization (for interoperability) by defining the basic framework of a grid application structure. Some of the mechanisms employed in the standards formulation of grid computing

The objectives of OGSA are

Manage resources across distributed heterogeneous platforms

Support QoS-oriented Service Level Agreements (SLAs). The topology of grids is often complex; the interactions between/among grid resources are almost invariably dynamic.

Provide a common base for autonomic management. A grid can contain a plethora of resources, along with an abundance of combinations of resource MPICH-G2: Grid-enabled message passing (Message Passing Interface)

_ CoG Kits, GridPort: Portal construction, based on N-tier architectures

_ Condor-G: workflow management

_ Legion: object models for grid computing

_ Cactus: Grid-aware numerical solver framework

Portals

_ N-tier architectures enabling thin clients, with middle tiers using grid functions _ Thin clients = web browsers

_ Middle tier = e.g., Java Server Pages, with Java CoG Kit, GPDK, GridPort utilities _ Bottom tier = various grid resources

_ Numerous applications and projects, e.g.,

_ Unicore, Gateway, Discover, Mississippi Computational Web Portal, NPACI Grid Port, Lattice Portal, Nimrod-G, Cactus, NASA IPG Launchpad, Grid Resource Broker

High-Throughput Computing and Condor

_ High-throughput computing

_ Processor cycles/day (week, month, year?) under nonideal circumstances

_ ―How many times can I run simulation X in a month using all available machines?‖

_ Condor converts collections of distributively owned workstations and dedicated clusters into a distributed high-throughput computing facility _ Emphasis on policy management and reliability

Object-Based Approaches

_ Grid-enabled CORBA

_ NASA Lewis, Rutgers, ANL, others

_ CORBA wrappers for grid protocols

_ Some initial successes

_ Legion

_ University of Virginia

_ Object models for grid components (e.g., ―vault‖ = storage, ―host‖ = computer) Cactus: Modular, portable framework for parallel, multidimensional simulations Construct codes by linking

_ Small core: management services

_ Selected modules: Numerical methods, grids and domain decomps, visualization and steering, etc.
_ Custom linking/configuration tools
_ Developed for astrophysics, but not astrophysics specific

Table 4.6 Proposed OGSA grid service interfaces*

Port type	Operation	Description
GridService	FindServiceData	Query a variety of information about the grid service instance, including basic introspection information (handle, reference, primary key, home handle map: terms to be defined), richer per-interface information, and service-specific information (e.g., service instances known to a registry). Extensible support for various query languages.
	SetTermination Time	Set (and get) termination time for grid service instance
	Destroy	Terminate grid service instance.
Notification-Source	SubscribeTo-NotificationTopic	Subscribe to notifications of service-related events, based on message type and interest statement. Allows for delivery via third-party messaging services.
Notification-Sink	Deliver Notification	Carry out asynchronous delivery of notification messages.
Registry	RegisterService UnregisterService	Conduct soft-state registration of grid service handles. Deregister a grid service handle.
Factory	CreateService	Create new grid service instance.
Handle Map	FindByHandle	Return grid service reference currently associated

There are two fundamental requirements for describing Web services based on the OGSI
1.The ability to describe interface inheritance—a basic concept with most of the distributed object systems.
7) The ability to describe additional information elements with the interface definitions.

2 Explain in detail about Detailed view of OGSA/OGSI

Provides a more detailed view of OGSI based on the OGSI specification itself. For a more comprehensive description of these concepts, the reader should consult the specification OGSI defines a component model that extends WSDL and XML schema definition to incorporate the concepts of
Stateful Web services
_ Extension of Web services interfaces _ Asynchronous notification of state change
_ References to instances of services _ Collections of service instances
_ Service state data that augment the constraint capabilities of XML schema definition **Setting the Context**

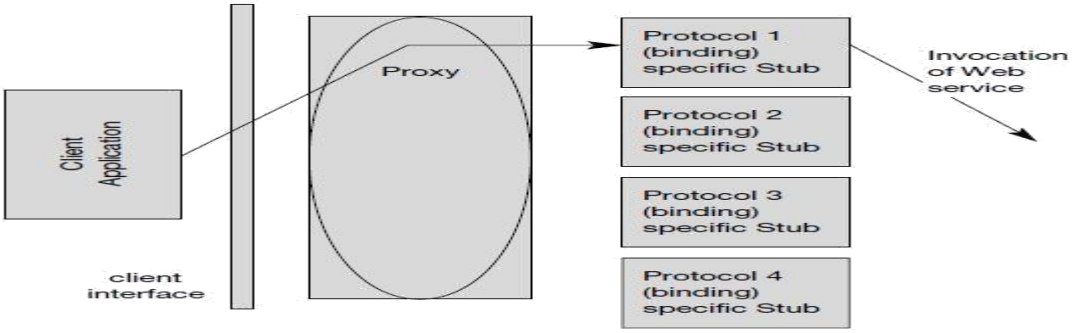
GGF calls OGSI the ―base for OGSA.‖ Specifically, there is a relationship between OGSI and distributed object systems and also a relationship between OGSI and the existing (and evolving) Web services framework

Relationship to Distributed Object Systems

Given grid service implementation is an addressable and potentially stateful instance that implements one or more interfaces described by WSDL portTypes. Grid service factories can be used to create instances implementing a given set of portType(s).

Client-Side Programming

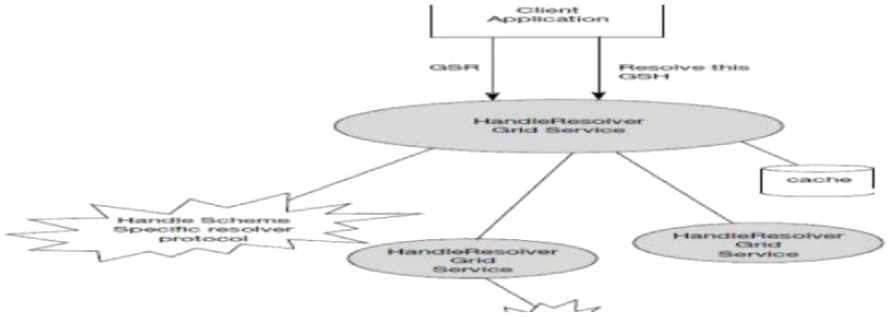
Patterns Another important issue is how OGSI interfaces are likely to be invoked from client applications. OGSI exploits an important component of the Web services framework: the use of WSDL to describe multiple protocol bindings, encoding styles, messaging styles (RPC versus document oriented), and so on, for a given Web service.



Possible client-side runtime architecture.

Client Use of Grid Service Handles and References

Client gains access to a grid service instance through grid service handles and grid service references. A grid service handle (GSH) can be thought of as a permanent network pointer to a particular grid service instance.



Resolving a GSH.

Relationship to Hosting Environment

OGSI does not dictate a particular service-provider-side implementation architecture. A variety of approaches are possible, ranging from implementing the grid service instance directly as an operating system process to a sophisticated server-side mponent model such as J2EE. In the former case, most or even all support for standard grid service behaviors (invocation, lifetime management, registration, etc.)

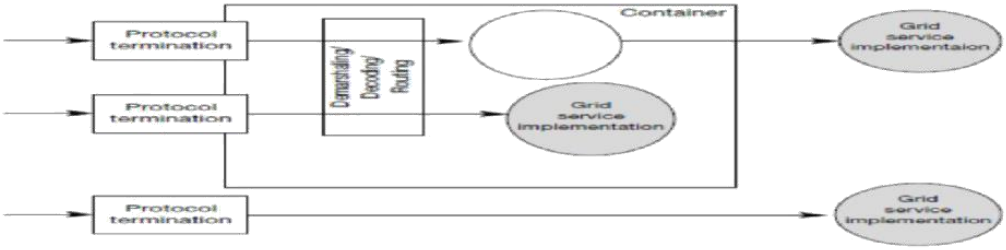


Figure 4.16 Two approaches to the implementation of argument demarshaling functions in a grid service hosting environment.

The Grid Service

The purpose of the OGSI document is to specify the (standardized) interfaces and behaviors that define a *grid service*

WSDL Extensions and Conventions

OGSI is based on Web services; in particular, it uses WSDL as the mechanism to describe the public interfaces of grid services.

Service Data

The approach to *stateful* Web services introduced in OGSI identified the need for a common mechanism to expose a service instance's state data to service requestors for query, update, and change notification.

Motivation and Comparison to JavaBean Properties

OGSI specification introduces the serviceData concept to provide a flexible, properties- style approach to accessing state data of a Web service. The serviceData concept is similar to the notion of a public instance variable or field in object-oriented programming languages such as Java, Smalltalk, and C++.

Extending portType with serviceData

ServiceData defines a newportType child element named serviceData, used to define serviceData elements, or SDEs, associated with that portType. These serviceData element definitions are referred to as serviceData declarations, or SDDs.

serviceDataValues.

Each service instance is associated with a collection of serviceData elements: those serviceData elements defined within the various portTypes that form the service's interface, and also, potentially, additional service

SDE Aggregation within a portType Interface Hierarchy

WSDL 1.2 has introduced the notion of multiple portType extension, and one can model that construct within the GWSDL namespace. A portType can extend zero or more other portTypes

Dynamic serviceData Elements

Although many serviceData elements are most naturally defined in a service's interface definition, situations can

arise in which it is useful to add or move serviceData elements dynamically to or from an instance.

6. Short Notes on

a) Core Grid Service Properties

Service Description and Service Instance

One can distinguish in OGSI between the *description* of a grid service and an *instance* of a grid service:

A *grid service description* describes how a client interacts with service instances.

This description is independent of any particular instance. Within a WSDL document, the grid service description is embodied in the most derived portType

A grid service description may be simultaneously used by any number of *grid service instances,* each of which

_ Embodies some state with which the service description describes how to
interact _ Has one or more grid service handles

_ Has one or more grid service references to it

Modeling Time in OGSI

The need arises at various points throughout this specification to represent time that is meaningful to multiple parties in the distributed Grid.

The GMT global time standard is assumed for grid services, allowing operations to refer unambiguously to absolute times. However, assuming the GMT time standard to represent time does *not* imply any particular level of clock synchronization between clients and services in the grid. In fact, no specific accuracy of synchronization is specified or expected by OGSI, as this is a service-quality issue

XML Element Lifetime Declaration Properties

Service Data elements may represent instantaneous observations of the dynamic state of a service instance, it is critical that consumers of serviceData be able to understand the valid lifetimes of these observations.

The three lifetime declaration properties are:

1.ogsi:goodFrom. Declares the time from which the content of the element is said to be valid. This is typically the time at which the value was created.

11) ogsi:goodUntil. Declares the time until which the content of the element is said to be valid. This property must be greater than or equal to the goodFrom time

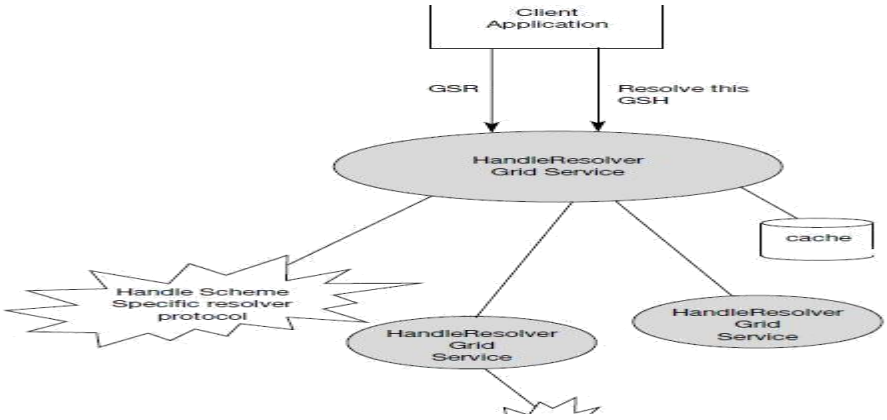
3.ogsi:availableUntil. Declares the time until which this element itself is expected to be available, perhaps with updated values. Prior to this time, a client should be able to obtain an updated copy of this element

b) Grid Service Handles and Grid Service References

Client gains access to a grid service instance through grid service handles and grid service references. A grid service handle (GSH) can be thought of as a permanent network pointer to a particular grid service instance.

Resolving a GSH.

The client resolves a GSH into a GSR by invoking a HandleResolver grid service instance identified by some out-of-band mechanism. The HandleResolver can use various means to do the resolution



7. Explain in detail about Data-Intensive Grid Service Models

Applications in the grid are normally grouped into two categories: computation-intensive and dataintensive. For data-intensive applications, we may have to deal with massive amounts of data. For example, the data produced annually by a Large Hadron Collider may exceed several petabytes (1015 bytes). The grid system must be specially designed to discover, transfer, and manipulate these massive data sets. Transferring massive data sets is a time-consuming task. Efficient data management demands low-cost storage and high-speed data movement

Data Replication and Unified Namespace

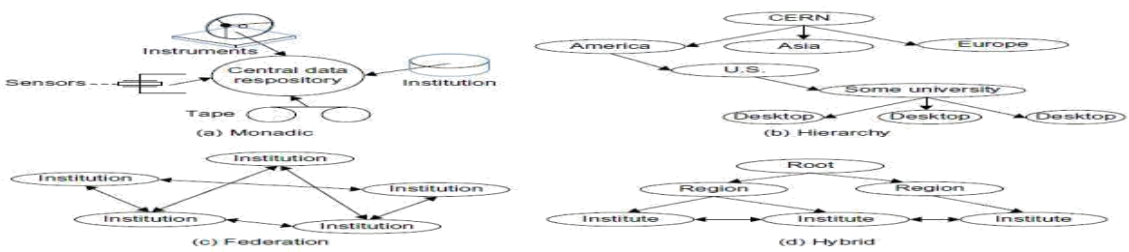
This data access method is also known as caching, which is often applied to enhance data efficiency in a grid environment. By replicating the same data blocks and scattering them in multiple regions of a grid, users can access the same data with locality of references. Replication strategies determine when and where to create a replica of the data. The factors to consider include data demand, network conditions, and transfer cost

Grid Data Access Models

Multiple participants may want to share the same data collection. To retrieve any piece of data, we need a grid with a unique global namespace. Similarly, we desire to have unique file names. To achieve these, we have to resolve inconsistencies among multiple data objects bearing the same name Monadic model: This is a centralized data repository model, All the data is saved in a central data repository. When users want to access some data they have to submit requests directly to the central repository.

Hierarchical model: The hierarchical model,is suitable for building a large data grid which has only one large data access directory. The data may be transferred from the source to a second-level center.

Federation model: This data access model is better suited for designing a data grid with multiple sources of data supplies. Sometimes this model is also known as a mesh model. **Hybrid model**: This data access model. The model combines the best features of the hierarchical and mesh models. Traditional data transfer technology, such as FTP, applies for networks with lower bandwidth



Four architectural models for building a data grid.

Parallel versus Striped Data Transfers

Compared with traditional FTP data transfer, parallel data transfer opens multiple data streams for passing subdivided segments of a file simultaneously. Although the speed of each stream is the same as in sequential streaming, the total time to move data in all streams can be significantly reduced compared to FTP transfer.

2 Short notes on OGSA Service

a) Metering Service

Different grid deployments may integrate different services and resources and feature different underlying economic motivations and models; however, regardless of these differences, it is a quasiuniversal requirement that resource utilization can be monitored, whether for purposes of cost allocation (i.e., charge back), capacity and trend analysis, dynamic provisioning, grid-service pricing, fraud and intrusion detection, and/or billing.

A grid service may consume multiple resources and a resource may be shared by multiple service instances. Ultimately, the sharing of underlying resources is managed by middleware and operating systems.

A metering interface provides access to a standard description of such aggregated data (metering serviceData). A key parameter is the time window over which measurements are aggregated. In commercial Unix systems, measurements are aggregated at administrator-defined intervals (chronological entry), usually daily, primarily for the purpose of accounting.

Several use cases require metering systems that support multitier, end-to-end flows involving multiple services. An OGSA metering service must be able to meter the resource consumption of configurable classes of these types of flows executing on widely distributed, loosely coupled server, storage, and network resources. Configurable classes should support, for example, a departmental charge-back scenario where incoming requests and their subsequent flows are partitioned into account classes determined by the department providing the service.

b) Service Groups and Discovery Services

GSHs and GSRs together realize a two-level naming scheme, with HandleResolver services mapping from handles to references; however, GSHs are not intended to contain semantic information and indeed may be viewed for most purposes as opaque. Thus, other entities (both humans and applications) need other means for

discovering services with particular properties, whether relating to interface, function, availability, location, policy

Attribute naming schemes associate various metadata with services and support retrieval via queries on attribute values. A registry implementing such a scheme allows service providers to publish the existence and properties of the services that they provide, so that service consumers can discover them

A ServiceGroup is a collection of entries, where each entry is a grid service implementing the rviceGroupEntry interface. The ServiceGroup interface also extends the GridService interface

It is also envisioned that many registries will inherit and implement the notificationSource interface so as to facilitate client subscription to register state changes

Path naming **or** ***directory*** schemes (as used, for example, in file systems) represent an alternative approach to attribute schemes for organizing services into a hierarchical name space that can be navigated. The two pproaches can be combined, as in LDAP.

c) Rating Service

A rating interface needs to address two types of behaviors. Once the metered information is available, it has to be translated into financial terms. That is, for each unit of usage, a price has to be associated with it. This step is accomplished by the rating interfaces, which provide operations that take the metered information and a rating package as input and output the usage in terms of chargeable amounts.

For example,
a commercial UNIX system indicates that 10 hours of prime-time resource and 10 hours on nonprime-time resource are consumed, and the rating package indicates that each hour of prime-time resource is priced at 2 dollars and each hour of nonprime- time resource is priced at 1 dollar, a rating service will apply the pricing indicated in the rating package

Furthermore, when a business service is developed, a rating service is used to aggregate the costs of the components used to deliver the service, so that the service owner can determine the pricing, terms, and conditions under which the service will be offered to subscribe

d) Other Data Services

A variety of higher-level data interfaces can and must be defined on top of the
base data interfaces, to address functions such as: _ Data access and movement

_ Data replication and caching
_ Data and schema mediation
_ Metadata management and looking

Data Replication. Data replication can be important as a means of meeting performance objectives by allowing local computer resources to have access to local data. Although closely related to caching (indeed, a ―replica store‖ and a ―cache‖may differ only in their policies), replicas may provide different interfaces

Data Caching. In order to improve performance of access to remote data items, caching services will be employed. At the minimum, caching services for traditional flat file data will be employed. Caching of other data types, such as views on RDBMS data, streaming data, and application binaries, are also envisioned

Consistency—Is the data in the cache the same as in the source? If not, what is the coherence window? Different applications have very different requirements. _ Cache invalidation protocols—How and when is cached data invalidated? _ Write through or write back? When are writes to the cache committed back to the original data source?

Security—How will access control to cached items be handled? Will access control enforcement be delegated to the cache, or will access control be somehow enforced by the original data source? _ Integrity of cached data—Is the cached data kept in memory or on disk? How is it protected from unauthorized access? Is it encrypted

Schema Transformation. Schema transformation interfaces support the transformation of data from one schema to another. For example, XML transformations as specified in XSLT.

9. Explain in detail about Open Grid Services Infrastructure and Distributed Logging

The OGSI defines fundamental mechanisms on which OGSA is constructed. These mechanisms address issues relating to the creation, naming, management, and exchange of information among entities called grid services. The following list recaps the key OGSI features and briefly discusses their relevance to OGSA.

Grid Service descriptions and instances. OGSI introduces the twin concepts of the grid service description and grid service instance as organizing principles of distributed systems.

Grid Service descriptions and instances. OGSI introduces the twin concepts of the grid service description and grid service instance as organizing principles of distributed systems.

Naming and name resolution. OGSI defines a two-level naming scheme for grid service instances based on abstract, long-lived *grid service handles* that can be mapped by HandleMapper services to concrete but potentially lesslong- lived *grid service references.*

Fault model. OGSI defines a common approach for conveying fault information from operations.

Life cycle. OGSI defines mechanisms for managing the life cycle of a grid service instance, including both explicit destruction and soft-state lifetime management functions for grid service instances, and grid service factories that can be used to create instances implementing specified interfaces

Service groups. OGSI defines a means of organizing groups of service
instances. **Distributed Logging**

Distributed logging can be viewed as a typical messaging application in which *message producers* generate *log artifacts,* (atomic expressions of diagnostic information) that may or may not be used at a later time by other independent *message consumers.* OGSA-based logging can leverage the notification mechanism available in
OGSI as the transport for messages.

Logging services provide the extensions needed to deal with the following issues: *Decoupling.* The logical separation of logging artifact creation from logging artifact consumption. The ultimate usage of the data (e.g., logging, tracing, management) is determined by the message consumer

Transformation and common representation. Logging packages commonly annotate the data that they generate with useful common information such as category, priority, time stamp, and location

Filtering and aggregation. The amount of logging data generated can be large, whereas the amount of data actually consumed can be small. Therefore, it can be desirable to have a mechanism for controlling the amount of data generated and for filtering out what is actually kept and where.

Configurable persistency. Depending on consumer needs, data may have different durability characteristics. For example, in a real-time monitoring application, data may become irrelevant quickly, but be needed as soon as it is generated; data for an auditing program may be needed months or even years after it was generated.

Consumption patterns. Consumption patterns differ according to the needs of the consumer application. For example, a real-time monitoring application needs to be notified whenever a particular event occurs, whereas a postmortem problem determination program queries historical data, trying to find known patterns.

12) Short notes on

1. Job Agreement Service

The job agreement service is created by the agreement factory service with a set of job terms, including command line, resource requirements, execution environment, data staging, job control, scheduler directives, and accounting and notification term.

The job agreement service provides an interface for placing jobs on a resource manager (i.e., representing a machine or a cluster), and for interacting with the job once it has been dispatched to the resource manager. The job agreement service provides basic matchmaking capabilities between the requirements of the job and the underlying resource manager available for running the job.

The interfaces provided by the job agreement service are:

_ Manageability interface

_ Supported job terms: defines a set of service data used to publish the job terms supported by this job service, including the job definition (command line and application name), resource requirements, execution ironment, data staging, job control, scheduler directives, and accounting and notification terms.

_ Workload status: total number of jobs, statuses such as number of jobs running or pending and suspended jobs.

_ Job control: control the job after it has been instantiated. This would include the ability to suspend/resume, checkpoint, and kill the job.

b) Reservation Agreement Service

The reservation agreement service is created by the agreement factory service with a set of terms including time duration, resource requirement specification, and authorized user/project agreement terms. The reservation agreement service allows end users or a job agreement service to reserve resources under the control of a resource manager to guarantee their availability to run a job. The service allows reservations on any type of resource (e.g., hosts, software licenses, or network bandwidth). Reservations can be specific (e.g., provide access to host ―A‖ from noon to 5 PM), or more general (e.g., provide access to 16 Linux cpus on Sunday).

The reservation service makes use of information about the existing resource managers available and any policies that might be defined at the VO level, and will make use of a logging service to log reservations. It will use the resource manager adapter interfaces to make reservations and to delete existing reservations.

c) Base Data Services

OGSA data interfaces are intended to enable a service-oriented treatment of data so that data can be treated in the same way as other resources within the Web/grid services architecture

Four *base data interfaces* (WSDL portTypes) can be used to implement a variety of different data service behaviors:

13) ***DataDescription*** defines OGSI service data elements representing key parameters of the data virtualization encapsulated by the data service.

14) ***DataAccess*** provides operations to access and/or modify the contents of the data virtualization encapsulated by the data service.

15) ***DataFactory*** provides an operation to create a new data service with a data virtualization derived from the data virtualization of the parent (factory) data service.

16)***DataManagement*** provides operations to monitor and manage the data service's data virtualization, including (depending on the implementation) the data sources (such as database management systems) that underlie the data service.

Unit – 3 - Virtualization

Part – A

1. Define private cloud.

The *private cloud* is built within the domain of an intranet owned by a single organization. Therefore, they are client owned and managed. Their access is limited to the owning clients and their partners. Their deployment was not meant to sell capacity over the Internet through publicly accessible interfaces. Private clouds give local users a flexible and agile private infrastructure to run service workloads within their administrative domains.

2. Define public cloud.

A *public cloud* is built over the Internet, which can be accessed by any user who has paid for the service. Public clouds are owned by service providers. They are accessed by subscription. Many companies have built public clouds, namely Google App Engine, Amazon AWS, Microsoft Azure, IBM Blue Cloud, and Salesforce Force.com. These are commercial providers that offer a publicly accessible remote interface for creating and managing VM instances within their proprietary infrastructure.

3. Define hybrid cloud.

A *hybrid cloud* is built with both public and private clouds, Private clouds can also support a *hybrid cloud* model by supplementing local infrastructure with computing capacity from an external public cloud. For example, the *research compute cloud* (RC2) is a private cloud built by IBM.

4. List the essential characteristics of cloud computing

1. On-demand capabilities 2. Broad network access 3. Resource pooling
4. Rapid elasticity 5. Measured service

JJ. List the design objectives of cloud computing.

Shifting Computing from Desktops to Datacenters
Service Provisioning and Cloud Economics
Scalability in Performance
Data Privacy Protection.

- High Quality of Cloud Services.

6. Define anything-as-a-service.

Providing services to the client on the basis on meeting their demands at some pay per use cost such as data storage as a service, network as a service, communication as a service etc. it is generally denoted as anything as a service (XaaS).

7. What is mean by SaaS?

The software as a service refers to browser initiated application software over thousands of paid customer. The SaaS model applies to business process industry application, consumer relationship management (CRM), Enterprise resource Planning (ERP), Human Resources (HR) and collaborative application.

8. What is mean by IaaS?

The Infrastructure as a Service model puts together the infrastructure demanded by the user namely servers, storage, network and the data center fabric. The user can deploy and run on multiple VM's running guest OS on specific application**.**

9. What is PaaS?

The Platform as a Service model enables the user to deploy user built applications onto a virtualized cloud platform. It includes middleware, database, development tools and some runtime support such as web2.0 and java. It includes both hardware and software integrated with specific programming interface.

10. What is mean by Virtualization?

Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in the same hardware machine. The purpose of a VM is to enhance resource sharing by many users and improve computer performance in terms of resource utilization and application flexibility.

11. Define virtual machine monitor.

A traditional computer runs with a host operating system specially tailored for its hardware architecture, After virtualization, different user applications managed by their own operating systems (guest OS) can run on the same hardware, independent of the host OS. This is often done by adding additional software, called a virtualization layer. This virtualization layer is known as hypervisor or virtual machine monitor (VMM).

12. List the requirements of VMM.

999. VMM should provide an environment for programs which is essentially identical to the original machine.

2 • Programs run in this environment should show, at worst, only minor decreases in speed.

III. VMM should be in complete control of the system resources. Any program run under a VMM

3 should exhibit a function identical to that which it runs on the original machine directly.

The guest OS, which has control ability, is called Domain 0, and the others are called Domain U. Domain 0 is a privileged guest OS of Xen. It is first loaded when Xen boots without any file system drivers being available. Domain 0 is designed to access hardware directly and manage devices.

14. What are the responsibilities of VMM?

- The VMM is responsible for allocating hardware resources for programs.
- It is not possible for a program to access any resource not explicitly allocated to it.

A It is possible under certain circumstances for a VMM to regain control of resources already

14) allocated.

15. Define CPU virtualization.

CPU architecture is virtualizable if it supports the ability to run the VM's privileged and unprivileged instructions in the CPU's user mode while the VMM runs in supervisor mode. When the privileged instructions including control- and behavior-sensitive instructions of a VM are executed, they are trapped in the VMM. In this case, the VMM acts as a unified mediator for hardware access from different VMs to guarantee the correctness and stability of the whole system.

16. Define memory virtualization.

Virtual memory virtualization is similar to the virtual memory support provided by modern operating systems. In a traditional execution environment, the operating system maintains mappings of virtual memory to machine memory using page tables, which is a one-stage mapping from virtual memory to machine memory. All modern x86 CPUs include a memory management unit (MMU) and a translation look aside buffer (TLB) to optimize virtual memory performance.

17. What is mean by I/O virtualization?

I/O virtualization involves managing the routing of I/O requests between virtual devices and the shared physical hardware. There are three ways to implement I/O virtualization:

full device emulation, Full device emulation is the first approach for I/O virtualization

para-virtualization

16) • direct I/O.

18. Distinguish the physical and virtual cluster. (Jan.2014)

A physical cluster is a collection of servers (physical machines) connected by a physical network such as a LAN. Virtual clusters have different properties and potential applications. There are three critical design issues of virtual clusters: live migration of virtual machines (VMs), memory and file migrations, and dynamic deployment of virtual clusters.

19. What is memory migration?

Moving the memory instance of a VM from one physical host to another can be approached in any number of ways. Memory migration can be in a range of hundreds of megabytes to a few gigabytes in a typical system today, and it needs to be done in an efficient manner. The Internet Suspend-Resume (ISR) technique exploits temporal locality as memory states are likely to have considerable overlap in the suspended and the resumed instances of a VM.

20. What is mean by host based virtualization?

An alternative VM architecture is to install a virtualization layer on top of the host OS. This host OS is still responsible for managing the hardware. The guest OSes are installed and run on top of the virtualization layer. Dedicated applications may run on the VMs. Certainly, some other applications can also run with the host OS directly.

21. Define KVM.

Kernel-Based VM:- This is a Linux para-virtualization system—a part of the Linux version 2.6.20 kernel. Memory management and scheduling activities are carried out by the existing Linux kernel. The KVM does the rest, which makes it simpler than the hypervisor that controls the entire machine. KVM is a hardware-assisted para-virtualization tool, which improves performance and supports unmodified guest OSes such as Windows, Linux, Solaris, and other UNIX variants

Part – B

J. Explain the cloud computing service and deployment models of cloud computing

Cloud computing service

Infrastructure as a Service (IaaS)

The infrastructure layer builds on the virtualization layer by offering the virtual machines as a service to users. Instead of purchasing servers or even hosted services, IaaS customers can create and remove virtual machines and network them together at will. Clients are billed for infrastructure services based on what resources are consumed. This eliminates the need to procure and operate physical servers, data storage systems, or networking resources.

Platform as a Service (PaaS)

The platform layer rests on the infrastructure layer's virtual machines. At this layer customers do not manage their virtual machines; they merely create applications within an existing API or programing language. There is no need to manage an operating system, let alone the underlying hardware and virtualization layers. Clients merely create their own programs which are hosted by the platform services they are paying for.

Software as a Service (SaaS)

Services at the software level consist of complete applications that do not require development. Such applications can be email, customer relationship management, and other office productivity applications. Enterprise services can be billed monthly or by usage, while software as service offered directly to consumers, such as email, is often provided for free.

Deployment models of cloud computing

The Private Cloud

This model doesn't bring much in terms of cost efficiency: it is comparable to buying, building and managing your own infrastructure. Still, it brings in tremendous value from a security point of view. During their initial adaptation to the cloud, many organizations face challenges and have concerns related to data security. These concerns are taken care of by this model, in which hosting is built and maintained for a specific client. The infrastructure required for hosting can be on-premises or at a third-party location. Security concerns are addressed through secure-access VPN or by the physical location within the client's firewall system.

Public Cloud

The public cloud deployment model represents true cloud hosting. In this deployment model, services and infrastructure are provided to various clients. Google is an example of a public cloud. This service can be provided by a vendor free of charge or on the basis of a pay-per-user license policy. This model is best suited for business requirements wherein it is required to manage load spikes, host SaaS applications, utilize interim infrastructure for developing and testing applications, and manage applications which are consumed by many users that would otherwise require large investment in infrastructure from businesses.

Hybrid Cloud

This deployment model helps businesses to take advantage of secured applications and data

hosting on a private cloud, while still enjoying cost benefits by keeping shared data and applications on the public cloud. This model is also used for handling cloud bursting, which refers to a scenario where the existing private cloud infrastructure is not able to handle load spikes and requires a fallback option to support the load. Hence, the cloud migrates workloads between public and private hosting without any inconvenience to the users. Many PaaS deployments expose their APIs, which can be further integrated with internal applications or applications hosted on a private cloud, while still maintaining the security aspects. Microsoft Azure and Force.com are two examples of this model.

Community Cloud

In the community deployment model, the cloud infrastructure is shared by several organizations with the same policy and compliance considerations. This helps to further reduce costs as compared to a private cloud, as it is shared by larger group.Various state-level government departments requiring access to the same data relating to the local population or information related to infrastructure, such as hospitals, roads, electrical stations, etc., can utilize a community cloud to manage applications and data.Cloud computing is not a ―silver–bullet‖ technology; hence, investment in any deployment model should be made based on business requirements, the criticality of the application and the level of support required.

2. a. Compare public cloud with private cloud

A private cloud hosting solution, also known as an internal or enterprise cloud, resides on company's intranet or hosted data center where all of your data is protected behind a firewall. This can be a great option for companies who already have expensive data centers because they can use their current infrastructure. However, the main drawback people see with a private cloud is that all management, maintenance and updating of data centers is the responsibility of the company. Over time, it's expected that your servers will need to be replaced, which can get very expensive. On the other hand, private clouds offer an increased level of security and they share very few, if any, resources with other organizations.

The main differentiator between public and private clouds is that you aren't responsible for any of the management of a public cloud hosting solution. Your data is stored in the provider's data center and the provider is responsible for the management and maintenance of the data center. This type of cloud environment is appealing to many companies because it reduces lead times in testing and deploying new products. However, the drawback is that many companies feel security could be lacking with a public cloud. Even though you don't control the security of a public cloud, all of your data remains separate from others and security breaches of public clouds are rare.

2 b. Pros and Cons of cloud computing

Pros:

17) **Cloud Computing has lower software costs.** With Cloud Computing a lot of software is paid on a monthly basis which when compared to buying the software in the beginning, software through Cloud Computing is often a fraction of the cost.
18) Eventually your company may want to migrate to a new operating system, the associated **costs to migrate to a new operating system, is**often **less** than in a traditional server environment.
19) **Centralized data**- Another key benefit with Cloud Computing is having all the data (which could be **for multiple branch offices** or project sites) in a **single location** "the Cloud".
20) **Access from anywhere**- never leave another important document back at the office. With Cloud computing and an Internet connection, your data are always nearby, even if you are on the other side of the world.
21) **Internet connection** is a **required** for Cloud Computing. You must have an Internet connection to access your data.

Cons

1. Internet Connection Quality & Cloud Computing

Low Bandwidth -If you can only get low bandwidth Internet (like dial-up) then you **should not**

consider using Cloud Computing. Bandwidth is commonly referred to as "how fast a connection is" or what the "speed" of your Internet is. The bandwidth to download data may not be the same as it is to send data.

Unreliable Internet connection -If you can get high speed Internet but it is unreliable (meaning your connection drops frequently and/or can be down for long periods at a time), depending on your business and how these outages will impact your operations, Cloud Computing may not be for you (or you may need to look into a more reliable and/or additional Internet connection).

Your company will **still need a Disaster Recovery Plan**, and if you have one now, it will need to be revised to address the changes for when you are using Cloud Computing.

J. Compare virtual and physical clusters. Explain how resource management done for virtual clusters.

A physical cluster is a collection of servers (physical machines) connected by a physical network such as a LAN. Virtual clusters have different properties and potential applications. There are three critical design issues of virtual clusters: live migration of virtual machines (VMs), memory and file migrations, and dynamic deployment of virtual clusters

Virtual clusters are built with VMs installed at distributed servers from one or more physical clusters. The VMs in a virtual cluster are interconnected logically by a virtual network across several physical networks. Below figure illustrates the concepts of virtual clusters and physical clusters. Each virtual cluster is formed with physical machines or a VM hosted by multiple physical clusters. The virtual cluster boundaries are shown as distinct boundaries.

The provisioning of VMs to a virtual cluster is done dynamically to have the following interesting properties:

The virtual cluster nodes can be either physical or virtual machines. Multiple VMs running with different OSes can be deployed on the same physical node.

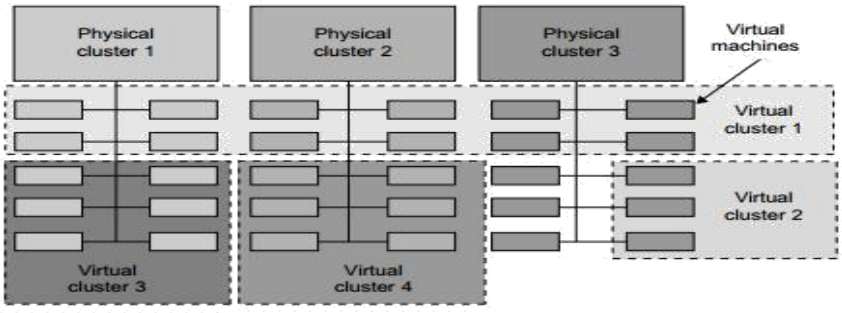
A VM runs with a guest OS, which is often different from the host OS, that manages the resources in the physical machine, where the VM is implemented.

The purpose of using VMs is to consolidate multiple functionalities on the same server. This will greatly enhance server utilization and application flexibility.

VMs can be colonized (replicated) in multiple servers for the purpose of promoting distributed parallelism, fault tolerance, and disaster recovery.

The size (number of nodes) of a virtual cluster can grow or shrink dynamically, similar to the way an overlay network varies in size in a peer-to-peer (P2P) network.

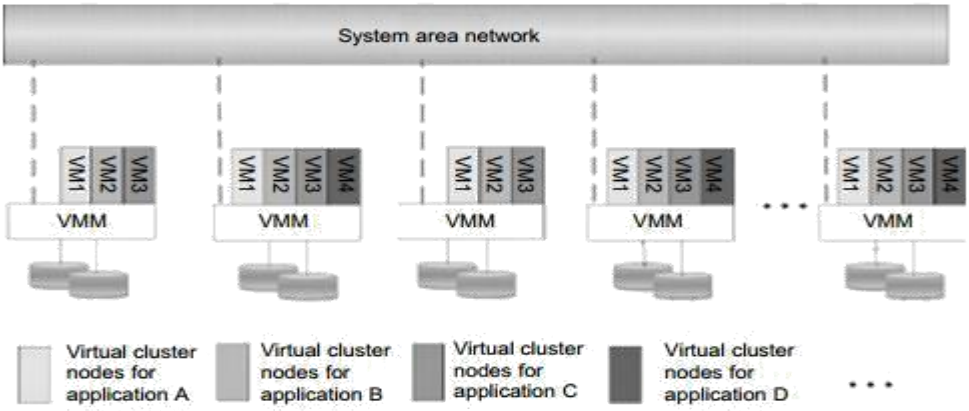
The failure of any physical nodes may disable some VMs installed on the failing nodes. But the failure of VMs will not pull down the host system.



A cloud platform with four virtual clusters over three physical clusters shaded differently.

Below diagram shows the concept of a virtual cluster based on application partitioning or customization. The different colors in the figure represent the nodes in different virtual clusters. As a large number of VM images might be present, the most important thing is to determine how to store those images in the system efficiently. There are common installations for most users or applications, such as operating systems or user-level programming libraries. These software packages can be preinstalled as templates (called template VMs). With these templates, users can build their own software stacks. New OS instances can be copied from the template VM. User-

specific components such as programming libraries and applications can be installed to those instances.



The concept of a virtual cluster based on application partitioning.

4. Explain the trust management in virtual clusters.

A VMM changes the computer architecture. It provides a layer of software between the operating systems and system hardware to create one or more VMs on a single physical platform. A VM entirely encapsulates the state of the guest operating system running inside it. Encapsulated machine state can be copied and shared over the network and removed like a normal file, which proposes a challenge to VM security. In general, a VMM can provide secure isolation and a VM accesses hardware resources through the control of the VMM, so the VMM is the base of the security of a virtual system. Normally, one VM is taken as a management VM to have some privileges such as creating, suspending, resuming, or deleting a VM.

Once a hacker successfully enters the VMM or management VM, the whole system is in danger. A subtler problem arises in protocols that rely on the ―freshness‖ of their random number source for generating session keys. Considering a VM, rolling back to a point after a random number has been chosen, but before it has been used, resumes execution; the random number, which must be ―fresh‖ for security purposes, is reused. With a stream cipher, two different plaintexts could be encrypted under the same key stream, which could, in turn, expose both plaintexts if the plaintexts have sufficient redundancy. Non-cryptographic protocols that rely on freshness are also at risk. For example, the reuse of TCP initial sequence numbers can raise TCP hijacking attacks.

VM-Based Intrusion Detection

Intrusions are unauthorized access to a certain computer from local or network users and intrusion detection is used to recognize the unauthorized access. An intrusion detection system (IDS) is built on operating systems, and is based on the characteristics of intrusion actions. A typical IDS can be classified as a host-based IDS (HIDS) or a network-based IDS (NIDS), depending on the data source. A HIDS can be implemented on the monitored system. When the monitored system is attacked by hackers, the HIDS also faces the risk of being attacked. A NIDS is based on the flow of network traffic which can't detect fake actions. Virtualization-based intrusion detection can isolate guest VMs on the same hardware platform. Even some VMs can be invaded successfully; they never influence other VMs, which is similar to the way in which a NIDS operates. Furthermore, a VMM monitors and audits access requests for hardware and system software. This can avoid fake actions and possess the merit of a HIDS. There are two different methods for implementing a VM-based IDS: Either the IDS is an independent process in each VM or a high-privileged VM on the VMM; or the IDS is integrated into the VMM and has the same privilege to access the hardware as well as the VMM.

The VM-based IDS contains a policy engine and a policy module. The policy framework can monitor events in different guest VMs by operating system interface library and PTrace indicates trace to secure policy of monitored host. It's difficult to predict and prevent all intrusions without delay. Therefore, an analysis of the intrusion action is extremely important after an intrusion occurs. At the time of this writing, most computer systems use logs to analyze attack actions, but it is hard to ensure the credibility and integrity of a log. The IDS log service is based on the operating system kernel. Thus, when an operating system is invaded by attackers, the log service should be unaffected.

Besides IDS, honeypots and honey nets are also prevalent in intrusion detection. They attract and provide a fake system view to attackers in order to protect the real system. In addition, the attack action can be analyzed, and a secure IDS can be built. A honeypot is a purposely defective system that simulates an operating system to cheat and monitor the actions of an attacker. A honeypot can be divided into physical and virtual forms. A guest operating system and the applications running on it constitute a VM. The host operating system and VMM must be guaranteed to prevent attacks from the VM in a virtual honeypot.

5. Explain the virtualization for data center automation.

The dynamic nature of cloud computing has pushed data center workload, server, and even hardware automation to whole new levels. Now, any data center provider looking to get into cloud computing must look at some form of automation to help them be as agile as possible in the cloud world.

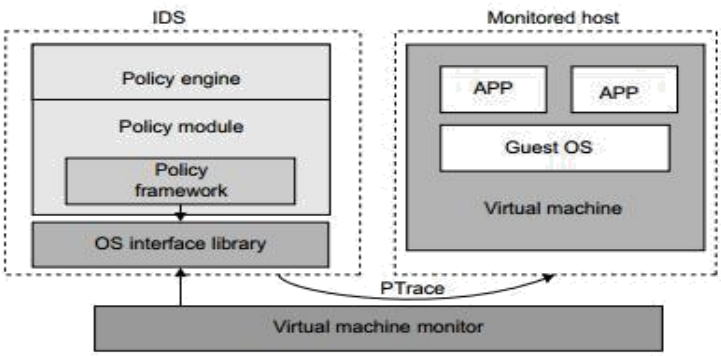
New technologies are forcing data center providers to adopt new methods to increase efficiency, scalability and redundancy. Let's face facts; there are numerous big trends which have emphasized the increased use of data center facilities. These trends include:

- More users
- More devices
- More cloud
- More workloads
- A lot more data

As infrastructure improves, more companies have looked towards the data center provider to offload a big part of their IT infrastructure. With better cost structures and even better incentives in moving towards a data center environment, organizations of all sizes are looking at colocation as an option for their IT environment.

With that, data center administrators are teaming with networking, infrastructure and cloud architects to create an even more efficient environment. This means creating intelligent systems from the hardware to the software layer. This growth in data center dependency has resulted in direct growth around automation and orchestration technologies.

Now, organizations can granularly control resources, both internally and in the cloud. This type of automation can be seen at both the software layer as well as the hardware layer. Vendors like BMC, ServiceNow, and Microsoft SCCM/SCOM are working towards unifying massive systems under one management engine to provide a single pain of glass into the data center workload environment



Furthermore, technologies like the Cisco UCS platform allow administrators to virtualize the hardware layer and create completely automated hardware profiles for new blades and servers. This hardware automation can then be tied into software-based automation tools like SCCM. Already we're seeing direct integration between software management tools and the hardware layer.

Finally, from a cloud layer, platforms like CloudStack and OpenStack allow organizations to create orchestrated and automated fluid cloud environments capable of very dynamic scalability. Still, when a physical server or hardware component breaks – we still need a person to swap out that blade.

To break it down, it's important to understand what layers of automation and orchestration are available now – and what might be available in the future. The automation and orchestration layers

Server layer. Server and hardware automation have come a long way. As mentioned earlier, there are systems now available which take almost all of the configuration pieces out of deploying a server. Administrators only need to deploy one server profile and allow new servers to pick up those settings. More data centers are trying to get into the cloud business. This means deploying high-density, fast-provisioned, servers and blades. With the on-demand nature of the cloud, being able to quickly deploy fully configured servers is a big plus for staying agile and very proactive.

Software layer. Entire applications can be automated and provisioned based on usage and resource utilization. Using the latest load-balancing tools, administrators are able to set thresholds for key applications running within the environment. If a load-balancer, a NetScaler for example, sees that a certain type of application is receiving too many connections, it can set off a process that will allow the administrator to provision another instance of the application or a new server which will host the app.

Virtual layer. The modern data center is now full of virtualization and virtual machines. In using solutions like Citrix's Provisioning Server or Unidesk's layering software technologies, administrators are able to take workload provisioning to a whole new level. Imagine being able to set a process that will kick-start the creation of a new virtual server when one starts to get over-utilized. Now, administrators can create truly automated virtual machine environments where each workload is monitored, managed and controlled.

Cloud layer. This is a new and still emerging field. Still, some very large organizations are already deploying technologies like CloudStack, OpenStack, and even OpenNebula. Furthermore, they're tying these platforms in with big data management solutions like MapReduce and Hadoop. What's happening now is true cloud-layer automation. Organizations can deploy distributed data centers and have the entire cloud layer managed by a cloud-control software platform. Engineers are able to monitor workloads, how data is being distributed, and the health of the cloud infrastructure. The great part about these technologies is that organizations can deploy a true private cloud, with as much control and redundancy as a public cloud instance.

Data center layer. Although entire data center automation technologies aren't quite here yet, we are seeing more robotics appear within the data center environment. Robotic arms already control massive tape libraries for Google and robotics automation is a thoroughly discussed concept among other large data center providers. In a recent article, we discussed the concept of a ―lights-out‖ data center in the future. Many experts agree that eventually, data center automation and robotics will likely make its way into the data center of tomorrow. For now, automation at the physical data center layer is only a developing concept.

The need to deploy more advanced cloud solution is only going to grow. More organizations of all verticals and sizes are seeing benefits of moving towards a cloud platform. At the end of the day, all of these resources, workloads and applications have to reside somewhere. That somewhere is always the data center.

In working with modern data center technologies administrators strive to be as efficient and agile as possible. This means deploying new types of automation solutions which span the

entire technology stack. Over the upcoming couple of years, automation and orchestration technologies will continue to become popular as the data center becomes an even more core piece for any organization.

6. Explain implementation levels of virtualization in details.

Virtualization is computer architecture technology by which multiple virtual machines are multiplexed in the same hardware machine. The idea of VMs can be dated back to the 1960s. The purpose of a VM is to enhance resource sharing by many users and improve computer performance in terms of resource utilization and application flexibility. Hardware resources or software resources can be virtualized in various functional layers. Levels of virtualization implementation

A traditional computer runs with a host OS specially tailored for its hardware architecture. After virtualization different user applications managed by their own OS (Guest OS) can run on the same hardware, independent of the host OS. This is often done by adding additional software called virtualization layer. This virtualization layer is known as hypervisor or **Virtual Machine Monitor.**

The main function of the software layer for virtualization is to virtualize the physical hardware of a host machine into virtual resources to be used by the VMs exclusively. Common virtualization layers include the instruction set architecture level, hardware level, OS Level, Library support level and application level.

Instruction set architecture

At the ISA level virtualization is performed by emulating a given ISA by the ISA of the host machine. For example, MIPS binary code can run on an 8086 based host machine with help of ISA emulation.

The basic emulation method is through code interpretation. An interpreter program interprets the source instructions to target instructions one by one. One source instruction may require tens or hundreds of native target instructions to perform its function.

Hardware Architecture

Hardware level virtualization is performed right on top of the bar hardware. On the one hand this approach generates a virtual hardware environment for a VM. On the other hand the process manages the underlying hardware through virtualization.

The idea is to virtualize a computers resources, such as its processors, memory and I/O devices **OS Level**

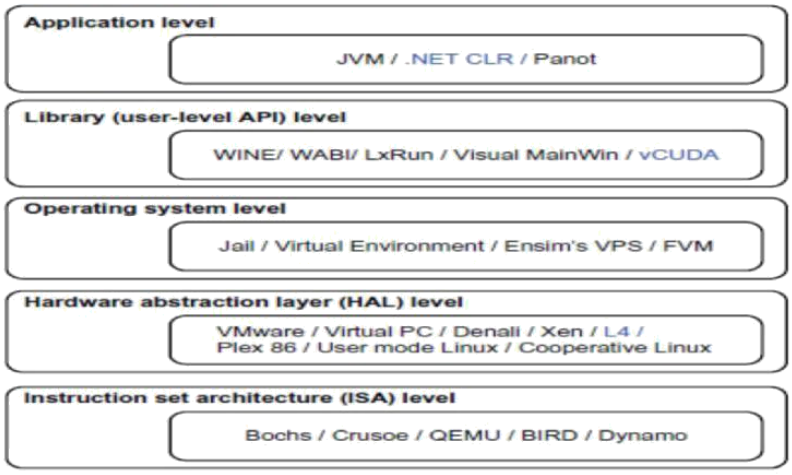
This refers to an abstraction layer between traditional OS and user application. OS level virtualization creates isolated containers on a single physical server and the OS instances to utilize the hardware and software in data centers.

Library Support Level

Most application use APIs exported by user level libraries rather than using lengthy system calls by the OS. Since most systems provide well documented APIs, such an interface becomes another candidate for virtualization. Virtualization with library interfaces is possible by controlling the communication link between applications and the rest of a system through API hooks.

User Application Level

Virtualization at the application level virtualizes an application as a VM. On a traditional OS, an application often runs as a process. Therefore, application level virtualization is known as process level virtualization. The most popular approach is to deploy high level language VMs.



Virtualization ranging from hardware to applications in five abstraction levels.

2 Explain the virtualization of CPU, Memory and I/O devices Virtualization of CPU

A VM is a duplicate of an existing computer system in which a majority of the VM instructions are executed on the host processor in native mode. Thus, unprivileged instructions of VMs run directly on the host machine for higher efficiency. The critical instructions are divided into three categories.

- Privileged instructions
- Control sensitive instructions
- Behaviour sensitive instructions

Privileged instructions execute in a privileged mode and will be trapped if executes outside this mode.

Control sensitive instructions attempt to change the configuration of resources used.

Behavior sensitive instructions have different behaviors depending on the configuration of resources, including the load and store operations over the virtual memory.

A CPU architecture is virtualizable if it supports the ability to run the VM's privileged and unprivileged instructions in the CPU's user mode while the VMM run in supervisor mode. When the privileged instructions including control and behavior sensitive instructions of a VM are executed they are trapped in the VMM.

RISC CPU architectures can be naturally virtualized because all control and behavior sensitive instructions are privileged instruction.

Hardware Assisted CPU virtualization

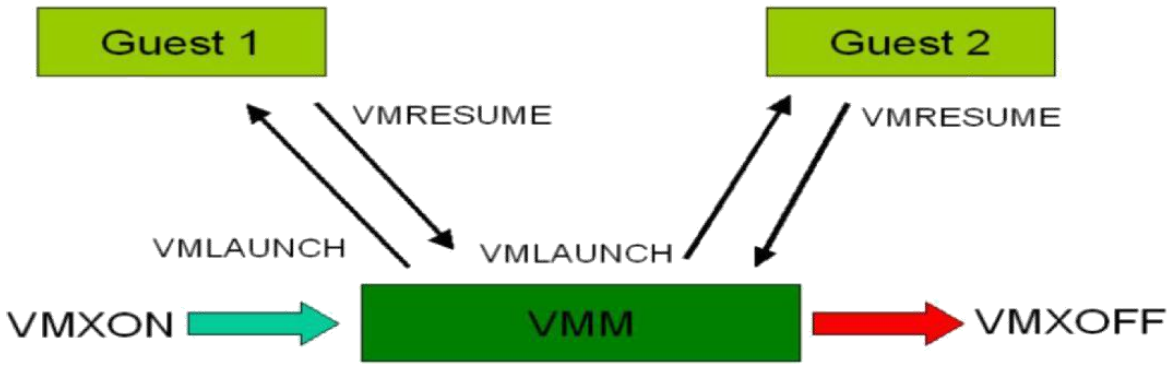
- Processors with virtualization technology have extra instruction set called virtual machine extensions or VMX.

There are two modes to run under virtualization: root operation and non-root operation. Usually only the virtualization controlling software, called Virtual Machine Monitor (VMM), runs under root operation, while operating systems running on top of the virtual machines run

under non -root operation. Software running on top of virtual machines is also called 'guest

software,.

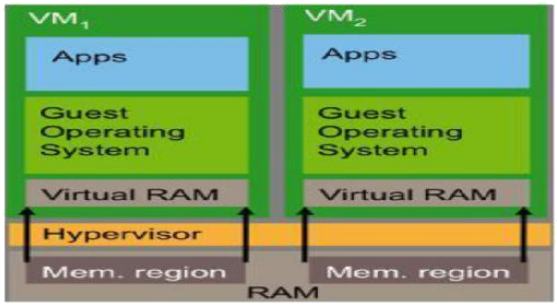
To enter virtualization mode, the software should execute the VMXON instruction and then call the VMM software. Then VMM software can enter each virtual machine using the VMLAUNCH instruction, and exit it by using the VMRESUME. If VMM wants to shut down and exit virtualization mode, it executes the VMXOFF instruction.



Memory Virtualization

Virtual memory virtualization is similar to the virtual memory support provided by modern operating systems. In a traditional execution environment the OS maintains mappings of virtual memory to machine memory using page tables, which is one stage mapping from virtual memory to machine memory. All modern x86 CPUs include a Memory management Unit and a translation Look-aside Buffer to optimize virtual memory performance. In virtual execution environment virtual memory virtualization involves sharing the physical system memory in RAM and dynamically allocating it to the physical memory of the VMs.

Guest OS sees flat ‚physical' address space.
Page tables within guest OS: • Translate from virtual to physical addresses.
Second-level mapping: • Physical addresses to machine addresses.
VMM can swap a VM's pages to disk.
Traditional way is to have the VMM maintain a shadow of the VM's page table.
The shadow page table controls which pages of machine memory are assigned to a given VM. When OS updates it's page table, VMM updates the shadow



Memory Virtualization

I/O Virtualization

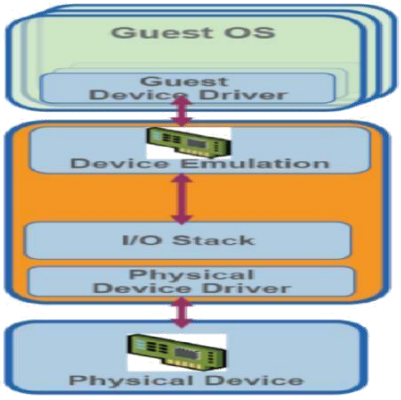
Input/output (I/O) virtualization is a methodology to simplify management, lower costs and improve performance of servers in enterprise environments. I/O virtualization environments are created by abstracting the upper layer protocols from the physical connections.

The technology enables one physical adapter card to appear as multiple virtual network interface cards (vNICs) and virtual host bus adapters (vHBAs). Virtual NICs and HBAs function as conventional NICs and HBAs, and are designed to be compatible with existing operating systems, hypervisors, and applications. To networking resources (LANs and SANs), they appear as normal cards.

In the physical view, virtual I/O replaces a server's multiple I/O cables with a single cable that provides a shared transport for all network and storage connections. That cable (or commonly two cables for redundancy) connects to an external device, which then provides connections to the data center networks.

Server I/O is a critical component to successful and effective server deployments, particularly with virtualized servers. To accommodate multiple applications, virtualized servers demand more network bandwidth and connections to more networks and storage. According to a survey, 75% of virtualized servers require 7 or more I/O connections per device, and are likely to require more frequent I/O reconfigurations.

In virtualized data centers, I/O performance problems are caused by running numerous virtual machines (VMs) on one server. In early server virtualization implementations, the number of virtual machines per server was typically limited to six or less. But it was found that it could safely run seven or more applications per server, often using 80 percentage of total server capacity, an improvement over the average 5 to 15 percentage utilized with non-virtualized servers.



I/O Virtualization architecture consists of

- Guest driver
- Virtual device
- Communication mechanism between virtual device and virtualization stack
- Virtualization I/O stack
- Physical device driver
- Real device

Virtualization I/O stack

Translates guest I/O addresses to host addresses

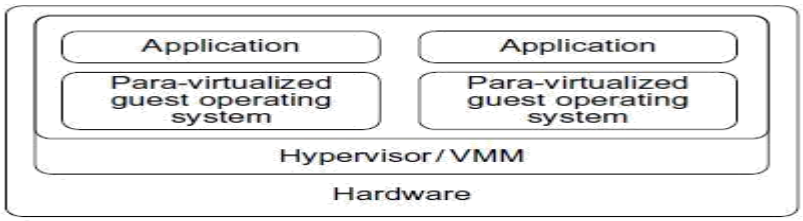
Handles inter VM communication

Multiplexes I/O requests from/to the physical device

Provides enterprise-class I/O features to the Guest

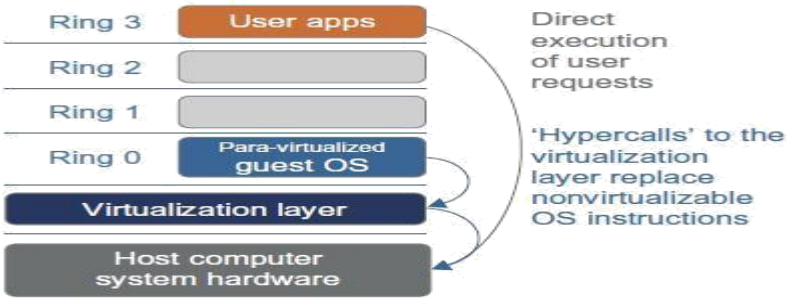
8. a. Para-Virtualization with Compiler Support

Para-virtualization needs to modify the guest operating systems. A para-virtualized VM provides special APIs requiring substantial OS modifications in user applications. Performance degradation is a critical issue of a virtualized system. No one wants to use a VM if it is much slower than using a physical machine. The virtualization layer can be inserted at different positions in a machine software stack. However, para-virtualization attempts to reduce the virtualization overhead, and thus improve performance by modifying only the guest OS kernel. The concept of a para-virtualized VM architecture. The guest operating systems are para-virtualized.



Para-virtualized VM architecture, which involves modifying the guest OS kernel to replace nonvirtualizable instructions with hypercalls for the hypervisor or the VMM to carry out the virtualization process

They are assisted by an intelligent compiler to replace the nonvirtualizable OS instructions by hypercalls as llustrated in below figure. The traditional x86 processor offers four instruction execution rings: Rings 0, 1, 2, and 3. The lower the ring number, the higher the privilege of instruction being executed. The OS is responsible for managing the hardware and the privileged instructions to execute at Ring 0, while user-level applications run at Ring 3.



The use of a para-virtualized guest OS assisted by an intelligent compiler to replace nonvirtualizable OS instructions by hypercalls.

Para-Virtualization Architecture

When the x86 processor is virtualized, a virtualization layer is inserted between the hardware and the OS. According to the x86 ring definition, the virtualization layer should also be installed at Ring 0. Different instructions at Ring 0 may cause some problems. In above diagram, we show that para-virtualization replaces nonvirtualizable instructions with hypercalls that communicate directly with the hypervisor or VMM. However, when the guest OS kernel is modified for virtualization, it can no longer run on the hardware directly.

Although para-virtualization reduces the overhead, it has incurred other problems. First, its compatibility and portability may be in doubt, because it must support the unmodified OS as well. Second, the cost of maintaining para-virtualized OSes is high, because they may require deep OS kernel modifications. Finally, the performance advantage of para-virtualization varies greatly due to workload variations. Compared with full virtualization, para-virtualization is relatively easy and more practical. The main problem in full virtualization is its low performance in binary translation. To speed up binary translation is difficult. Therefore, many virtualization products employ the para-virtualization architecture. The popular Xen, KVM, and VMware ESX are good examples.

8. b. Binary Translation with Full Virtualization

Depending on implementation technologies, hardware virtualization can be classified into two categories:

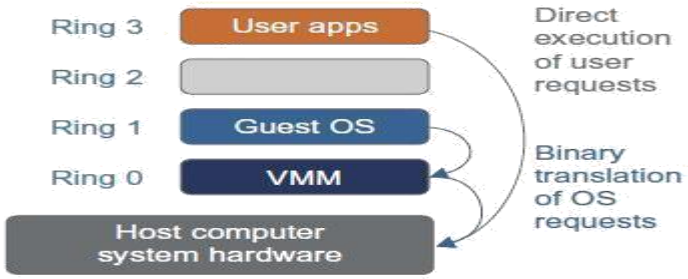
Full virtualization and host-based virtualization. Full virtualization does not need to modify the host OS. It relies on binary translation to trap and to virtualize the execution of certain sensitive, nonvirtualizable instructions. The guest OSes and their applications consist of noncritical and critical instructions. In a host-based system, both a host OS and a guest OS are used. A virtualization software layer is built between the host OS and guest OS. These two classes of VM architecture are introduced next.

Full Virtualization

With full virtualization, noncritical instructions run on the hardware directly while critical instructions are discovered and replaced with traps into the VMM to be emulated by software. Both the hypervisor and VMM approaches are considered full virtualization. Why are only critical instructions trapped into the VMM? This is because binary translation can incur a large performance overhead. Noncritical instructions do not control hardware or threaten the security of the system, but critical instructions do. Therefore, running noncritical instructions on hardware not only can promote efficiency, but also can ensure system security.

Binary Translation of Guest OS Requests Using a VMM

This approach was implemented by VMware and many other software companies. VMware puts the VMM at Ring 0 and the guest OS at Ring 1. The VMM scans the instruction stream and identifies the privileged, control- and behavior-sensitive instructions. When these instructions are identified, they are trapped into the VMM, which emulates the behavior of these instructions. The method used in this emulation is called binary translation. Therefore, full virtualization combines binary translation and direct execution. The guest OS is completely decoupled from the underlying hardware. Consequently, the guest OS is unaware that it is being virtualized. The performance of full virtualization may not be ideal, because it involves binary translation which is rather time-consuming. In particular, the full virtualization of I/O-intensive applications is a really a big challenge. Binary translation employs a code cache to store translated hot instructions to improve performance, but it increases the cost of memory usage. At the time of this writing, the performance of full virtualization on the x86 architecture is typically 80 percent to 97 percent that of the host machine.



Indirect execution of complex instructions via binary translation of guest OS requests using the VMM plus direct execution of simple instructions on the same host.

Host-Based Virtualization

An alternative VM architecture is to install a virtualization layer on top of the host OS. This host OS is still responsible for managing the hardware. The guest OSes are installed and run on top of the virtualization layer. Dedicated applications may run on the VMs. Certainly, some other applications can also run with the host OS directly. This hostbased architecture has some distinct advantages, as enumerated next. First, the user can install this VM architecture without modifying the host OS. The virtualizing software can rely on the host OS to provide device drivers and other low-level services. This will simplify the VM design and ease its deployment. Second, the host-based approach appeals to many host machine configurations. Compared to the hypervisor/VMM architecture, the performance of the host-based architecture may also be low. When an application requests hardware access, it involves four layers of mapping which downgrades performance significantly. When the ISA of a guest OS is different from the ISA of the underlying hardware, binary translation must be adopted. Although the host-based architecture has flexibility, the performance is too low to be useful in practice.

9. Explain the characteristics and types of virtualization in cloud computing.

Virtualization is using computer resources to imitate other computer resources or whole computers. It separates resources and services from the underlying physical delivery environment.

Virtualization has three characteristics that make it ideal for cloud computing:

Partitioning: In virtualization, many applications and operating systems (OSes) are supported in a single physical system by partitioning (separating) the available resources.

Isolation: Each virtual machine is isolated from its host physical system and other virtualized machines. Because of this isolation, if one virtual-instance crashes, it doesn't affect the other virtual machines. In addition, data isn't shared between one virtual container and another.

Encapsulation: A virtual machine can be represented (and even stored) as a single file, so you can identify it easily based on the service it provides. In essence, the encapsulated process could be a business service. This encapsulated virtual machine can be presented to an application as a

complete entity. Therefore, encapsulation can protect each application so that it doesn't interfere with another application.

Types:

Virtualization can be utilized in many different ways and can take many forms aside from just server virtualization. The main types include application, desktop, user, storage and hardware.

Application virtualization allows the user to access the application, not from their workstation, but from a remotely located server. The server stores all personal information and other characteristics of the application, but can still run on a local workstation. Technically, the application is not installed, but acts like it is.

Desktop virtualization allows the users' OS to be remotely stored on a server in the data center, allowing the user to then access their desktop virtually, from any location.

User virtualization is pretty similar to desktop, but allows users the ability to maintain a fully personalized virtual desktop when not on the company network. Users can basically log into their ―desktop‖ from different types of devices like smartphones and tablets. With more companies migrating to a BYOD policy, desktop and user virtualization are becoming increasingly popular.

Storage virtualization is the process of grouping the physical storage from multiple network storage devices so that it acts as if it's on one storage device.

Hardware virtualization (also referred to as hardware-assisted virtualization) is a form of virtualization that uses one processor to act as if it were several different processors. The user can then run different operating systems on the same hardware, or more than one user can use the processor at the same time. This type of virtualization requires a virtual machine manager (VM) called a hypervisor.

10. Explain the NIST reference architecture of cloud computing in detail

The Conceptual Reference Model Figure 1 presents an overview of the NIST cloud computing reference architecture, which identifies the major actors, their activities and functions in cloud computing. The diagram depicts a generic high-level architecture and is intended to facilitate the understanding of the requirements, uses, characteristics and standards of cloud computing.

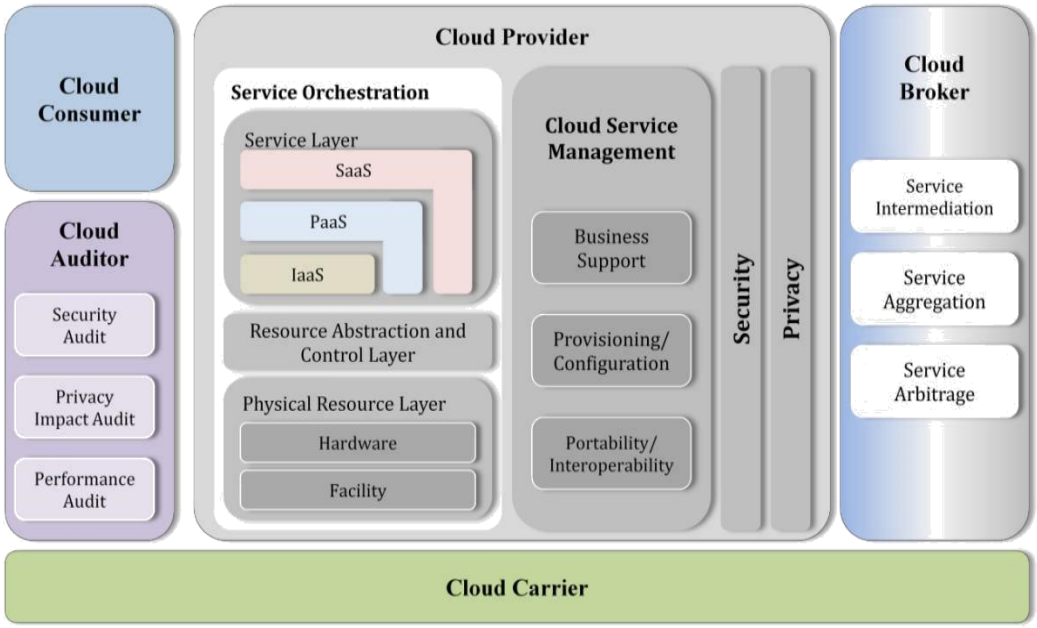


Figure 1: The Conceptual Reference Model

As shown in Figure 1, the NIST cloud computing reference architecture defines five major actors: cloud consumer, cloud provider, cloud carrier, cloud auditor and cloud broker. Each actor is an entity (a person or an organization) that participates in a transaction or process and/or performs tasks in cloud computing.

Actor	Definition
Cloud Consumer	A person or organization that maintains a business relationship with, and uses service from, *Cloud Providers*.
Cloud Provider	A person, organization, or entity responsible for making a service available to interested parties.
Cloud Auditor	A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.
Cloud Broker	An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between *Cloud Providers* and *Cloud Consumers.*
Cloud Carrier	An intermediary that provides connectivity and transport of cloud services from *Cloud Providers* to *Cloud Consumers*.

Unit – 4 – Progamming Model
Part – A

1. List out the grid middleware packages

Package	Description
BOINC	Berkeley Open Infrastructure for Network Computing.
UNICORE	Middleware developed by the German grid computing community
Globus (GT4)	A middleware library jointly developed by Argonne National Lab.
CGSP in ChinaGrid	The CGSP (ChinaGrid Support Platform) is a middleware library developed by 20 top universities in China as part of the ChinaGrid Project
Condor-G	Originally developed at the Univ. of Wisconsin for general distributed computing, and later extended to Condor-G for grid job management.
Sun Grid Engine (SGE)	Developed by Sun Microsystems for business grid applications. Applied to private grids and local clusters within enterprises or campuses.

2. Define MapReduce.
The mapreduce software framework provides an abstraction layer with the data flow and flow of control of users and hides implementation of all data flow steps such as data partitioning mapping, synchronization, communication and scheduling. The data flow is such framework is predefined the abstraction layer provides two well defined interface in the form of two functions map and reduce**.**

3. What is the role of Map function?
Each Map function receives the input data split as a set of (key, value) pairs to process and produce the intermediated (key, value) pairs.

4. What is the role of Reduce function?
The reduce worker iterates over the grouped (key, value) pairs, and for each unique key, it sends the key and corresponding values to the Reduce function. Then this function processes its input data and stores the output results in predetermined files in the user's program.

5. List out the Hadoop core fundamental layers
The Hadoop core is divided into two fundamental layers: the MapReduce engine and HDFS. The MapReduce engine is the computation engine running on top of HDFS as its data storage manager. HDFS is a distributed file system inspired by GFS that organizes files and stores their data on a distributed computing system.

6. What are the features of HDFS?
HDFS is not a general-purpose file system, as it only executes specific types of applications, it does not need all the requirements of a general distributed file system. For example, security has never been supported for HDFS systems.

2 List the areas where HDFS cannot be used? Low-latency data access
Lots of small files
Multiple writers, arbitrary file modifications

3 Why is a block in HDFS so large?

HDFS blocks are large compared to disk blocks, and the reason is to minimize the cost of seeks. By making a block large enough, the time to transfer the data from the disk can be made to be significantly larger than the time to seek to the start of the block. Thus the time to transfer a large file made of multiple blocks operates at the disk transfer rate.

9. Define Namenode in HDFS

The namenode manages the filesystem namespace. It maintains the filesystem tree and the metadata for all the files and directories in the tree. This information is stored persistently on the local disk in the form of two files: the namespace image and the edit log. The namenode also knows the datanodes on which all the blocks for a given file are located, however, it does not store block locations persistently, since this information is reconstructed from datanodes when the system starts.

10. Define Datanode in HDFS

Datanodes are the work horses of the filesystem. They store and retrieve blocks when they are told to (by clients or the namenode), and they report back to the namenode periodically with lists of blocks that they are storing.

11. What are the permission models for files and directories in HDFS

There are three types of permission: the read permission (r), the write permission (w) and the execute permission (x). The read permission is required to read files or list the contents of a directory. The write permission is required to write a file, or for a directory, to create or delete files or directories in it. The execute permission is ignored for a file since you can't execute a file on HDFS (unlike POSIX), and for a directory it is required to access its children.

12. Define FUSE interface?

Filesystem in Userspace (FUSE) allows filesystems that are implemented in user space to be integrated as a Unix filesystem. Hadoop's Fuse-DFS contrib module allows any Hadoop filesystem (but typically HDFS) to be mounted as a standard filesystem. You can then use Unix utilities (such as ls and cat) to interact with the filesystem, as well as POSIX libraries to access the filesystem from any programming language. Fuse-DFS is implemented in C using *libhdfs* as the interface to HDFS.

13. Define globbing in HDFS?

It is a common requirement to process sets of files in a single operation.. To enumerate each file and directory to specify the input, it is convenient to use wildcard characters to match multiple files with a single expression, an operation that is known as *globbing*.

14. How to process globs in hadoop filesystem?

Hadoop provides two FileSystem methods for processing globs:

public FileStatus[] globStatus(Path pathPattern) throws IOException
public FileStatus[] globStatus(Path pathPattern, PathFilter filter) throws IOException

The globStatus() methods returns an array of FileStatus objects whose paths match the supplied pattern, sorted by path. An optional PathFilter can be specified to restrict the matches further

15. How to delete file or directory in hadoop filesystem?

Use the delete() method on FileSystem to permanently remove files or directories:

public boolean delete(Path f, boolean recursive) throws IOException

If *f* is a file or an empty directory, then the value of recursive is ignored. A nonempty directory is only deleted, along with its contents, if recursive is true (otherwise an IOException is thrown).

16. Define iterative MapReduce.

It is important to understand the performance of different runtime and in particular to compare MPI and map reduce. The two major sources of parallel overhead are load imbalance and communication. The communication overhead in mapreduce can be high for two reasons.

9. Mapreduce read and writes files whereas MPI transfer information directly between nodes over the network.

19) • MPI does not transfer all data from node to node.

17. Define HDFS.

HDFS is a distributed file system inspired by GFS that organizes files and stores their data on a distributed computing system. The hadoop implementation of mapreduce uses the hadoop distributed file system as in underlying layer rather than GFS.

2 List the characteristics of HDFS.

II. HDFS fault tolerance

ϑϑ. Block replication

KK. Relica placement

ΛΛ.Heartbeat and block report messages

- HDFS high throughput access to large dataset. **19. What are the operations of HDFS?**

The control flow of HDFS operation such as read and write can properly highlights role of the name node and data node in the managing operations. The control flow of the main operations of HDFS on file is further described to manifest the interaction between the users**.**

20. Define block replication.

The reliably store data in HDFS is the file blocks, it is replicated in this system. HDFS store a file as a set of blocks and each block is replicated and distributed across the whole cluster.

21. Define heart beat in Hadoop. What are the advantages of heart beat?

The heart beat are periodic messages sent to the name node by each data node in the cluster. Receipt of a heartbeat implies that data mode is functioning properly while each block report contains list of all blocks in a data mode. The name node receives such messages because it is the sole decision maker of all replicas in the system.

22. List out the functional modules in globus GT4 library

Service Functionality	**Module Name**	**Functional Description**
Global Resource Allocation Manager	GRAM	Grid Resource Access and Management (HTTP-based)
Communication	Nexus	Unicast and multicast communication
Grid Security Infrastructure	GSI	Authentication and related security services
Monitory and Discovery Service	MDS	Distributed access to structure and state information
Health and Status	HBM	Heartbeat monitoring of system components
Global Access of Secondary Storage	GASS	Grid access of data in remote secondary storage
Grid File Transfer	GridFTP	Inter-node fast file transfer

23. Define Globus Resource Allocation Manager

Globus Resource Allocation Manager (GRAM) provides resource allocation, process creation, monitoring, and management services. GRAM implementations map requests expressed in a resource specification language (RSL) into commands to local schedulers and computers.

24. Define Monitoring and Discovery Service

The Monitoring and Discovery Service (MDS) is an extensible grid information service that combines data discovery mechanisms with the LDAP (LDAP defines a data model, query language, and other related protocols). MDS provides a uniform framework for providing and accessing system configuration and status information such as computer server configuration, network status, or the locations of replicated datasets.

Part- B

1. Explain in detail about Grid Middleware Packages

We first introduce some grid standards and popular APIs. Then we present the desired software support and middleware developed for grid computing.

Grid Standards and APIs

The Open Grid Forum (formally Global Grid Forum) and Object Management Group are two well-formed organizations behind those standards. we have also reported some grid standards including the GLUE for resource representation, SAGA (Simple API for Grid Applications), GSI (Grid Security Infrastructure), OGSI (Open Grid Service Infrastructure), and WSRE (Web Service Resource Framework).

Software Support and Middleware

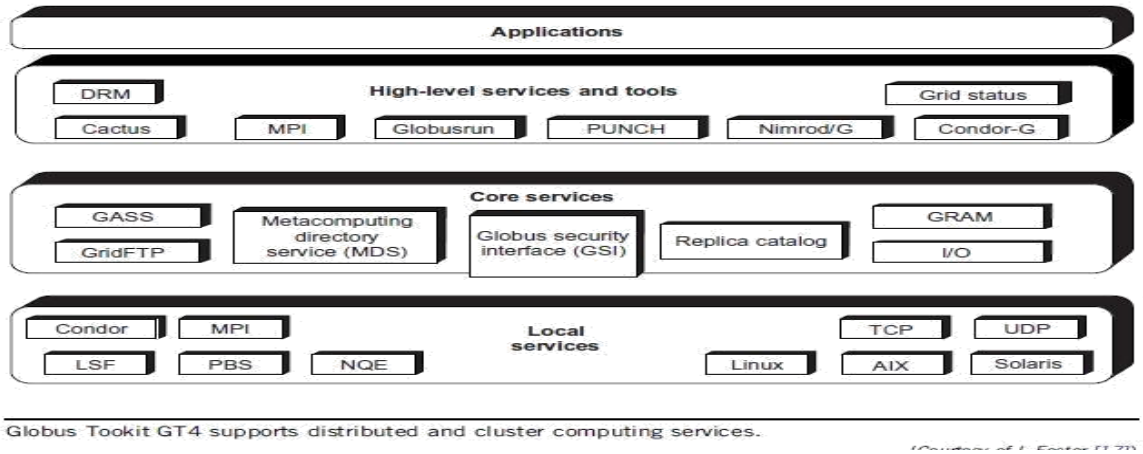
Grid middleware is specifically designed a layer between hardware and the software. The middleware products enable the sharing of heterogeneous resources and managing virtual organizations created around the grid. Middleware glues the allocated resources with specific

user applications. Popular grid middleware tools include the Globus Toolkits (USA), gLight, UNICORE (German), BOINC (Berkeley), CGSP (China), Condor-G, and Sun Grid Engine, etc.

Package	Description
BOINC	Berkeley Open Infrastructure for Network Computing.
UNICORE	Middleware developed by the German grid computing community
Globus (GT4)	A middleware library jointly developed by Argonne National Lab.
CGSP in ChinaGrid	The CGSP (ChinaGrid Support Platform) is a middleware library developed by 20 top universities in China as part of the ChinaGrid Project
Condor-G	Originally developed at the Univ. of Wisconsin for general distributed computing, and later extended to Condor-G for grid job management.
Sun Grid Engine (SGE)	Developed by Sun Microsystems for business grid applications. Applied to private grids and local clusters within enterprises or campuses.

2. The Globus Toolkit Architecture

GT4 is an open middleware library forthe grid computing communities. These open source software libraries support many operational grids and their applications on an international basis. The toolkit addresses common problems and issues related to grid resource discovery, management, communication, security, fault detection, and portability. The software itself provides a variety of components and capabilities. The library includes a rich set of service implementations.



Globus Tookit GT4 supports distributed and cluster computing services.

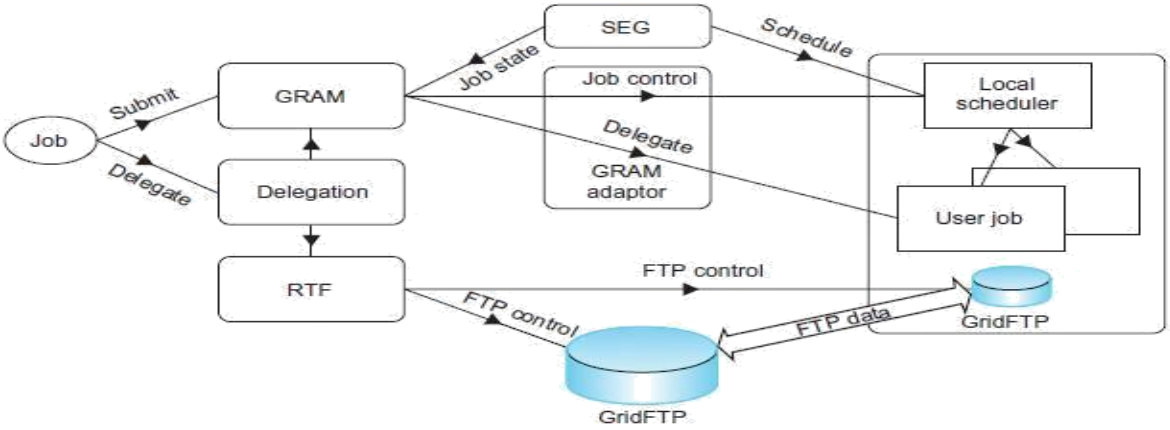
The GT4 Library

GT4 offers the middle-level core services in grid applications. The high-level services and tools, such as MPI, Condor-G, and Nirod/G, are developed by third parties for general-purpose distributed computing applications. The local services, such as LSF, TCP, Linux, and Condor, are at the bottom level and are fundamental tools supplied by other developers.

Service Functionality	Module Name	Functional Description
Global Resource Allocation Manager	GRAM	Grid Resource Access and Management (HTTP-based)
Communication	Nexus	Unicast and multicast communication
Grid Security Infrastructure	GSI	Authentication and related security services
Monitory and Discovery Service	MDS	Distributed access to structure and state information
Health and Status	HBM	Heartbeat monitoring of system components
Global Access of Secondary Storage	GASS	Grid access of data in remote secondary storage
Grid File Transfer	GridFTP	Inter-node fast file transfer

Globus Job Workflow

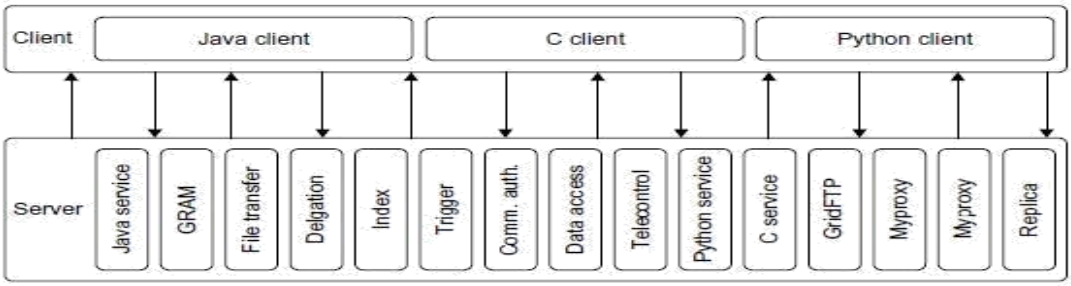
A typical job execution sequence proceeds as follows: The user delegates his credentials to a delegation service. The user submits a job request to GRAM with the delegation identifier as a parameter. GRAM parses the request, retrieves the user proxy certificate from the delegation service, and then acts on behalf of the user. GRAM sends a transfer request to the RFT (Reliable File Transfer), which applies GridFTP to bring in the necessary files. GRAM invokes a local scheduler via a GRAM adapter and the SEG (Scheduler Event Generator) initiates a set of user jobs. The local scheduler reports the job state to the SEG. Once the job is complete, GRAM uses RFT and GridFTP to stage out the resultant files. The grid monitors the progress of these operations and sends the user a notification when they succeed, fail, or are delayed.



Globus job workflow among interactive functional modules.

Client-Globus Interactions

GT4 service programs are designed to support user applications. There are strong interactions between provider programs and user code. GT4 makes heavy use of industry-standard web service protocols and mechanisms in service description, discovery, access, authentication, authorization, and the like. GT4 makes extensive use of Java, C, and Python to write user code. Web service mechanisms define specific interfaces for grid computing. Web services provide flexible, extensible, and widely adopted XML-based interfaces.



Client and GT4 server interactions; vertical boxes correspond to service programs and horizontal boxes represent the user codes.

3. Explain the MapReduce technique

MapReduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster. A MapReduce program is composed of a **Map()** procedure that performs filtering and sorting (such as sorting students by first name into queues, one queue for each name) and a **Reduce()** procedure that performs a summary operation (such as counting the number of students in each queue, yielding name frequencies). The "MapReduce System" (also called "infrastructure" or "framework") orchestrates the processing by marshalling the distributed servers, running the various tasks in parallel, managing all communications and data transfers between the various parts of the system, and providing for redundancy and fault tolerance.

The model is inspired by the map and reduce functions commonly used in functional programming, although their purpose in the MapReduce framework is not the same as in their original forms. The key contributions of the MapReduce framework are not the actual map and reduce functions, but the scalability and fault-tolerance achieved for a variety of applications by optimizing the execution engine once. As such, a single-threaded implementation of MapReduce (such as MongoDB) will usually not be faster than a traditional (non-MapReduce) implementation, any gains are usually only seen with multi-threaded implementations. Only when the optimized distributed shuffle operation (which reduces network communication cost) and fault tolerance features of the MapReduce framework come into play, is the use of this model beneficial. Optimizing the communication cost is essential to a good MapReduce algorithm.

MapReduce libraries have been written in many programming languages, with different levels of optimization. A popular open-source implementation that has support for distributed shuffles is part of Apache Hadoop. The name MapReduce originally referred to the proprietary Google technology, but has since been genericized.

Hadoop is an open-source framework for writing and running distributed applications that process very large data sets. There has been a great deal of interest in the framework, and it is very popular in industry as well as in academia. Hadoop cases include: web indexing, scientific simulation, social network analysis, fraud analysis, recommendation engine, ad targeting, threat analysis, risk modeling and other. Hadoop is core part of a cloud computing infrastructure and is being used by companies like Yahoo, Facebook, IBM, LinkedIn, and Twitter. The main benefits of Hadoop framework can be summarized as follows:

Accessible: it runs on clusters of commodity servers Scalable: it scales linearly to handle larger data by adding nodes to the cluster

Fault-tolerant: it is designed with the assumption of frequent hardware failures

Simple: it allows user to quickly write efficiently parallel code

Global: it stores and analyzes data in its native format

Hadoop is designed for data-intensive processing tasks and for that reason it has adopted a move- code-to-data" philosophy. According to that philosophy, the programs to run, which are small in size, Are transferred to nodes that store the data. In that way, the framework achieves better performance and resource utilization. In addition, Hadoop solves the hard scaling problems caused by large amounts of complex data. As the amount of data in a cluster grows, new servers can be incrementally and inexpensively added to store and analyze it.

Hadoop has two major subsystems: the Hadoop Distributed File System (HDFS) and a distributed data processing framework called MapReduce. Apart from these two main components, Hadoop has grown into a complex ecosystem, including a range of software systems. Core related applications that are built on top of the HDFS are presented in figure and a short description per project is given in table.

MapReduce is a framework for processing parallelizable problems across huge datasets using a large number of computers (nodes), collectively referred to as a cluster (if all nodes are on the same local network and use similar hardware) or a grid (if the nodes are shared across geographically and administratively distributed systems, and use more heterogenous hardware). Processing can occur on data stored either in a filesystem (unstructured) or in a database (structured). MapReduce can take advantage of locality of data, processing it on or near the storage assets in order to reduce the distance over which it must be transmitted.

2 **"Map" step:** Each worker node applies the "map()" function to the local data, and writes the output to a temporary storage. A master node orchestrates that for redundant copies of input data, only one is processed.

20) **"Shuffle" step:** Worker nodes redistribute data based on the output keys (produced by the "map()" function), such that all data belonging to one key is located on the same worker node.

21) **"Reduce" step:** Worker nodes now process each group of output data, per key, in parallel.

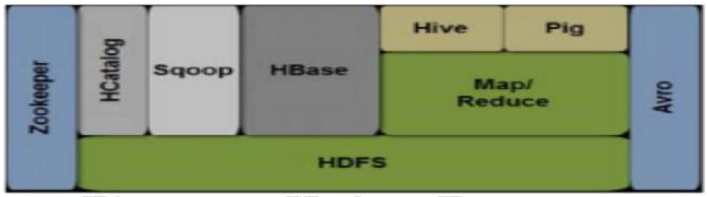


Figure Hadoop Ecosystem

Project	Info
Hdfs	Hadoop distributed file system
Map reduce	Distributed computation framework
Zookeeper	High-performance collaboration service
Hbase	Column-oriented table service
Pig	Dataflow language and parallel execution
Hive	Data warehouse infrastructure
Hcatalog	Table and storage management service
Sqoop	Bulk data transfer
Avron	Data serialization system

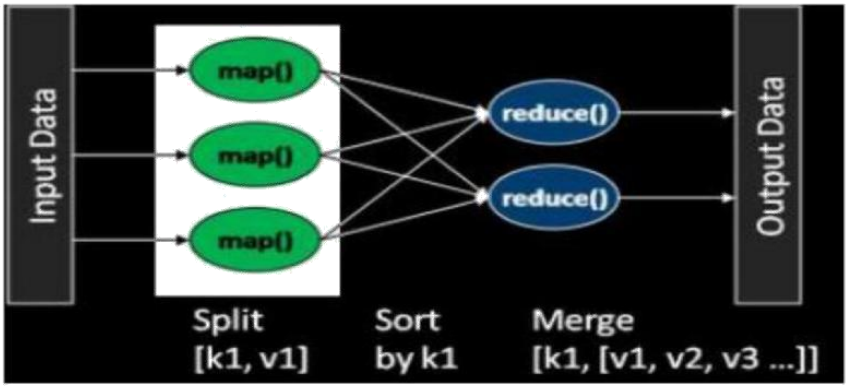
Table Project Descriptions

4. Explain the architecture of MapReduce in Hadoop?

The Hadoop MapReduce MRv1 framework is based on a centralized master/slave architecture. The architecture utilizes a single master server (JobTracker) and several slave servers (TaskTracker's). Please see Appendix A for a discussion on the MapReduce MRv2 framework. The JobTracker represents a centralized program that keeps track of the slave nodes, and provides an interface infrastructure for job submission. The TaskTracker executes on each of the slave nodes where the actual data is normally stored. In other words, the JobTracker reflects the interaction point among the users and the Hadoop framework. Users submit MapReduce jobs to the JobTracker, which inserts the jobs into the pending jobs queue and executes them (normally) on a FIFO basis (it has to be pointed out that other job schedulers are available - see Hadoop Schedulers below). The JobTracker manages the map and reduce task assignments with the TaskTracker's. The TaskTracker's execute the jobs based on the instructions from the JobTracker and handle the data movement between the maps and reduce phases, respectively. Any map/reduce construct basically reflects a special form of a Directed Acyclic Graph (DAG). A DAG can execute anywhere in parallel, as long as one entity is not an ancestor of another entity. In other words, parallelism is achieved when there are no hidden dependencies among shared states. In the MapReduce model, the internal organization is based on the map function that transforms a piece of data into entities of [key, value] pairs. Each of these elements is sorted (via their key) and ultimately reaches the same cluster node where a reduce function is used to merge the values (with the same key) into a single result (see code below). The Map/Reduce DAG is organized as depicted in Figure.

```
Map Function

map(input_record){
…
emit(k1,v1)
…
emit(k2,v2)
…
}
```

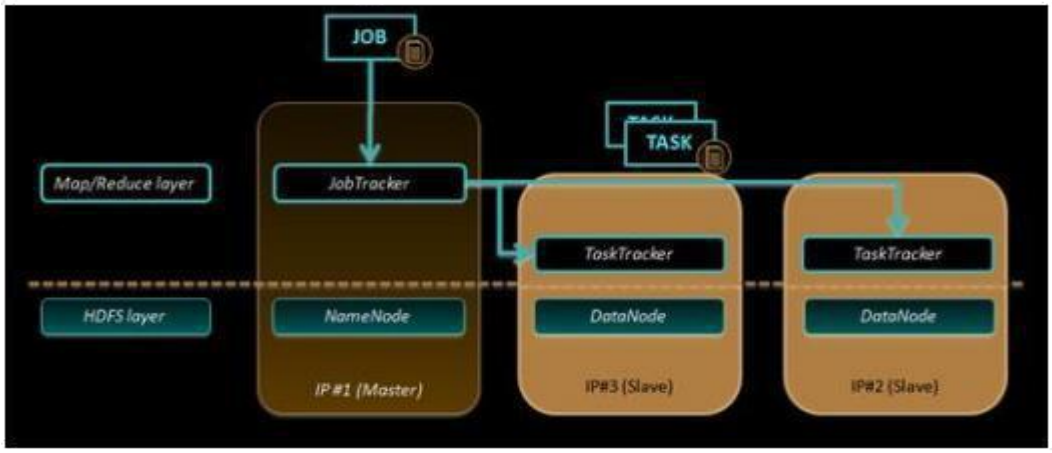
```
Reduce Function

reduce(key,values){
while(values.has_next){
aggregate-merge(values.next)
  }
collect(key,aggregate)
}
```



The Hadoop MapReduce framework is based on a pull model, where multiple TaskTracker's communicate with the JobTracker requesting tasks (either map or reduce tasks). After an initial setup phase, the JobTracker is informed about a job submission. The JobTracker provides a job ID to the client program, and starts allocating map tasks to idle TaskTracker's requesting work items (see below Figure). Each TaskTracker contains a defined number of task slots based on the capacity potential of the system. Via the heartbeat protocol, the JobTracker knows the number of free slots in the TaskTracker (the TaskTracker's send heartbeat messages indicating the free slots

6) true for the FIFO scheduler). Hence, the JobTracker can determine the appropriate job setup for a TaskTracker based on the actual availability behavior. The assigned TaskTracker will fork a MapTask to execute the map processing cycle (the MapReduce framework spawns 1 MapTask for each InputSplit generated by the InputFormat). In other words, the MapTask extracts the input data from the splits by using the RecordReader and InputFormat for the job, and it invokes the user provided map function, which emits a number of [key, value] pairs in the memory buffer.

After the MapTask finished executing all input records, the commit process cycle is initiated by flushing the memory buffer to the index and data file pair. The next step consists of merging all the index and data file pairs into a single construct that is (once again) being divided up into local directories. As some map tasks are completed, the JobTracker starts initiating the reduce tasks phase. The TaskTracker's involved in this step download the completed files from the map task nodes, and basically concatenate the files into a single entity. As more map tasks are being completed, the JobTracker notifies the involved TaskTracker's, requesting the download of the additional region files and to merge the files with the previous target file. Based on this design, the process of downloading the region files is interleaved with the on-going map task procedures.



Eventually, all the map tasks will be completed, at which point the JobTracker notifies the involved TaskTracker's to proceed with the reduce phase. Each TaskTracker will fork a ReduceTask (separate JVM's are used), read the downloaded file (that is already sorted by key), and invoke the reduce function that assembles the key and aggregated value structure into the final output file (there is one file per reducer node). Each reduce task (or map task) is single threaded, and this thread invokes the reduce [key, values] function in either ascending or descending order. The output of each reducer task is written to a temp file in HDFS. When the reducer finishes processing all keys, the temp file is atomically renamed into its final destination file name.

As the MapReduce library is designed to process vast amounts of data by potentially utilizing hundreds or thousands of nodes, the library has to be able to gracefully handle any failure scenarios. The TaskTracker nodes periodically report their status to the JobTracker that oversees the overall job progress. In scenarios where the JobTracker has not been contacted by a TaskTracker for a certain amount of time, the JobTracker assumes a TaskTracker node failure and hence, reassigns the tasks to other available TaskTracker nodes. As the results of the map phase are stored locally, the data will no longer be available if a TaskTracker node goes offline.

In such a scenario, all the map tasks from the failed node (regardless of the actual completion percentage) will have to be reassigned to a different TaskTracker node that will re-execute all the newly assigned splits. The results of the reduce phase are stored in HDFS and hence, the data is globally available even if a TaskTracker node goes offline. Hence, in a scenario where during the reduce phase a TaskTracker node goes offline, only the set of incomplete reduce tasks have to be reassigned to a different TaskTracker node for re-execution.

5. Explain the dataflow and control flow of MapReduce

MapReduce is the heart of Hadoop. It is a programming model designed for processing large volumes of data in parallel by dividing the work into a set of independent tasks.

The framework possesses the feature of data locality. Data locality means movement of algorithm to the data instead of data to algorithm. When the processing is done on the data algorithm is moved across the DataNodes rather than data to the algorithm. The architecture is so constructed because Moving Computation is Cheaper than Moving Data.

It is fault tolerant which is achieved by its daemons using the concept of replication.The daemons associated with the MapReduce phase are job-tracker and task-trackers.

Map-Reduce jobs are submitted on job-tracker. The JobTracker pushes work out to available TaskTracker nodes in the cluster, striving to keep the work as close to the data as possible. A heartbeat is sent from the TaskTracker to the JobTracker every few minutes to check its status whether the node is dead or alive. Whenever there is negative status, the job tracker assigns the task to another node on the replicated data of the failed node stored in this node.

Let's see how the data flows:

MapReduce has a simple model of data processing: inputs and outputs for the map and reduce functions are key-value pairs. The map and reduce functions in Hadoop MapReduce have the following general form:

map: (K1, V1) → list(K2, V2)

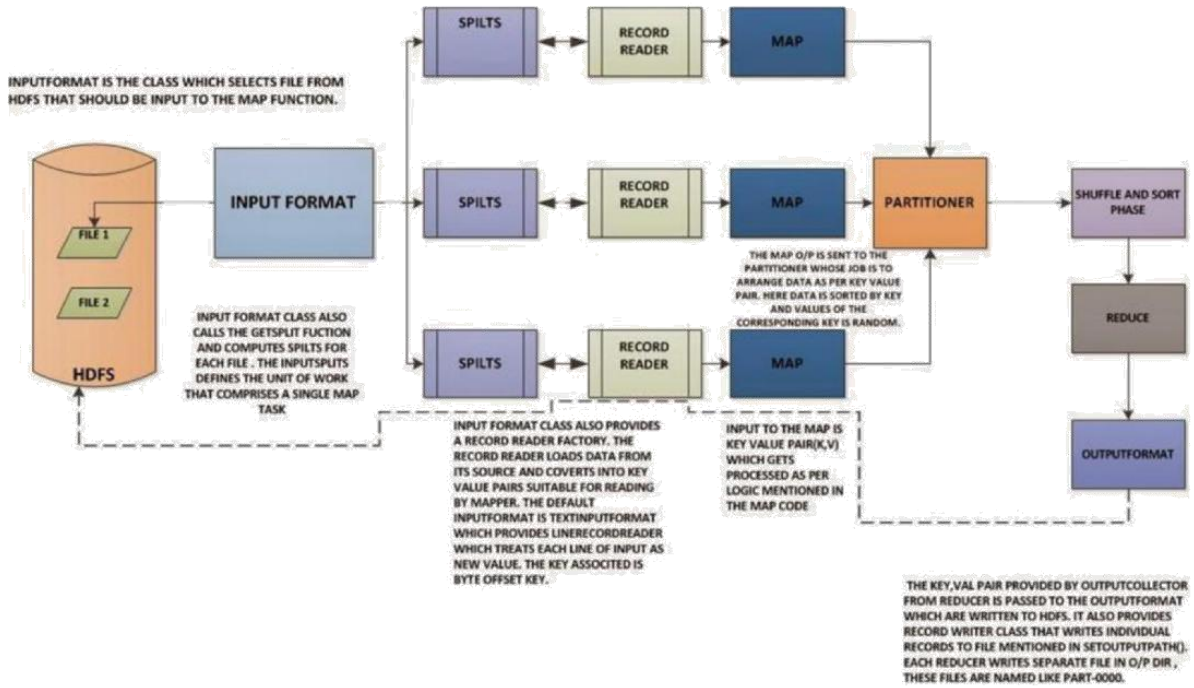
reduce: (K2, list(V2)) → list(K3, V3)

Now before processing it needs to know on which data to process, this is achieved with the InputFormat class. InputFormat is the class which selects file from HDFS that should be input to the map function. An InputFormat is also responsible for creating theinput splits and dividing them into records. The data is divided into number of splits (typically 64/128mb) in HDFS. An input split is a chunk of the input that is processed by a single map.

InputFormat class calls the getSplits() function and computes splits for each file and then sends them to the jobtracker, which uses their storage locations to schedule map tasks to process them on the tasktrackers. On a tasktracker, the map task passes the split to the createRecordReader() method on InputFormat to obtain a RecordReader for that split. The RecordReader loads data from its source and converts into key-value pairs suitable for reading by mapper. The default InputFormat is TextInputFormat which treats each value of input a new value and the associated key is byte offset.

A RecordReader is little more than an iterator over records, and the map task uses one to generate record key-value pairs, which it passes to the map function. We can see this by looking at the Mapper's run() method:

```
public void run(Context context) throws IOException,
InterruptedException { setup(context);
while (context.nextKeyValue()) {
map(context.getCurrentKey(), context.getCurrentValue(), context);
}
cleanup(context);
}
```



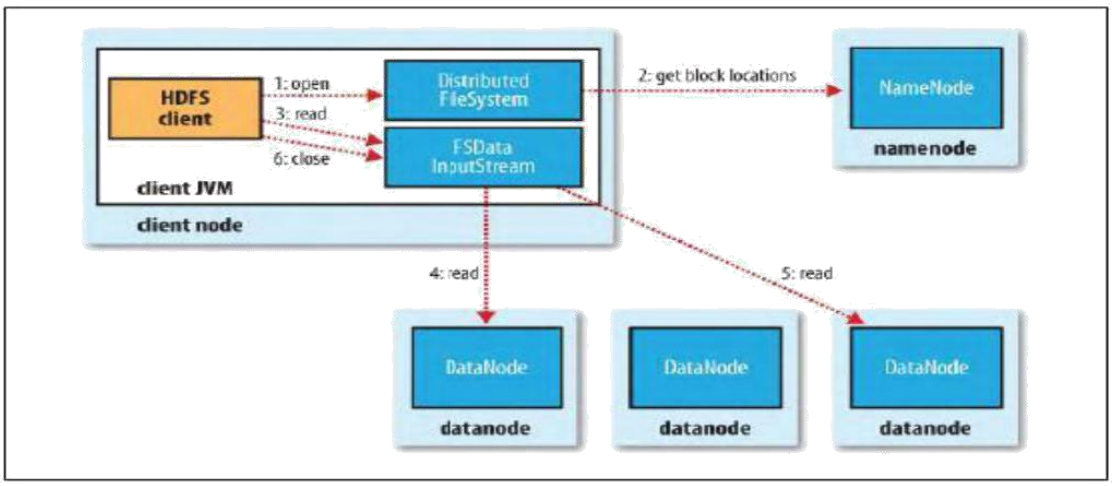
After running setup(), the nextKeyValue() is called repeatedly on the Context, (which delegates to the identically-named method on the the RecordReader) to populate the key and value objects for the mapper. The key and value are retrieved from the Record Reader by way of the Context, and passed to the map() method for it to do its work. Input to the map function which is the key-value pair (K, V) gets processed as per the logic mentioned in the map code.

When the reader gets to the end of the stream, the nextKeyValue() method returns false, and the map task runs its cleanup() method.

The output of the mapper is sent to the partitioner. Partitioner controls the partitioning of the keys of the intermediate map-outputs. The key (or a subset of the key) is used to derive the partition, typically by a hash function. The total number of partitions is the same as the number of reduce tasks for the job. Hence this controls which of the m reduce tasks the intermediate key (and hence the record) is sent for reduction. The use of partitioners is optional.

6. Describe in detail about dataflow of file read in HDFS

To get an idea of how data flows between the client interacting with HDFS, the namenode and the datanode, consider the below diagram, which shows the main sequence of events when reading a file.



A client reading data from HDFS

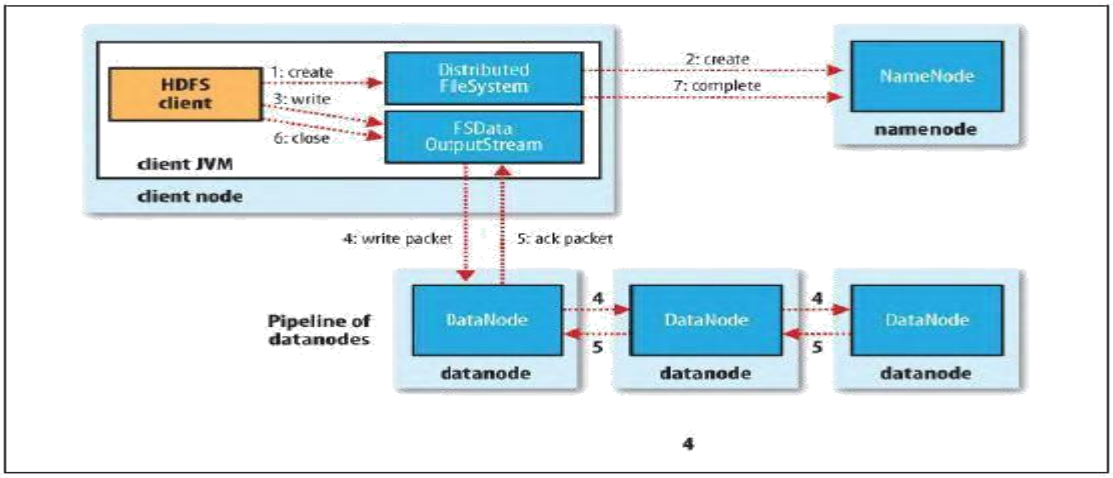
The client opens the file it wishes to read by calling open() on the FileSystem object, which for HDFS is an instance of DistributedFileSystem (step 1). DistributedFileSystem calls the namenode, using RPC, to determine the locations of the blocks for the first few blocks in the file (step 2). For each block, the namenode returns the addresses of the datanodes that have a copy of that block. Furthermore, the datanodes are sorted according to their proximity to the client. If the client is itself a datanode (in the case of a MapReduce task, for instance), then it will read from the local datanode.

The DistributedFileSystem returns a FSDataInputStream to the client for it to read data from. FSDataInputStream in turn wraps a DFSInputStream, which manages the datanode and namenode I/O. The client then calls read() on the stream (step 3). DFSInputStream, which has stored the datanode addresses for the first few blocks in the file, then connects to the first (closest) datanode for the first block in the file. Data is streamed from the datanode back to the client, which calls read() repeatedly on the stream (step 4). When the end of the block is reached, DFSInputStream will close the connection to the datanode, then find the best datanode for the next block (step 5). This happens transparently to the client, which from its point of view is just reading a continuous stream. Blocks are read in order with the DFSInputStream opening new connections to datanodes as the client reads through the stream. It will also call the namenode to retrieve the datanode locations for the next batch of blocks as needed. When the client has finished reading, it calls close() on the FSDataInputStream (step 6).

One important aspect of this design is that the client contacts datanodes directly to retrieve data, and is guided by the namenode to the best datanode for each block. This design allows HDFS to scale to large number of concurrent clients, since the data traffic is spread across all the datanodes in the cluster. The namenode meanwhile merely has to service block location requests (which it stores in memory, making them very efficient), and does not, for example, serve data, which would quickly become a bottleneck as the number of clients grew.

7. Describe in detail about dataflow of file write in HDFS

The case we're going to consider is the case of creating a new file, writing data to it, then closing the file



A client writing data to HDFS

The client creates the file by calling create() on DistributedFileSystem (step 1). DistributedFileSystem makes an RPC call to the namenode to create a new file in the filesystem's namespace, with no blocks associated with it (step 2). The namenode performs various checks to make sure the file doesn't already exist, and that the client has the right permissions to create the file. If these checks pass, the namenode makes a record of the new file; otherwise, file creation fails and the client is thrown an IOException. The DistributedFileSystem returns a SDataOutputStream for the client to start writing data to. Just as in the read case, FSDataOutputStream wraps a DFSOutputStream, which handles communication with the datanodes and namenode.

As the client writes data (step 3), DFSOutputStream splits it into packets, which it writes to an internal queue, called the *data queue*. The data queue is consumed by the DataStreamer, whose responsibility it is to ask the namenode to allocate new blocks by picking a list of suitable datanodes to store the replicas. The list of datanodes forms a pipeline—we'll assume the replication level is 3, so there are three nodes in the pipeline. The DataStreamer streams the packets to the first datanode in the pipeline, which stores the packet and forwards it to the second datanode in the pipeline. Similarly, the second datanode stores the packet and forwards it to the third (and last) datanode in the pipeline (step 4). DFSOutputStream also maintains an internal queue of packets that are waiting to be acknowledged by datanodes, called the *ack queue*. A packet is removed from the ack queue only when it has been acknowledged by all the datanodes in the pipeline (step 5).

If a datanode fails while data is being written to it, then the following actions are taken, which are transparent to the client writing the data. First the pipeline is closed, and any packets in the ack queue are added to the front of the data queue so that datanodes that are downstream from the failed node will not miss any packets. The current block on the good datanodes is given a new identity, which is communicated to the namenode, so that the partial block on the failed datanode will be deleted if the failed datanode recovers later on. The failed datanode is removed from the pipeline and the remainder of the block's data is written to the two good datanodes in the pipeline. The namenode notices that the block is under-replicated, and it arranges for a further replica to be created on another node. Subsequent blocks are then treated as normal.

When the client has finished writing data it calls close() on the stream (step 6). This action flushes all the remaining packets to the datanode pipeline and waits for acknowledgments before contacting the namenode to signal that the file is complete (step7). The namenode already knows which blocks the file is made up of (via Data Streamer asking for block allocations), so it only has to wait for blocks to be minimally replicated before returning successfully.

8. Explain Reading Data from a Hadoop URL and Deleting Data

The Hadoop's FileSystem class: the API for interacting with one of Hadoop's filesystems. While we focus mainly on the HDFS implementation, DistributedFileSystem, in general you should strive to write your code against the FileSystem abstract class, to retain portability across filesystems. This is very useful when testing your program.

One of the simplest ways to read a file from a Hadoop filesystem is by using a java.net.URL object to open a stream to read the data from. The general idiom is:

```
try {
in = new URL("hdfs://host/path").openStream();
3)process in } finally {

IOUtils.closeStream(in);

}
```

There's a little bit more work required to make Java recognize Hadoop's hdfs URL scheme. This is achieved by calling the setURLStreamHandlerFactory method on URL with an instance of FsUrlStreamHandlerFactory. This method can only be called once per JVM, so it is typically executed in a static block. This limitation means that if some other part of your program perhaps a third-party component outside your control sets RLStreamHandlerFactory, you won't be able to use this approach for reading data from Hadoop. The next section discusses an alternative. A program for displaying files from Hadoop filesystems on standard output, like the Unix cat command.

We make use of the handy IOUtils class that comes with Hadoop for closing the stream in the finally clause, and also for copying bytes between the input stream and the output stream (System.out in this case). The last two arguments to the copyBytes method are the buffer size used for copying, and whether to close the streams when the copy is complete. We close the input stream ourselves, and System.out doesn't need to be closed.

Displaying files from a Hadoop filesystem on standard output using a URLStreamHandler

```
public class URLCat {

  static {
    URL.setURLStreamHandlerFactory(new FsUrlStreamHandlerFactory());
  }

  public static void main(String[] args) throws Exception {
    InputStream in = null;
    try {
      in = new URL(args[0]).openStream();
      IOUtils.copyBytes(in, System.out, 4096, false);
    } finally {
      IOUtils.closeStream(in);
    }
  }
}
```

Deleting Data

Use the delete() method on FileSystem to permanently remove files or directories: public boolean delete(Path f, boolean recursive) throws IOException. If f is a file or an empty directory, then the value of recursive is ignored. A nonempty directory is only deleted, along with its contents, if recursive is true (otherwise an IOException is thrown).

9.(a). File pattern in HDFS

It is a common requirement to process sets of files in a single operation. For example, a MapReduce job for log processing might analyze a month worth of files, contained in a number of directories. Rather than having to enumerate each file and directory to specify the input, it is convenient to use wildcard characters to match multiple files with a single expression, an operation that is known as *globbing*. Hadoop provides two FileSystem methods for processing globs:

public FileStatus[] globStatus(Path pathPattern) throws IOException

public FileStatus[] globStatus(Path pathPattern, PathFilter filter) throws IOException

The globStatus() methods returns an array of FileStatus objects whose paths match the supplied pattern, sorted by path. An optional PathFilter can be specified to restrict the matches further.

Glob characters and their meanings

Glob	Name	Matches
*	*asterisk*	Matches zero or more characters
?	*question mark*	Matches a single character
[ab]	*character class*	Matches a single character in the set {a, b}
[^ab]	*negated character class*	Matches a single character that is not in the set {a, b}
[a-b]	*character range*	Matches a single character in the (closed) range [a, b], where a is lexicographically less than or equal to b
[^a-b]	*negated character range*	Matches a single character that is not in the (closed) range [a, b], where a is lexicographically less than or equal to b
{a,b}	*alternation*	Matches either expression a or b
\c	*escaped character*	Matches character c when it is a metacharacter

Hadoop supports the same set of glob characters as Unix *bash.*

Imagine that logfiles are stored in a directory structure organized hierarchically by date. So, for example, logfiles for the last day of 2007 would go in a directory named */2007/12/31*. Suppose that the full file listing is:

/2007/12/30
/2007/12/31
/2008/01/01
/2008/01/02

9. (b) Pathfilter

Glob patterns are not always powerful enough to describe a set of files you want to access. For example, it is not generally possible to exclude a particular file using a glob pattern. The listStatus() and globStatus() methods of FileSystem take an optional PathFilter, which allows programmatic control over matching:

```
package org.apache.hadoop.fs;
public interface PathFilter
{
boolean accept(Path path);
}
```

PathFilter is the equivalent of java.io.FileFilter for Path objects rather than File objects.

A PathFilter for excluding paths that match a regular expression

```
public class RegexExcludePathFilter implements PathFilter {

  private final String regex;

  public RegexExcludePathFilter(String regex) {
    this.regex = regex;
  }

  public boolean accept(Path path) {
    return !path.toString().matches(regex);
  }
}
```

The filter passes only files that *don't* match the regular expression. We use the filter in conjunction with a glob that picks out an initial set of files to include: the filter is used to refine the results. For example:

fs.globStatus(new Path("/2007/*/*"), new RegexExcludeFilter("^.*/2007/12/31$"))

Will expand to */2007/12/30*. Filters can only act on a file's name, as represented by a Path. They can't use a file's properties, such as creation time, as the basis of the filter. Nevertheless, they can perform matching that neither glob patterns nor regular expressions can achieve.

10. Explain in detail about command line interface in HDFS

There are many other interfaces to HDFS, but the command line is one of the simplest, and to many developers the most familiar. We are going to run HDFS on one machine, so first follow the instructions for setting up Hadoop in pseudo-distributed mode.Later you'll see how to run on a cluster of machines to give us scalability and fault tolerance.

There are two properties that we set in the pseudo-distributed configuration that deserve further explanation. The first is fs.default.name, set to *hdfs://localhost/*, which is used to set a default filesystem for Hadoop. Filesystems are specified by a URI, and here we have used a hdfs URI to configure Hadoop to use HDFS by default. The HDFS daemons will use this property to determine the host and port for the HDFS namenode. We'll be running it on localhost, on the default HDFS port, 8020. And HDFS clients will use this property to work out where the namenode is running so they can connect to it.

We set the second property, dfs.replication, to one so that HDFS doesn't replicate filesystem blocks by the usual default of three. When running with a single datanode, HDFS can't replicate blocks to three datanodes, so it would perpetually warn about blocks being under-replicated. This setting solves that problem.

Basic Filesystem Operations

The filesystem is ready to be used, and we can do all of the usual filesystem operations such as reading files, creating directories, moving files, deleting data, and listing directories. You can type hadoop fs -help to get detailed help on every command. Start by copying a file from the local filesystem to HDFS:

2 **hadoop fs -copyFromLocal input/docs/quangle.txt hdfs://localhost/user/tom/quangle.txt**

This command invokes Hadoop's filesystem shell command fs, which supports a number of subcommands—in this case, we are running -copyFromLocal. The local file *quangle.txt* is copied to the file */user/tom/quangle.txt* on the HDFS instance running on localhost. In fact, we could have omitted the scheme and host of the URI and picked up the default, hdfs://localhost, as specified in *core-site.xml*.

4) **hadoop fs -copyFromLocal input/docs/quangle.txt /user/tom/quangle.txt**

We could also have used a relative path, and copied the file to our home directory in HDFS, which in this case is */user/tom*:

6) **hadoop fs -copyFromLocal input/docs/quangle.txt quangle.txt**

Let's copy the file back to the local filesystem and check whether it's the same:

8) **hadoop fs -copyToLocal quangle.txt quangle.copy.txt**

9) **md5 input/docs/quangle.txt quangle.copy.txt**

MD5 (input/docs/quangle.txt) = a16f231da6b05e2ba7a339320e7dacd9

MD5 (quangle.copy.txt) = a16f231da6b05e2ba7a339320e7dacd9

The MD5 digests are the same, showing that the file survived its trip to HDFS and is back intact.

Finally, let's look at an HDFS file listing. We create a directory first just to see how it is displayed in the listing:

10) **hadoop fs -mkdir books**

11) **hadoop fs -ls .**

Found 2 items

drwxr-xr-x - tom supergroup 0 2009-04-02 22:41 /user/tom/books

-rw-r--r-- 1 tom supergroup 118 2009-04-02 22:29 /user/tom/quangle.txt

The information returned is very similar to the Unix command ls -l, with a few minor differences. The first column shows the file mode. The second column is the replication factor of the file (something a traditional Unix filesystems does not have). Remember we set the default replication factor in the site-wide configuration to be 1, which is why we see the same value here. The entry in this column is empty for directories since the concept of replication does not apply to them—directories are treated as metadata and stored by the namenode, not the datanodes. The third and fourth columns show the file owner and group. The fifth column is the size of the file in bytes, or zero for directories. The six and seventh columns are the last modified date and time. Finally, the eighth column is the absolute name of the file or directory.

Unit – 5 - Security

Part – A

1. What are the challenges of grid sites

- ✓
- ✓ The first challenge is integration with existing systems and technologies.
- ✓ The second challenge is interoperability with different hosting environments.

The third challenge is to construct trust relationships among interacting hosting environments.

2. Define Reputation-Based Trust Model

In a reputation-based model, jobs are sent to a resource site only when the site is trustworthy to meet users' demands. The site trustworthiness is usually calculated from the following information: the defense capability, direct reputation, and recommendation trust.

3. Define direct reputation

Direct reputation is based on experiences of prior jobs previously submitted to the site. The reputation is measured by many factors such as prior job execution success rate, cumulative site utilization, job turnaround time, job slowdown ratio, and so on. A positive experience associated with a site will improve its reputation. On the contrary, a negative experience with a site will decrease its reputation.

4. What are the major authentication methods in the grid?

The major authentication methods in the grid include passwords, PKI, and Kerberos. The password is the simplest method to identify users, but the most vulnerable one to use. The PKI is the most popular method supported by GSI.

5. List the types of authority in grid

The authority can be classified into three categories: attribute authorities, policy authorities, and identity authorities. Attribute authorities issue attribute assertions; policy authorities issue authorization policies; identity authorities issue certificates. The authorization server makes the final authorization decision.

6. Define grid security infrastructure

The Grid Security Infrastructure (GSI), formerly called the Globus Security Infrastructure, is a specification for secret, tamper-proof, delegatable communication between software in a grid computing environment. Secure, authenticatable communication is enabled using asymmetric encryption.

7. What are the functions present in GSI

GSI may be thought of as being composed of four distinct functions: message protection, authentication, delegation, and authorization.

8. List the protection mechanisms in GSI

GSI allows three additional protection mechanisms. The first is integrity protection, by which a receiver can verify that messages were not altered in transit from the sender. The second is encryption, by which messages can be protected to provide confidentiality. The third is replay prevention, by which a receiver can verify that it has not.

9. What is the primary information of GSI

GSI authentication, a certificate includes four primary pieces of information: (1) a subject name, which identifies the person or object that the certificate represents; (2) the public key belonging to the subject; 12) the identity of a CA that has signed the certificate to certify that the public key and the identity both belong to the subject; and (4) the digital signature of the named CA.

10. Define blue pill

The blue pill is malware that executes as a hypervisor to gain control of computer resources. The hypervisor installs without requiring a restart and the computer functions normally, without degradation of speed or services, which makes detection difficult.

11. What are the host security threats in public IaaS

1. Stealing keys used to access and manage hosts (e.g., SSH private keys)
2. Attacking unpatched, vulnerable services listening on standard ports (e.g., FTP, SSH)
3. Hijacking accounts that are not properly secured (i.e., no passwords for standard accounts)
4. Attacking systems that are not properly secured by host firewalls
5. Deploying Trojans embedded in the software component in the VM or within the VM image (the OS) itself

12. List the Public Cloud Security Limitations

2 There are limitations to the public cloud when it comes to support for custom security features. Security requirements such as an application firewall, SSL accelerator, cryptography, or rights management using a device that supports PKCS 12 are not supported in a public SaaS, PaaS, or IaaS cloud.

14) Any mitigation controls that require deploymentof an appliance or locally attached peripheral devices in the public IaaS/PaaS cloud are not feasible.

13. Define Data lineage

Data lineage is defined as a data life cycle that includes the data's origins and where it moves over time. It describes what happens to data as it goes through diverse processes. It helps provide visibility into the analytics pipeline and simplifies tracing errors back to their sources.

14. Define Data remanence

Data remanence is the residual representation of data that has been in some way nominally erased or removed.

15. What are the IAM processes operational activities.

- ✓
- ✓ Provisioning
  Credential and attribute management
- ✓
- ✓ Entitlement management
- ✓ Compliance management
  Identity federation management

16. What are the functions of Cloud identity administrative

Cloud identity administrative functions should focus on life cycle management of user identities in the cloud—provisioning, deprovisioning, identity federation, SSO, password or credentials management, profile management, and administrative management. Organizations that are not capable of supporting federation should explore cloud-based identity management services.

17. List the factors to manage the IaaS virtual infrastructure in the cloud

✓

✓ Availability of a CSP network, host, storage, and support application infrastructure.

Availability of your virtual servers and the attached storage (persistent and ephemeral) for compute services

✓ Availability of virtual storage that your users and virtual server depend on for storage Service

✓ Availability of your network connectivity to the Internet or virtual network connectivity to IaaS services.

✓ Availability of network services

18. What is meant by the terms data-in-transit

It is the process of the transfer of the data between all of the versions of the original file, especially when data may be in transit on the Internet. It is data that is exiting the network via email, web, or other Internet protocols.

19. List the IAM process business category

- User management

1. Authentication management
2. Authorization management
3. Access management
4. Data management and provisioning

21) • Monitoring and auditing

6) What are the key components of IAM automation process?

User Management, New Users

User Management, User Modifications ✓

Authentication Management

✓ Authorization Management

Part – B

1. Trust Models for Grid Security

A user job demands the resource site to provide security assurance by issuing a security demand (SD). On the other hand, the site needs to reveal its trustworthiness, called its trust index (TI). These two parameters must satisfy a security-assurance condition: TI ≥ SD during the job mapping process. When determining its security demand, users usually care about some typical attributes. These attributes and their values are dynamically changing and depend heavily on the trust model, security policy, accumulated reputation, self-defense capability, attack history, and site vulnerability.

Three challenges are outlined below to establish the trust among grid sites

The first challenge is integration with existing systems and technologies. The resources sites in a grid are usually heterogeneous and autonomous. It is unrealistic to expect that a single type of security can be compatible with and adopted by every hosting environment. At the same time, existing security infrastructure on the sites cannot be replaced overnight. Thus, to be successful, grid security architecture needs to step up to the challenge of integrating with existing security architecture and models across platforms and hosting environments.

The second challenge is interoperability with different ―hosting environments.‖ Services are often invoked across multiple domains, and need to be able to interact with one another. The interoperation is demanded at the protocol, policy, and identity levels. For all these levels, interoperation must be protected securely. The third challenge is to construct trust relationships among interacting hosting environments. Grid service requests can be handled by combining resources on multiple security domains. Trust relationships are required by these domains during the end-to-end traversals. A service needs to be open to friendly and interested entities so that they can submit requests and access securely.

The grid aims to construct a large-scale network computing system by integrating distributed, heterogeneous, and autonomous resources. The security challenges faced by the grid are much greater than other computing systems. Before any effective sharing and cooperation occurs, a trust relationship has to be established among participants. A Generalized Trust Model

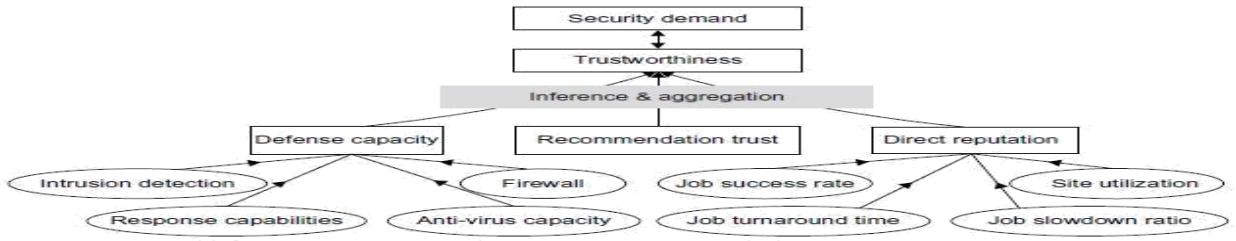
At the bottom, we identify three major factors which influence the trustworthiness of a resource site. An inference module is required to aggregate these factors. Followings are some existing inference or aggregation methods. An intra-site fuzzy inference procedure is called to

assess defense capability and direct reputation. Defense capability is decided by the firewall, intrusion detection system (IDS), intrusion response capability, and anti-virus capacity of the individual resource site. Direct reputation is decided based on the job success rate, site utilization, job turnaround time, and job slowdown ratio measured. Recommended trust is also known as secondary trust and is obtained indirectly over the grid network. Reputation-Based Trust Model

In a reputation-based model, jobs are sent to a resource site only when the site is trustworthy to meet users' demands. The site trustworthiness is usually calculated from the following information: the defense capability, direct reputation, and recommendation trust. The defense capability refers to the site's ability to protect itself from danger. It is assessed according to such factors as intrusion detection, firewall, response capabilities, anti-virus capacity, and so on. Direct reputation is based on experiences of prior jobs previously submitted to the site. The reputation is measured by many factors such as prior job execution success rate, cumulative site utilization, job turnaround time, job slowdown ratio, and so on. A positive experience associated with a site will improve its reputation. On the contrary, a negative experience with a site will decrease its reputation.

A Fuzzy-Trust Model

The job security demand (SD) is supplied by the user programs. The trust index (TI) of a resource site is aggregated through the fuzzy-logic inference process over all related parameters. Specifically, one can use a two-level fuzzy logic to estimate the aggregation of numerous trust parameters and security attributes into scalar quantities that are easy to use in the job scheduling and resource mapping process. The TI is normalized as a single real number with 0 representing



A general trust model for grid computing.

the condition with the highest risk at a site and 1 representing the condition which is totally risk-free or fully trusted. The fuzzy inference is accomplished through four steps: fuzzification, inference, aggregation, and defuzzification. The second salient feature of the trust model is that if a site's trust index cannot match the job security demand (i.e., SD > TI), the trust model could deduce detailed security features to guide the site security upgrade as a result of tuning the fuzzy system.

2. Authentication and Authorization Methods

The major authentication methods in the grid include passwords, PKI, and Kerberos. The password is the simplest method to identify users, but the most vulnerable one to use. The PKI is the most popular method supported by GSI. To implement PKI, we use a trusted third party, called the certificate authority (CA). Each user applies a unique pair of public and private keys. The public keys are issued by the CA by issuing a certificate, after recognizing a legitimate user. The private key is exclusive for each user to use, and is unknown to any other users. A digital certificate in IEEE X.509 format consists of the user name, user public key, CA name, and a secrete signature of the user. The following example illustrates the use of a PKI service in a grid environment.

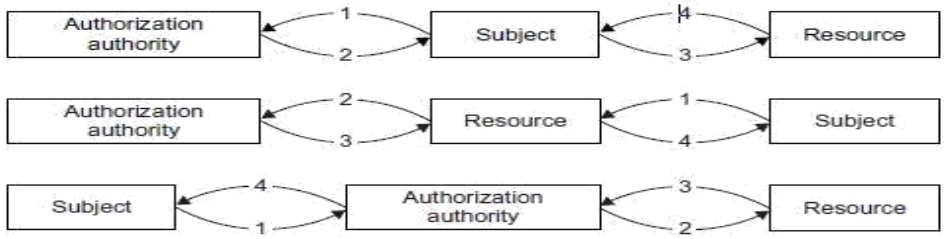
Authorization for Access Control

The authorization is a process to exercise access control of shared resources. Decisions can be made either at the access point of service or at a centralized place. Typically, the resource is a host that provides processors and storage for services deployed on it. Based on a set predefined policies or rules, the resource may enforce access for local services. The central authority is a

special entity which is capable of issuing and revoking polices of access rights granted to remote accesses. The authority can be classified into three categories: attribute authorities, policy authorities, and identity authorities. Attribute authorities issue attribute assertions; policy authorities issue authorization policies; identity authorities issue certificates. The authorization server makes the final authorization decision.

Three Authorization Models

Three authorization models are shown in diagram. The subject is the user and the resource refers to the machine side. The subject-push model is shown at the top diagram. The user conducts handshake with the authority first and then with the resource site in a sequence. The resource-pulling model puts the resource in the middle. The user checks the resource first. Then the resource contacts its authority to verify the request, and the authority authorizes at step 3. Finally the resource accepts or rejects the request from the subject at step 4. The authorization agent model puts the authority in the middle. The subject check with the authority at step 1 and the authority makes decisions on the access of the requested resources. The authorization process is complete at steps 3 and 4 in the reverse direction.



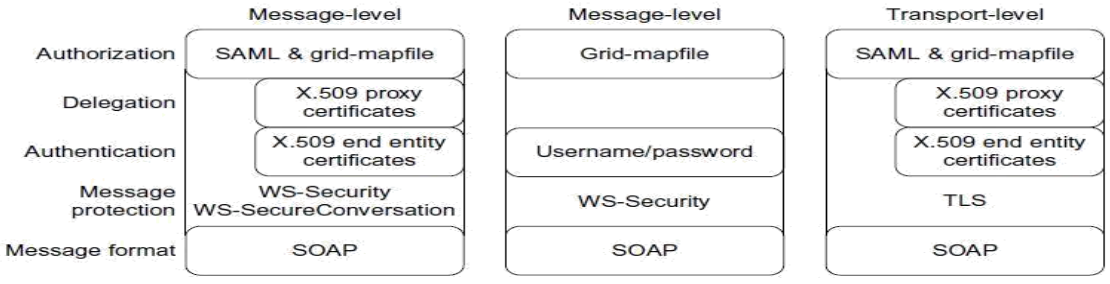
Three authorization models: the subject-push model, resource-pulling model, and the authorization agent model.

3. Explain in detail about Grid Security Infrastructure

GSI is a portion of the Globus Toolkit and provides fundamental security services needed to support grids, including supporting for message protection, authentication and delegation, and authorization. GSI enables secure authentication and communication over an open network, and permits mutual authentication across and among distributed sites with single sign-on capability. No centrally managed security system is required, and the grid maintains the integrity of its members' local policies. GSI supports both message-level security, which supports the WS-Security standard and the WS-Secure Conversation specification to provide message protection for SOAP messages, and transport-level security, which means authentication via TLS with support for X.509 proxy certificates.

GSI Functional Layers

GT4 provides distinct WS and pre-WS authentication and authorization capabilities. Both build on the same base, namely the X.509 standard and entity certificates and proxy certificates, which are used to identify persistent entities such as users and servers and to support the temporary delegation of privileges to other entities, respectively. As shown in diagram, GSI may be thought of as being composed of four distinct functions: message protection, authentication, delegation, and authorization.



GSI functional layers at the message and transport levels.

Transport-Level Security

Transport-level security entails SOAP messages conveyed over a network connection protected by TLS. TLS provides for both integrity protection and privacy (via encryption). Transport-level security is normally used in conjunction with X.509 credentials for authentication, but can also be used without such credentials to provide message protection without authentication, often referred to as ―anonymous transport-level security.‖ In this mode of operation, authentication may be done by username and password in a SOAP message

GSI also provides message-level security for message protection for SOAP messages by implementing the WS-Security standard and the WS-Secure Conversation specification. The WS-Security standard from OASIS defines a framework for applying security to individual SOAP messages; WS-Secure Conversation is a proposed standard from IBM and Microsoft that allows for an initial exchange of messages to establish a security context which can then be used to protect subsequent messages in a manner that requires less computational overhead (i.e., it allows the trade-off of initial overhead for setting up the session for lower overhead for messages).

GSI conforms to this standard. GSI uses these mechanisms to provide security on a per-message basis, that is, to an individual message without any preexisting context between the sender and receiver (outside of sharing some set of trust roots). GSI, as described further in the subsequent section on authentication, allows for both X.509 public key credentials and the combination of username and password for authentication; however, differences still exist. With username/password, only the WS-Security standard can be used to allow for authentication; that is, a receiver can verify the identity of the communication initiator.

GSI allows three additional protection mechanisms. The first is integrity protection, by which a receiver can verify that messages were not altered in transit from the sender. The second is encryption, by which messages can be protected to provide confidentiality. The third is replay prevention, by which a receiver can verify that it has not received the same message previously. These protections are provided between WS-Security and WS-Secure Conversation. The former applies the keys associated with the sender and receiver's X.509 credentials. The X.509 credentials are used to establish a session key that is used to provide the message protection.

Authentication and Delegation

GSI has traditionally supported authentication and delegation through the use of X.509 certificates and public keys. As a new feature in GT4, GSI also supports authentication through plain usernames and passwords as a deployment option. We discuss both methods in this section. GSI uses X.509 certificates to identify persistent users and services.

As a central concept in GSI authentication, a certificate includes four primary pieces of information: (1) a subject name, which identifies the person or object that the certificate represents; (2) the public key belonging to the subject; (3) the identity of a CA that has signed the certificate to certify that the public key and the identity both belong to the subject; and (4) the digital signature of the named CA. X.509 provides each entity with a unique identifier (i.e., a distinguished name) and a method to assert that identifier to another party through the use of an asymmetric key pair bound to the identifier by the certificate.

Trust Delegation

To reduce or even avoid the number of times the user must enter his passphrase when several grids are used or have agents (local or remote) requesting services on behalf of a user, GSI provides a delegation capability and a delegation service that provides an interface to allow clients to delegate (and renew) X.509 proxy certificates to a service. The interface to this service is based on the WS-Trust specification. A proxy consists of a new certificate and a private key. The key pair that is used for the proxy, that is, the public key embedded in the certificate and the private key, may either be regenerated for each proxy or be obtained by other means. The new certificate contains the owner's identity, modified slightly to indicate that it is a proxy. The new certificate is signed by the owner, rather than a CA

4. Explain cloud infrastructure security at application level

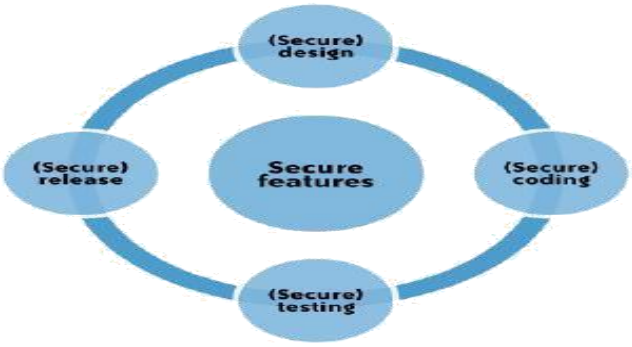
We will limit our discussion to web application security: web applications in the cloud accessed by users with standard Internet browsers, such as Firefox, Internet Explorer, or Safari, from any computer connected to the Internet.

Application-Level Security Threats

The existing threats exploit well-known application vulnerabilities including cross-site scripting (XSS), SQL injection, malicious file execution, and other vulnerabilities resulting from programming errors and design flaws. Armed with knowledge and tools, hackers are constantly scanning web applications (accessible from the Internet) for application vulnerabilities.

It has been a common practice to use a combination of perimeter security controls and network- and host-based access controls to protect web applications deployed in a tightly controlled environment, including corporate intranets and private clouds, from external hackers.

Web applications built and deployed in a public cloud platform will be subjected to a high threat level, attacked, and potentially exploited by hackers to support fraudulent and illegal activities. In that threat model, web applications deployed in a public cloud (the SPI model) must be designed for an Internet threat model, and security must be embedded into the Software Development Life Cycle (SDLC)



DoS and EDoS

Additionally, you should be cognizant of application-level DoS and EDDoS attacks that can potentially disrupt cloud services for an extended time. These attacks typically originate from compromised computer systems attached to the Internet.

Application-level DoS attacks could manifest themselves as high-volume web page reloads, XML* web services requests (over HTTP or HTTPS), or protocol-specific requests supported by a cloud service. Since these malicious requests blend with the legitimate traffic, it is extremely difficult to selectively filter the malicious traffic without impacting the service as a whole

DoS attacks on pay-as-you-go cloud applications will result in a dramatic increase in your cloud utility bill: you'll see increased use of network bandwidth, CPU, and storage consumption. This type of attack is also being characterized as *economic denial of sustainability* (EDoS)

End User Security

A customer of a cloud service, are responsible for end user security tasks—security procedures to protect your Internet-connected PC—and for practicing ―safe surfing.‖ Protection measures include use of security software, such as anti-malware, antivirus, personal firewalls, security patches, and IPS-type software on your Internet-connected computer.

The new mantra of ―the browser is your operating system‖ appropriately conveys the message that browsers have become the ubiquitous ―operating systems‖ for consuming cloud services. All Internet browsers routinely suffer from software vulnerabilities that make them vulnerable to end user security attacks. Hence, our recommendation is that cloud customers take appropriate steps to protect browsers from attacks. To achieve end-to-end security in a cloud, it is essential for customers to maintain good browser hygiene. The means keeping the browser

(e.g., Internet Explorer, Firefox, Safari) patched and updated to mitigate threats related to browser vulnerabilities.

Currently, although browser security add-ons are not commercially available, users are encouraged to frequently check their browser vendor's website for security updates, use the auto-update feature, and install patches on a timely basis to maintain end user security.

SaaS Application Security

The SaaS model dictates that the provider manages the entire suite of applications delivered to users. Therefore, SaaS providers are largely responsible for securing the applications and components they offer to customers. Customers are usually responsible for operational security functions, including user and access management as supported by the provider.

Extra attention needs to be paid to the authentication and access control features offered by SaaS CSPs. Usually that is the only security control available to manage risk to information. Most services, including those from Salesforce.com and Google, offer a web-based administration user interface tool to manage authentication and access control of the application.

Cloud customers should try to understand cloud-specific access control mechanisms—including support for strong authentication and privilege management based on user roles and functions—and take the steps necessary to protect information hosted in the cloud. Additional controls should be implemented to manage privileged access to the SaaS administration tool, and enforce segregation of duties to protect the application from insider threats. In line with security standard practices, customers should implement a strong password policy—one that forces users to choose strong passwords when authenticating to an application.

PaaS Application Security

PaaS vendors broadly fall into the following two major categories:

- Software vendors (e.g., Bungee, Etelos, GigaSpaces, Eucalyptus)
- CSPs (e.g., Google App Engine, Salesforce.com's Force.com, Microsoft Azure, Intuit QuickBase)

A PaaS cloud (public or private) offers an integrated environment to design, develop, test, deploy, and support custom applications developed in the language the platform supports. PaaS application security encompasses two software layers:

- Security of the PaaS platform itself (i.e., runtime engine)
- Security of customer applications deployed on a PaaS platform

PaaS CSPs (e.g., Google, Microsoft, and Force.com) are responsible for securing the platform software stack that includes the runtime engine that runs the customer applications. Since PaaS applications may use third-party applications, components, or web services, the third-party application provider may be responsible for securing their services. Hence, customers should understand the dependency of their application on all services and assess risks pertaining to third-party service providers.

IaaS Application Security

IaaS cloud providers (e.g., Amazon EC2, GoGrid, and Joyent) treat the applications on customer virtual instances as a black box, and therefore are completely agnostic to the operations and management of the customer's applications.

The entire stack—customer applications, runtime application platform (Java, .NET, PHP, Ruby on Rails, etc.), and so on— runs on the customer's virtual servers and is deployed and managed by customers. To that end, customers have full responsibility for securing their applications deployed in the IaaS cloud.

Web applications deployed in a public cloud must be designed for an Internet threat model, embedded with standard security countermeasures against common web vulnerabilities. In adherence with common security development practices, they should also be periodically tested for vulnerabilities, and most importantly, security should be embedded into the SDLC. Customers are solely responsible for keeping their applications and runtime platform patched to protect the system from malware and hackers scanning for vulnerabilities to gain unauthorized access to their data in the cloud. It is highly recommended that you design and implement applications with a ―least-privileged‖ runtime model

Developers writing applications for IaaS clouds must implement their own features to handle authentication and authorization. In line with enterprise identity management practices, cloud applications should be designed to leverage delegated authentication service features supported by an enterprise Identity Provider (e.g., OpenSSO, Oracle IAM, IBM, CA) or third-party identity service provider (e.g., Ping Identity, Symplified, TriCipher). Any custom implementations of Authentication, Authorization, and Accounting (AAA) features can become a weak link if they are not properly implemented, and you should avoid them when possible.

5. Describe in detail about provider data and its security

Customers should also be concerned about what data the provider collects and how the CSP protects that data. Additionally, your provider collects and must protect a huge amount of security-related data.

Storage

For data stored in the cloud (i.e., storage-as-a-service), we are referring to IaaS and not data associated with an application running in the cloud on PaaS or SaaS. The same three information security concerns are associated with this data stored in the cloud (e.g., Amazon's S3) as with data stored elsewhere: confidentiality, integrity, and availability.

Confidentiality

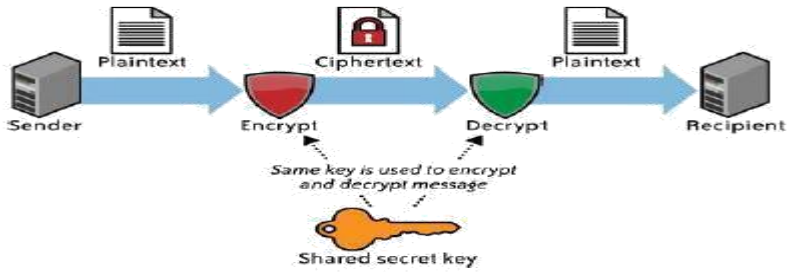
When it comes to the confidentiality of data stored in a public cloud, you have two potential concerns. First, what access control exists to protect the data? Access control consists of both authentication and authorization.

CSPs generally use weak authentication mechanisms (e.g., username + password), and the authorization (—access‖) controls available to users tend to be quite coarse and not very granular. For large organizations, this coarse authorization presents significant security concerns unto itself. Often, the only authorization levels cloud vendors provide are administrator authorization (i.e., the owner of the account itself) and user authorization (i.e., all other authorized users)—with no levels in between (e.g., business unit administrators, who are authorized to approve access for their own business unit personnel).

If a CSP does encrypt a customer's data, the next consideration concerns what encryption algorithm it uses. Not all encryption algorithms are created equal. Cryptographically, many algorithms provide insufficient security. Only algorithms that have been publicly vetted by a formal standards body (e.g., NIST) or at least informally by the cryptographic community should be used. Any algorithm that is proprietary should absolutely be avoided.

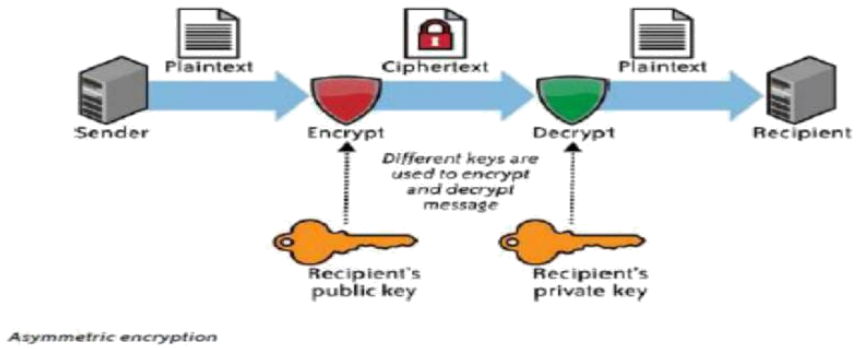
Symmetric encryption involves the use of a single secret key for both the encryption and decryption of data. Only symmetric encryption has the speed and computational efficiency to handle encryption of large volumes of data. It would be highly unusual to use an asymmetric algorithm for this encryption use case.

Although the example in diagram is related to email, the same concept (i.e., a single shared, secret key) is used in data storage encryption.



Symmetric encryption

Although the example in diagram is related to email, the same concept (i.e., a public key and a private key) is *not* used in data storage encryption.



Asymmetric encryption

Integrity

Confidentiality does not imply integrity; data can be encrypted for confidentiality purposes, and yet you might not have a way to verify the integrity of that data. Encryption alone is sufficient for confidentiality, but integrity also requires the use of message authentication codes (MACs). The simplest way to use MACs on encrypted data is to use a block symmetric algorithm (as opposed to a streaming symmetric algorithm) in cipher block chaining (CBC) mode, and to include a one-way hash function.

Another aspect of data integrity is important, especially with bulk storage using IaaS. Once a customer has several gigabytes (or more) of its data up in the cloud for storage, how does the customer check on the integrity of the data stored there? There are IaaS transfer costs associated with moving data into and back down from the cloud,* as well as network utilization (bandwidth) considerations for the customer's own network. What a customer really wants to do is to validate the integrity of its data while that data remains in the cloud—without having to download and reupload that data.

Availability

Assuming that a customer's data has maintained its confidentiality and integrity, you must also be concerned about the availability of your data. There are currently three major threats in this regard—none of which are new to computing, but all of which take on increased importance in cloud computing because of increased risk.

The first threat to availability is network-based attacks

The second threat to availability is the CSP's own availability.

Cloud storage customers must be certain to ascertain just what services their provider is actually offering. Cloud storage does not mean the stored data is actually backed up. Some cloud storage providers do back up customer data, in addition to providing storage. However, many cloud storage providers do not back up customer data, or do so only as an additional service for an additional cost.

All three of these considerations (confidentiality, integrity, and availability) should be encapsulated in a CSP's service-level agreement (SLA) to its customers. However, at this time, CSP SLAs are extremely weak—in fact, for all practical purposes, they are essentially worthless. Even where a CSP appears to have at least a partially sufficient SLA, how that SLA actually gets measured is problematic. For all of these reasons, data security considerations and how data is actually stored in the cloud should merit considerable attention by customers.

6. Explain identity and access management functional architecture

We'll present the basic concepts and definitions of IAM functions for any service:

Authentication

Authentication is the process of verifying the identity of a user or system. Authentication usually connotes a more robust form of identification. In some use cases, such as service-to-service interaction, authentication involves verifying the network service requesting access to information served by another service.

Authorization

Authorization is the process of determining the privileges the user or system is entitled to once the identity is established. —in other words, authorization is the process of enforcing policies.

Auditing

In the context of IAM, auditing entails the process of review and examination of authentication, authorization records, and activities to determine the adequacy of IAM system controls, to verify compliance with established security policies and procedures (e.g., separation of duties), to detect breaches in security services (e.g., privilege escalation), and to recommend any changes that are indicated for countermeasures.

IAM Architecture

Standard enterprise IAM architecture encompasses several layers of technology, services, and processes. At the core of the deployment architecture is a directory service (such as LDAP or Active Directory) that acts as a repository for the identity, credential, and user attributes of the organization's user pool. The directory interacts with IAM technology components such as authentication, user management, provisioning, and federation services that support the standard IAM practice and processes within the organization. It is not uncommon for organizations to use several directories that were deployed for environment-specific reasons (e.g., Windows systems using Active Directory, Unix systems using LDAP) or that were integrated into the environment by way of business mergers and acquisitions.

The IAM processes to support the business can be broadly categorized as follows:

User management

Activities for the effective governance and management of identity life cycles

Authentication management

Activities for the effective governance and management of the process for determining that an entity is who or what it claims to be

Authorization management

Activities for the effective governance and management of the process for determining entitlement rights that decide what resources an entity is permitted to access in accordance with the organization's policies.

Access management

Enforcement of policies for access control in response to a request from an entity (user, services) wanting to access an IT resource within the organization

Data management and provisioning

Propagation of identity and data for authorization to IT resources via automated or manual processes

Monitoring and auditing

Monitoring, auditing, and reporting compliance by users regarding access to resources within the organization based on the defined policies.

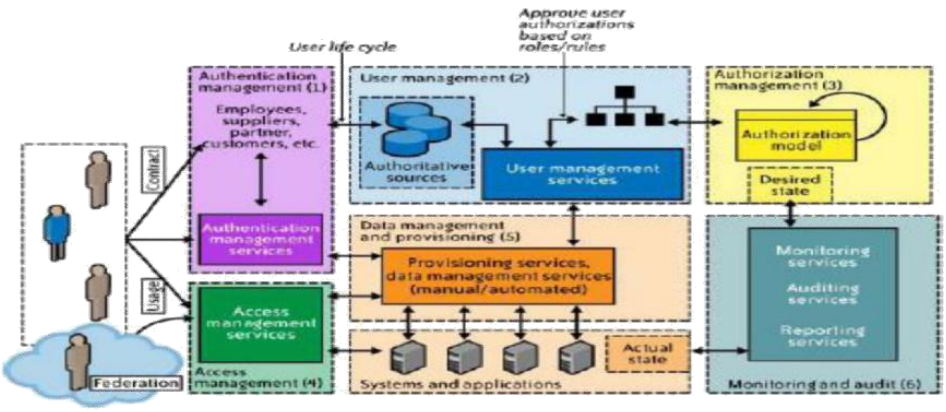
IAM processes support the following operational activities:

Provisioning

This is the process of on-boarding users to systems and applications. These processes provide users with necessary access to data and technology resources. The term typically is used in reference to enterprise-level resource management.

Credential and attribute management

These processes are designed to manage the life cycle of credentials and user attributes— create, issue, manage, revoke—to minimize the business risk associated with identity impersonation and inappropriate account use. Credentials are usually bound to an individual and are verified during the authentication process. The processes include provisioning of attributes, static (e.g., standard text password) and dynamic (e.g., one-time password) credentials that comply with a password standard (e.g., passwords resistant to dictionary attacks), handling password expiration, encryption management of credentials during transit and at rest, and access policies of user attributes



Enterprise IAM functional architecture

Entitlement management

Entitlements are also referred to as *authorization policies*. The processes in this domain address the provisioning and deprovisioning of privileges needed for the user to access resources including systems, applications, and databases.

Compliance management

This process implies that access rights and privileges are monitored and tracked to ensure the security of an enterprise's resources. The process also helps auditors verify compliance to various internal access control policies, and standards that include practices such as segregation of duties, access monitoring, periodic auditing, and reporting.

Identity federation management

Federation is the process of managing the trust relationships established beyond the internal network boundaries or administrative domain boundaries among distinct organizations. A federation is an association of organizations that come together to exchange information about their users and resources to enable collaborations and transactions

Centralization of authentication (authN) and authorization (authZ)

A central authentication and authorization infrastructure alleviates the need for application developers to build custom authentication and authorization features into their applications. Furthermore, it promotes a loose coupling architecture where applications become agnostic to the authentication methods and policies. This approach is also called an ―externalization of authN and authZ‖ from applications.

Identity life cycle

7. Explain user management functions in the cloud

User management functions in the cloud can be categorized as follows:

1. Cloud identity administration
2. Federation or SSO
3. Authorization management
4. Compliance management

Cloud Identity Administration

Cloud identity administrative functions should focus on life cycle management of user identities in the cloud—provisioning, deprovisioning, identity federation, SSO, password or credentials management, profile management, and administrative management. Organizations that are not capable of supporting federation should explore cloud-based identity management services.

By federating identities using either an internal Internet-facing IdP or a cloud identity management service provider, organizations can avoid duplicating identities and attributes and storing them with the CSP. Given the inconsistent and sparse support for identity standards among CSPs, customers may have to devise custom methods to address user management functions in the cloud. Provisioning users when federation is not supported can be complex and laborious.

Federated Identity (SSO)

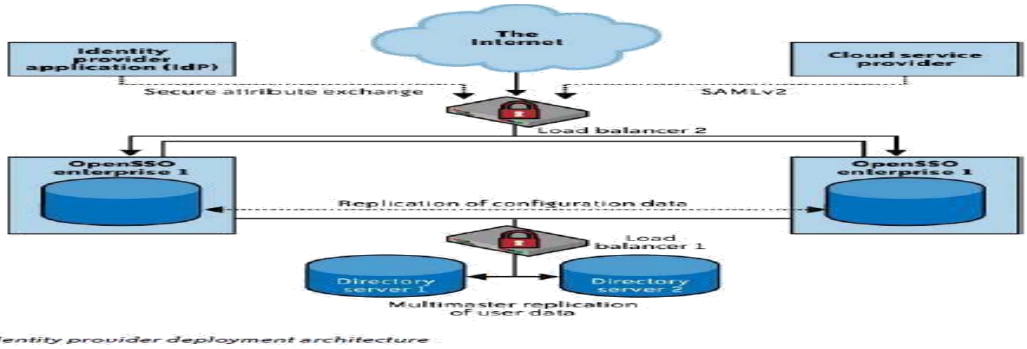
Organizations planning to implement identity federation that enables SSO for users can take one of the following two paths (architectures):

1. Implement an enterprise IdP within an organization perimeter.
2. Integrate with a trusted cloud-based identity management service provider.

Enterprise identity provider

In this architecture, cloud services will delegate authentication to an organization's IdP.

In this delegated authentication architecture, the organization federates identities within a trusted circle of CSP domains. A circle of trust can be created with all the domains that are authorized to delegate authentication to the IdP. In this deployment architecture, where the organization will provide and support an IdP, greater control can be exercised over user identities, attributes, credentials, and policies for authenticating and authorizing users to a cloud service.

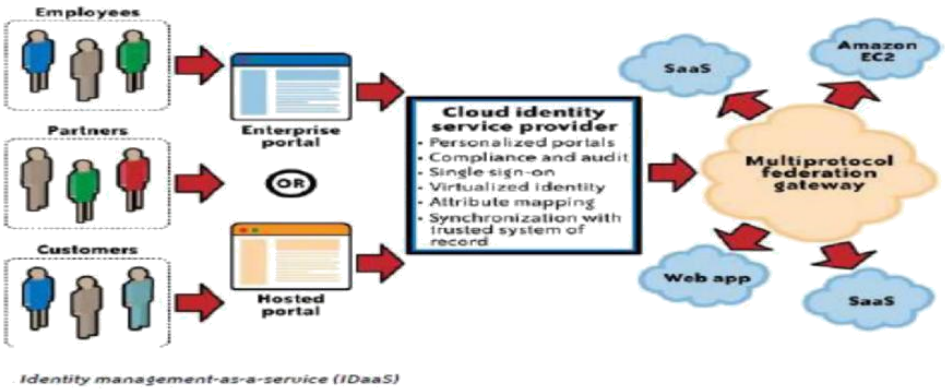


Identity provider deployment architecture

Identity management-as-a-service

In this architecture, cloud services can delegate authentication to an identity management-as-a- service (IDaaS) provider. In this model, organizations outsource the federated identity management technology and user management processes to a third-party service provider. In essence, this is a SaaS model for identity management, where the SaaS IdP stores identities in a ―trusted identity store‖ and acts as a proxy for the organization's users accessing cloud services.

The identity store in the cloud is kept in sync with the corporate directory through a provider proprietary scheme (e.g., agents running on the customer's premises synchronizing a subset of an organization's identity store to the identity store in the cloud using SSL VPNs). Once the IdP is established in the cloud, the organization should work with the CSP to delegate authentication to the cloud identity service provider. The cloud IdP will authenticate the cloud users prior to them accessing any cloud services



Identity management-as-a-service (IDaaS)

Cloud Authorization Management

Most cloud services support at least dual roles (privileges): administrator and end user. It is a normal practice among CSPs to provision the administrator role with administrative privileges. These privileges allow administrators to provision and deprovision identities, basic attribute profiles, and, in some cases, to set access control policies such as password strength and trusted networks from which connections are accepted.

As we mentioned earlier, XACML is the preferred standard for expressing and enforcing authorization and user authentication policies. As of this writing, we are not aware of any cloud services supporting XACML to express authorization policies for users.

IAM Support for Compliance Management

As much as cloud IAM architecture and practices impact the efficiency of internal IT processes, they also play a major role in managing compliance within the enterprise. Properly implemented IAM practices and processes can help improve the effectiveness of the controls identified by compliance frameworks.

IAM practices and processes offer a centralized view of business operations and an automated process that can stop insider threats before they occur. However, given the sparse support for IAM standards such as SAML (federation), SPML (provisioning), and XACML (authorization) by the CSP, you should assess the CSP capabilities on a case-by-case basis and institute processes for managing compliance related to identity (including attribute) and access management.

8. (a) PaaS Availability management

In a typical PaaS service, customers (developers) build and deploy PaaS applications on top of the CSP-supplied PaaS platform. The PaaS platform is typically built on a CSP owned and managed network, servers, operating systems, storage infrastructure, and application components (web services). Given that the customer PaaS applications are assembled with CSP-supplied application components and, in some cases, third-party web services components (mash-up applications), availability management of the PaaS application can be complicated

PaaS applications may rely on other third-party web services components that are not part of the PaaS service offerings; hence, understanding the dependency of your application on third-party services, including services supplied by the PaaS vendor, is essential. PaaS providers may also offer a set of web services, including a message queue service, identity and authentication service, and database service, and your application may depend on the availability of those service components.

App Engine resource is measured against one of two kinds of quotas: a billable quota or a fixed quota.

Billable quotas are resource maximums set by you, the application's administrator, to prevent the cost of the application from exceeding your budget. Every application gets an amount of each billable quota for free. You can increase billable quotas for your application by enabling billing, setting a daily budget, and then allocating the budget to the quotas. You will be charged only for the resources your app actually uses, and only for the amount of resources used above the free quota thresholds.

Fixed quotas are resource maximums set by the App Engine to ensure the integrity of the system. These resources describe the boundaries of the architecture, and all applications are expected to run within the same limits. They ensure that another app that is consuming too many resources will not affect the performance of your app.

Customer Responsibility

The PaaS application customer should carefully analyze the dependencies of the application on the third-party web services (components) and outline a holistic management strategy to manage and monitor all the dependencies.

PaaS platform service levels

Customers should carefully review the terms and conditions of the CSP's SLAs and understand the availability constraints.

Third-party web services provider service levels

When your PaaS application depends on a third-party service, it is critical to understand the SLA of that service.

PaaS Health Monitoring

The following options are available to customers to monitor the health of their service:

- Service health dashboard published by the CSP
- CSP customer mailing list that notifies customers of occurring and recently occurred outages
- RSS feed for RSS readers with availability and outage information
- Internal or third-party-based service monitoring tools that periodically check your PaaS application, as well as third-party web services that monitor your application

8. (b) IaaS Availability management

Availability considerations for the IaaS delivery model should include both a computing and storage (persistent and ephemeral) infrastructure in the cloud. IaaS providers may also offer other services such as account management, a message queue service, an identity and authentication service, a database service, a billing service, and monitoring services. Managing your IaaS virtual infrastructure in the cloud depends on five factors:

✓ Availability of a CSP network, host, storage, and support application infrastructure. This factor depends on the following:

1. CSP data center architecture, including a geographically diverse and fault-tolerance architecture.
2. Reliability, diversity, and redundancy of Internet connectivity used by the customer and the CSP.
3. Reliability and redundancy architecture of the hardware and software components used for delivering compute and storage services.
4. Availability management process and procedures, including business continuity processes established by the CSP.

✓ Availability of your virtual servers and the attached storage (persistent and ephemeral) for compute services

✓ Availability of virtual storage that your users and virtual server depend on for storage service. This includes both synchronous and asynchronous storage access use cases.

Synchronous storage access use cases demand low data access latency and continuous availability, whereas asynchronous use cases are more tolerant to latency and availability.

✓ Availability of your network connectivity to the Internet or virtual network connectivity to IaaS services. In some cases, this can involve virtual private network (VPN) connectivity between your internal private data center and the public IaaS cloud

✓ Availability of network services, including a DNS, routing services, and authentication services required to connect to the IaaS service.

IaaS Health Monitoring

✓ Service health dashboard published by the CSP.

✓ CSP customer mailing list that notifies customers of occurring and recently occurred outages.

✓

Web console or API that publishes the current health status of your virtual servers and network.

9. (a) What Are the Key Privacy Concerns in the Cloud?

These concerns typically mix security and privacy. Here are some additional considerations to be aware of:

Access

Data subjects have a right to know what personal information is held and, in some cases, can make a request to stop processing it. This is especially important with regard to marketing activities; in some jurisdictions, marketing activities are subject to additional regulations and are almost always addressed in the end user privacy policy for applicable organizations. In the cloud, the main concern is the organization's ability to provide the individual with access to all personal information, and to comply with stated requests.

Compliance

What are the privacy compliance requirements in the cloud? What are the applicable laws, regulations, standards, and contractual commitments that govern this information, and who is responsible for maintaining the compliance? How are existing privacy compliance requirements impacted by the move to the cloud? Clouds can cross multiple jurisdictions;

Storage

Where is the data in the cloud stored? Was it transferred to another data center in another country? Is it commingled with information from other organizations that use the same CSP? Privacy laws in various countries place limitations on the ability of organizations to transfer some types of personal information to other countries. When the data is stored in the cloud, such a transfer may occur without the knowledge of the organization, resulting in a potential violation of the local law.

Retention

How long is personal information (that is transferred to the cloud) retained? Which retention policy governs the data? Does the organization own the data, or the CSP? Who enforces the retention policy in the cloud, and how are exceptions to this policy (such as litigation holds) managed?

Destruction

How does the cloud provider destroy PII at the end of the retention period? How do organizations ensure that their PII is destroyed by the CSP at the right point and is not available to other cloud users? How do they know that the CSP didn't retain additional copies? Cloud storage providers usually replicate the data across multiple systems and sites—increased availability is one of the benefits they provide. This benefit turns into a challenge when the organization tries to destroy the data—can you truly destroy information once it is in the cloud? Did the CSP really destroy the data, or just make it inaccessible to the organization? Is the CSP keeping the information longer than necessary so that it can mine the data for its own use?

Audit and monitoring

How can organizations monitor their CSP and provide assurance to relevant stakeholders that privacy requirements are met when their PII is in the cloud?

Privacy breaches

How do you know that a breach has occurred, how do you ensure that the CSP notifies you when a breach occurs, and who is responsible for managing the breach notification process (and costs associated with the process)? If contracts include liability for breaches resulting from negligence of the CSP, how is the contract enforced and how is it determined who is at fault?

9. (b). SaaS Availability Management

SaaS service providers are responsible for business continuity, application, and infrastructure security management processes. This means the tasks your IT organization once handled will now be handled by the CSP. Some mature organizations that are aligned with industry standards, such as ITIL, will be faced with new challenges of governance of SaaS services as they try to map internal service-level categories to a CSP.

Customer Responsibility

Customers should understand the SLA and communication methods (e.g., email, RSS feed, website URL with outage information) to stay informed on service outages. When possible, customers should use automated tools such as Nagios or Siteuptime.com to verify the availability of the SaaS service. As of this writing, customers of a SaaS service have a limited number of options to support availability management. Hence, customers should seek to understand the availability management factors, including the SLA of the service, and clarify with the CSP any gaps in SLA exclusions and service credits when disruptions occur.

SaaS Health Monitoring

The following options are available to customers to stay informed on the health of their service:

- Service health dashboard published by the CSP. Usually SaaS providers, such as Salesforce.com, publish the current state of the service, current outages that may impact

customers, and upcoming scheduled maintenance services on their website

- The Cloud Computing Incidents Database (CCID).

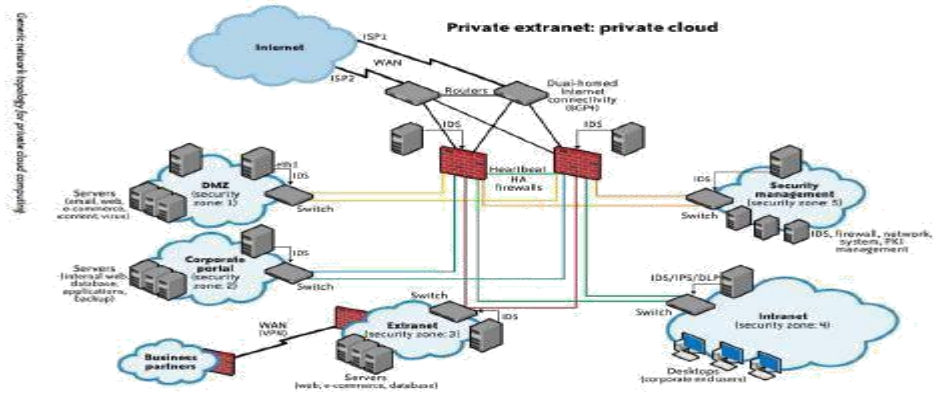
- Customer mailing list that notifies customers of occurring and recently occurred outages.
- Internal or third-party-based service monitoring tools that periodically check SaaS provider

health and alert customers when service becomes unavailable

- RSS feed hosted at the SaaS service provider.

10. Explain cloud infrastructure security at network level

Although your organization's IT architecture may change with the implementation of a private cloud, your current network topology will probably not change significantly. If you have a private extranet in place (e.g., for premium customers or strategic partners), for practical purposes you probably have the network topology for a private cloud in place already. The security considerations you have today apply to a private cloud infrastructure, too. And the security tools you have in place (or should have in place) are also necessary for a private cloud and operate in the same way.



If you choose to use public cloud services, changing security requirements will require changes to your network topology. You must address how your existing network topology interacts with your cloud provider's network topology. There are four significant risk factors in this use case:

- Ensuring the confidentiality and integrity of your organization's data-in-transit to and from your public cloud provider
- Ensuring proper access control (authentication, authorization, and auditing) to whatever resources you are using at your public cloud provider
- Ensuring the availability of the Internet-facing resources in a public cloud that are being used by your organization, or have been assigned to your organization by your public cloud providers
- Replacing the established model of network zones and tiers with domains

Ensuring Data Confidentiality and Integrity

Some resources and data previously confined to a private network are now exposed to the Internet, and to a shared public network belonging to a third-party cloud provider. Although use

of HTTPS (instead of HTTP) would have mitigated the integrity risk, users not using HTTPS (but using HTTP) did face an increased risk that their data could have been altered in transit without their knowledge.

Ensuring Proper Access Control

Since some subset of these resources (or maybe even all of them) is now exposed to the Internet, an organization using a public cloud faces a significant increase in risk to its data. The ability to audit the operations of your cloud provider's network (let alone to conduct any realtime monitoring, such as on your own network), even after the fact, is probably non-existent. You will have decreased access to relevant network-level logs and data, and a limited ability to thoroughly conduct investigations and gather forensic data.

However, the issue of ―non-aged‖ IP addresses and unauthorized network access to resources does not apply only to routable IP addresses (i.e., resources intended to be reachable directly from the Internet). The issue also applies to cloud providers' internal networks for customer use and the assignment of non-routable IP addresses.

Ensuring the Availability of Internet-Facing Resources

There are deliberate attacks as well. Although prefix hijacking due to deliberate attacks is far less common than misconfigurations, it still occurs and can block access to data. According to the same study presented to NANOG, attacks occur fewer than 100 times per month. Although prefix hijackings are not new, that attack figure will certainly rise, and probably significantly, along with a rise in cloud computing.

DNS attacks are another example of problems associated with this third risk factor. In fact, there are several forms of DNS attacks to worry about with regard to cloud computing. Although DNS attacks are not new and are not directly related to the use of cloud computing, the issue with DNS and cloud computing is an increase in an organization's risk at the network level because of increased external DNS querying.

Subject Code: IT6004

Subject Name: SOFTWARE TESTING

Year/Sem: IV/VII

TWO MARK QUESTIONS AND ANSWERS

UNIT 1 INTRODUCTION

1) Define Software Engineering.

Software Engineering is a discipline that produces error free software with in a time and budget.

2) Define software Testing.

Testing can be described as a process used for revealing defects in software, and for establishing that the software has attained a specified degree of quality with respect to selected attributes.

3) List the elements of the engineering disciplines.

1. Basic principles
2. Processes Standards
3. Measurements
4. Tools
5. Methods
6. Best practices Code of ethics
7. Body of knowledge

4) Differentiate between Verification and Validation.

S.No	Verification	Validation
1.	Verification is the process of evaluating software system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.	Verification is usually associated with activities such as inspections and reviews of the software deliverables.
2.	Verification is the process of evaluating software system or component during or at the end of the development phase to satisfy the conditions imposed at the start of that phase.	Verification is usually associated with Traditional execution based testing, i.e., Exercising the code with test cases.

5) Differentiate between testing and debugging. (U.Q Nov/Dec 2008)

S.No	Testing	Debugging
1.	Testing is used to find and locate the defect.	Debugging is used to fixing the defect
2.	Testing is done by testing team	Debugging is done by development team
3.	The main intention behind testing is to find as many defects as possible.	The main intention behind debugging is to remove those defects.

4.	Testing is a dual purpose process reveal defects and to evaluate quality attributes	Debugging or fault localization is the process of Locating the fault or defect repairing the code, and retesting the code.

6) Define process in the context of software quality. (U.Q Nov/Dec 2009)

Process in the software engineering domain, is a set of methods, practices, Standards, documents, activities, polices, and procedures that software engineers use to develop and maintain a software system and its associated artifacts, such as project and test plans, design documents, code, and manuals.

7) List the levels of TMM.

The testing maturity model or TMM contains five levels. They are

1. Level1: Initial
2. Level2: Phase definition
3. Level3: Integration
4. Level4: Management and Measurement
5. Leval5: Optimization /Defect prevention and Quality Control

8) List the members of the critical groups in a testing process (U.Q Nov/Dec 2008)

1. Manager
2. Developer/Tester
3. User/Client

9) Define Error.

An error is mistake or misconception or misunderstanding on the part of a software developer.

10) Define Faults (Defects).

A fault is introduced into the software as the result of an error. It is an anomaly in the software that may cause nit to behave incorrectly, and not according to its specification.

11) Define failures.

A failure is the inability of a software or component to perform its required functions within specified performance requirements.

12) Define Test Cases.

A test case in a practical sense is attest related item which contains the following information.

1. **A set of test inputs:** These are data items received from an external source by the code under test. The external source can be hardware, software, or human.
2. **Execution conditions:** These are conditions required for running the test, for example, a certain state of a database, or a configuration of a hardware device.

3. **Expected outputs:** These are the specified results to be produced by the code under test.

13) Define the term: Test, Test Set, and Test Suite.

1. A Test is a group of related test cases and test procedures.
2. A group of related test is sometimes referred to as a test set.
3. A group of related tests that are associated with a database, and are usually run together, is sometimes referred to as a Test Suite.

14) Define Test Oracle.

Test Oracle is a document, or a piece of software that allows tester to determine whether a test has been passed or failed.

15) Define Test Bed.

A test bed is an environment that contains all the hardware and software needed to test a software component or a software system.

16) Define Software Quality.

a. Quality relates to the degree to which a system, system component, or process meets specified requirements.

b. Quality relates to the degree to which a system, system component, or process meets Customer or user needs, or expectations.

17) List the Quality Attributes.

1. Correctness
2. Reliability
3. Usability Integrity
4. Portability
5. Maintainability
6. Interoperability

18) List the sources of Defects or Origins of defects. Or list the classification of defect (U.Q May/June 2009)

1. Education
2. Communication
3. Oversight
4. Transcription
5. Process

19) Define reviews.

A review is a group meeting whose purpose is to evaluate a software artifact or a set of software artifacts. Review and audit is usually conducted by a SQA group.

20) Programmer A and Programmer B are working on a group of interfacing modules. Programmer A tends to be a poor communicator and does not get along well with Programmer B. Due to this situation, what types of defects are likely to surface in these interfacing modules?

Communication defects.

UNIT 2 TESTING CASE DESIGN

1. Define Smart Tester.

Software must be tested before it is delivered to users. It is responsibility of the testers to design tests that

(i) reveal defects

(ii) can be used to evaluate software performance, usability and reliability.

To achieve these goals, tester must select a finite no. of test cases (i/p, o/p, & conditions).

2. Compare black box and white box testing.

S.No	Black Box	White box testing
1.	Black box testing sometimes called Functional or specification testing.	White box sometimes called clear or glass box testing
2.	In Black box testing , the tester have no Knowledge of its inner structure(i.e. how it works)The tester only has knowledge of what it does(Focus only input & output)	The White box approach focuses on the inner structure of the software to be tested.
3.	Black box approach is usually applied in large size piece of software.	White box approach is usually applied in small size piece of software.

3. Define the term: Random testing and Equivalence class partitioning.

Each software module or system has an input domain from which test input data is selected. If a tester randomly selects inputs from the domain, this is called random testing.

In equivalence class partitioning the input and output is divided in two equal classes or partitions.

4. Define COTS Components.

Test Strategy Knowledge Sources Methods

Black box

1. Requirements document
2. Specifications
3. Domain Knowledge
4. Defect analysis data

1. Equivalence class partitioning (ECP)
2. Boundary value analysis (BVA)
3. State Transition testing.(STT)
4. Cause and Effect Graphing.
5. Error Guessing

White box

High level design

1. Detailed design
2. Control flow graphs
3. Cyclomatic complexity

1. Statement testing
2. Branch testing
3. Path testing
4. Data flow testing Mutation testing
5. Loop testing

5. Define State.

A state is an internal configuration of a system or component. It is defined in terms of the values assumed at a particular time for the variables that characterize the system or component.

6. Define Finite-State machine.

A finite-state machine is an abstract machine that can be represented by a state graph having a finite number of states and a finite number of transitions between states.

7. Define Error Guessing.

The tester/developer is sometimes able to make an educated "guess' as to which type of defects may be present and design test cases to reveal them. Error Guessing is an ad-hoc approach to test design in most cases.

8. Define usage profiles and Certification.

Usage profiles are characterizations of the population of intended uses of the software in its intended environment. Certification refers to third party assurance that a product, process, or service meets a specific set of requirements.

9. Write the application scope of adequacy criteria?

1 Helping testers to select properties of a program to focus on during test.

2. Helping testers to select a test data set for a program based on the selected properties.

3. Supporting testers with the development of quantitative objectives for testing.

4. Indicating to testers whether or not testing can be stopped for that program.

10. What are the factors affecting less than 100% degree of coverage?

1. The nature of the unit
   a. Some statements/branches may not be reachable.
   b. The unit may be simple, and not mission, or safety, critical, and so complete coverage is thought to be unnecessary.
2. The lack of resources
   a. The time set aside for testing is not adequate to achieve complete coverage for all of the units.
   b. There is a lack of tools to support complete coverage.

   Other project related issues such as timing, scheduling. And marketing constraints.

11. What are the basic primes for all structured program.

Sequential (e.g., Assignment statements)
Condition (e.g., if/then/else statements)
Iteration (e.g., while, for loops)

The graphical representation of these three primes are given

Sequence Condition Iteration
False
True False True

12. Define path.

A path is a sequence of control flow nodes usually beginning from the entry node of a graph through to the exit node.

13. Write the formula for cyclomatic complexity?

The complexity value is usually calculated from control flow graph(G) by the formula.

$$V(G) = E-N+2$$

Where E is the number of edges in the control flow graph and N is number of nodes.

14. List the various iterations of Loop testing.

1.Zero iteration of the loop

2.One iteration of the loop

3.Two iterations of the loop

4.K iterations of the loop where k<n

5.n-1 iterations of the loop

6.n+1 iterations of the loop

15. Define test set.

A test set T is said to be mutation adequate for program p provided that for every in equivalent mutant pi of p there is an element t in T such that pi[t] is not equal to p[t].

16. What are the errors uncovered by black box testing?

1. Incorrect or missing functions
2. Interface errors
3. Errors in data structures
4. Performance errors
5. Initialization or termination error

17. List the Axioms/properties described by Weyuker.

1. Applicability property
2. Nonexhaustive Applicabilty property
3. Monotonicity property
4. Inadequate empty set
5. Antiextensionality property
6. General multiple change property
7. Antidecomposotion property
8. Renaming property
9. Complexity property
10. Statement coverage property

18. Write the assumptions of Mutation Testing.

1. The component programmer hypothesis
2. The coupling effect

19. Write the formula for Mutation score.

$$MS(P, T) = \frac{\#\text{of dead mutants}}{\#\text{total mutants} - \#\text{of equivalent mutants}}$$

20. What are the assumptions in which axioms based on?

a. Programs are written in structured programming language
b. Programs are SESE (Single entry/single exit)
c. All input statements appear at the beginning of the program.
d. All output statements appear at the end of the program.

UNIT 3 LEVELS OF TESTING

1. List the levels of Testing or Phases of testing.

    1. Unit Test
    2. Integration Test
    3. System Test
    4. Acceptance Test

2. Define Unit Test and characterized the unit test.

At a unit test a single component is tested. A unit is the smallest possible testable software component.

It can be characterized in several ways

1. A unit in a typical procedure oriented software systems.
2. It performs a single cohensive function.
3. It can be compiled separately.
4. It contains code that can fit on a single page or a screen.

3. List the phases of unit test planning.

Unit test planning having set of development phases.

Phase1: Describe unit test approach and risks.

Phase 2: Identify unit features to be

tested. Phase 3: Add levels of detail to

the plan.

4. List the work of test

planner.

- Identifies test risks.
- Describes techniques to be used for designing the test cases for the units.

  Describe techniques to be used for data validation and recording of test

  results.
- Describe the requirement for test harness and other software that interfaces with the unit to be tested, for ex, any special objects needed for testing object oriented.

5. Define integration Test.

At the integration level several components are tested as a group and the tester investigates component interactions.

6. Define System test.

When integration test are completed a software system has been assembled and its major subsystems have been tested. At this point the developers /testers begin to test it as a whole. System test planning should begin at the requirements phase.

7. Define Alpha and Beta Test.

Alpha test developer's to use the software and note the problems.

Beta test who use it under real world conditions and report the defect to the Developing organization.

8. What are the approaches are used to develop the software?

There are two major approaches to software development

- Bottom-Up
- Top_Down

These approaches are supported by two major types of programming languages. They are

- procedure_oriented
- Object_oriented

9. List the issues of class testing.

- Issue1: Adequately Testing classes
- Issue2: Observation of object states and state changes.
- Issue3: The retesting of classes-I
- Issue4: The retesting of classes-II

10. Define test Harness.

The auxiliary code developed into support testing of units and components is called a test harness. The harness consists of drivers that call the target code and stubs that represent modules it calls.

11. Define Test incident report.

The tester must determine from the test whether the unit has passed or failed the test. If the test is failed, the nature of the problem should be recorded in what is sometimes called a test incident report.

12. Define Summary report.

The causes of the failure should be recorded in the test summary report, which is the summary of testing activities for all the units covered by the unit test plan.

13. Goals of Integration test.

- ✓ To detects defects that occur on the interface of the units.
- ✓ To assemble the individual units into working subsystems and finally a completed system that ready for system test.

14. What are the Integration strategies?

Top_ Down: In this strategy integration of the module begins with testing the upper level modules.

Bottom_ Up: In this strategy integration of the module begins with testing the lowest level modules.

15. What is Cluster?

A cluster consists of classes that are related and they may work together to support a required functionality for the complete system.

16. List the different types of system testing.

- Functional testing
- Performance testing
- Stress testing
- Configuration testing
- Security testing
- Recovery testing

17. Define load generator and Load.

An important tool for implementing system tests is a load generator. A load generator is essential for testing quality requirements such as performance and stress A load is a series of inputs that simulates a group of transactions.

A transaction is a unit of work seen from the system user's view. A transaction consist of a set of operation that may be perform by a person , s/w system or device that is outside the system.

18. Define functional Testing.

Functional tests at the system level are used ensure that the behavior of the system dheres to the requirement specifications.

19. What are the two major requirements in the Performance testing?

Functional Requirement: User describes what functions the software should perform. We test for compliance of the requirement at the system level with the functional based system test.

Quality Requirement: They are nonfunctional in nature but describe quality levels expected for the software.

20. Define stress Testing.

When a system is tested with a load that causes it to allocate its resources in maximum amounts .It is important because it can reveal defects in real-time and other types of systems.

UNIT 4 TEST MANAGEMENT

1) Write the different types of goals?

- ✓ Business goal: To increase market share 10% in the next 2 years in the area of financial software
- ✓ Technical Goal: To reduce defects by 2% per year over the next 3 years.
- ✓ Business/technical Goal: To reduce hotline calls by 5% over the next 2 years
- ✓ Political Goal: To increase the number of women and minorities in high management positions by 15% in the next 3 years.

2) Define Goal and

Policy

A goal can be described as

- ✓ a statement of intent
- ✓ a statement of an accomplishment that an individual or an org wants to achieve.

A Policy can be defined as a high-level statement of principle or course of action that is used to govern a set of activities in an org.

3) Define Plan.

A plan is a document that provides a framework or approach for achieving a set of goals.

3) Define Milestones.

Milestones are tangible events that are expected to occur at a certain time in the Project's lifetime. Managers use them to determine project status.

4) Define a Work Breakdown Structure.(WBS)

A Work Breakdown Structure (WBS) is a hierarchical or treelike representation of all the

tasks that are required to complete a project.

5) List the Test plan components.

1. Test plan identifier
2. Introduction
3. Items to be tested

4. Features to be tested
5. Approach
6. Pass/fail criteria
7. Suspension and resumption criteria
8. Test deliverables
9. Testing Tasks
10. Test environment
11. Responsibilities
12. Staffing and training needs
13. Scheduling
14. Risks and contingencies
15. Testing costs
16. Approvals.

6) Write the approaches to test cost Estimation?

1. The COCOMO model and heuristics
2. Use of test cost drivers
3. Test tasks
4. Tester/developer ratios
5. Expert judgment

7) Write short notes on Cost driver.

A Cost driver can be described as a process or product factor that has an impact on overall project costs. Cost drivers for project the include

1. Product attributes such as the required level of reliability
2. Software quality assurance (V&V) plan
3. Master test plan Review plan: Inspections
4. and walkthroughs
5. Unit test plan Integration
6. test plan
7. System test
8. plan
9. Acceptance
10. test plan

- Hardware attributes such as memory constraints.

- Personnel attributes such as experience level.
- Project attributes such as tools and methods.

8) Write the WBS elements for testing.

1. Project startup
2. Management coordination
3. Tool selection
4. Test planning
5. Test design
6. Test development
7. Test execution
8. Test measurement, and monitoring
9. Test analysis and reporting
10. Test process improvement

9) What is the function of Test Item Transmittal Report or Locating Test Items?

Suppose a tester is ready to run tests on the data described in the test plan. We needs to be able to locate the item and have knowledge of its current status. This is the function of the Test Item Transmittal Report. Each Test Item Transmittal Report has a unique identifier.

10) What is the information present in the Test Item Transmittal Report or Locating Test Items?

1) Version/revision number of the item
2) Location of the item
3) Person responsible for the item (the developer)
4) References to item documentation and test plan it is related to.
5) Status of the item
6) Approvals – space for signatures of staff who approve the transmittal.

11) Define Test incident Report

The tester should record in attest incident report (sometimes called a problem report) any event that occurs during the execution of the tests that is unexpected, unexplainable, and that requires a follow- up investigation.

12) Define Test Log.

The Test log should be prepared by the person executing the tests. It is a diary of the events that take place during the test. It supports the concept of a test as a repeatable experiment.

13) What are the Three critical groups in testing planning and test plan policy?

Managers:

- ✓ Task forces, policies, standards, planning Resource allocation, support for education and training, Interact with users/Clients Developers/Testers

- ✓ Apply Black box and White box methods, test at all levels, Assst with test planning, Participate in task forces.

Users/Clients

- Specify requirement clearly, Support with operational profile,
- Participate in acceptance test planning

14) Define Procedure.

A procedure in general is a sequence of steps required to carry out a specific task.

15) What are the skills needed by a test specialist?

Personal and managerial Skills

- ✓ Organizational, and planning skills, work with others, resolve conflicts, mentor and train others, written /oral communication skills, think creatively.

Technical Skills

- ✓ General software engineering principles and practices, understanding of testing principles and practices, ability to plan, design, and execute test cases, knowledge of networks, database, and operating System.

17)Write the test term hierarchy?

1) Test Manager
2) Test leader
3) Test Engineer
4) Junior Test Engineer.

18)Define Breaking the System.

The goal of stress test is to try to break the system; Find the circumstances under which it will crash. This is sometimes called “breaking the system”.

19)What are the steps for top down integration?

1) Main control module is used as a test driver and stubs are substituted for all components directly subordinate to the main module.

2) Depending on integration approach (Depth or breadth first) subordinate stubs are replaced one at a time with actual components.
3) Tests are conducted as each component is integrated.
4) The completion of each set of tests another stub is replaced with real component
5) Regression testing may be conducted to ensure that new errors have not been introduced.

20)What is meant by regression testing?

Regression testing is used to check for defects propagated to other modules by changes made to existing program. Thus, regression testing is used to reduce the side effects of the changes.

UNIT 5

1. Define Project monitoring or tracking.

Project monitoring refers to the activities and tasks managers engage into periodically check the status of each project .Reports are prepared that compare the actual work done to the work that was planned.

2. Define Project Controlling.

It consists of developing and applying a set of corrective actions to get a project on track when monitoring shows a deviation from what was planned .

3. Define Milestone.

MileStones are tangible events that are expected to occur at a certain time in the projects life time .Mangers use them to determine project status.

4. Define SCM (Software Configuration management).

Software Configuration Management is a set of activities carried out for identifying, organizing and controlling changes throughout the lifecycle of computer software.

5. Define Base line.

Base lines are formally reviewed and agreed upon versions of software artifacts, from which all changes are measured. They serve as the basis for further development and can be changed only through formal change procedures.

6. Differentiate version control and change control.

Version Control combines procedures and tools to manage different versions of configuration objects that are created during software process.

Change control is a set of procedures to evaluate the need of change and apply the changes requested by the user in a controlled manner.

7. What is Testing?

Testing is generally described as a group of procedures carried out to evaluate some aspect of a piece of software. It used for revealing defect in software and to evaluate degree of quality.

8. Define Review.

Review is a group meeting whose purpose is to evaluate a software artifact or a set of software artifacts.

9. What are the goals of Reviewers?

1) Identify problem components or components in the software artifact that need improvement.
2) Identify components of the software artifact that don't need improvement.
3) Identify specific errors or defects in the software artifact.
4) Ensure that the artifact confirms to organizational standards.

10. What are the benefits of a Review program?

1) Higher quality software
2) Increased productivity
3) Increased awareness of quality issues
4) Reduced maintenance costs
5) Higher customer satisfaction

11. What are the various types of Reviews?

- Inspections
- WalkThroughs

12. What is Inspections?

It is a type of review that is formal in nature and requires pre review preparation on the part of the review team. The Inspection leader prepares is the checklist of items that serves as the agenda for the review.

13. What is Walk Throughs?

It is a type of technical review where the producer of the reviewed material serves as the review leader and actually guides the progression of the review .It have traditionally been applied to design and code.

14. List out the members present in the Review Team.

1) SQA(Software Quality Assurance) staff

2) Testers
3) Developers
4) Users /Clients.
5) Specialists.

15. List the components of review plans.

1. Review Goals
2. Items being reviewed
3. Preconditions for the review.
4. Roles, Team size, participants.
5. Training requirements.

6. Review steps.
7. Time

requirements 16.What is

test automation?

A software is developed to test the software. This is termed as test

automation.

17. What are the two types of test cases?

1. Manual
2. Automated

18. What are the disadvantages of first generation automation?

1. Scripts hold hardcoded values.
2. Test maintenance cost is maximized.

19. What are the types of reports?

1. Customized reports.
2. Technical Report
3. Debug reports.

20. What are the stop-test criteria's?

- All the planned tests that were developed have been executed and passed.
- All specified coverage goals have been met.
- The detection of a specific number of defects has been accomplished
- The rates of defect detection for a certain time period have fallen below a specified level.
- Fault seeding ratios are favorable.

IMPORTANT 16 MARKS QUESTIONS

UNIT I

1. Explain about principles of software testing.
   - The testing is the process of exercise a software component using a selected set of test cases, with the intent of revealing defects, evaluating quality.
   - When the test objective is to detect defects, then a good test case is one that has a high probability of revealing yet undetected defects.
   - Test results should be inspected meticulously.
   - A test case must contain the expected output or result.
   - Test cases should be developed for both valid and invalid input conditions.
   - The probability of the existence of additional defects in a software is proportional to the number of defects already detected in the component.
   - Testing should be carried out by a group that is independent of the development group.
   - Tests must be repeatable.
   - Testing should be planned.
   - Testing activities should be integrated into the software life cycle.
   - Testing is a creative and challenging task.

2. Explain: Testing as a process.
   - Requirement analysis and process
   - Procedure specification process.
   - Design process.
   - Testing process.
   - Verification and validation process.

3. Explain the role of process in software quality including components.
   - Reliability
   - Usability
   - Correctness
   - Integrity
   - Portability
   - Maintainability
   - Ability to meet all user requirements

4. Discuss the origin of defects.
   - Defect Sources:
     - ✓ Lack of Education
     - ✓ Poor communication
     - ✓ Oversight
     - ✓ Transcription
     - ✓ Immature process
     - ✓ Errors
     - ✓ Fault
     - ✓ Failures

5. Explain in detail about Tester's role in a software development organization.
    - The testers job is to reveal defects, find weak points, inconsistent behavior, and circumstances where the software does not work as expected.
    - Testers also need to work with designers testers to plan for integration and unit test.
    - Testers are specialist their main function is to plan, execute, record and analyse tests.
6. Discuss in detail about the functions involved in Design defects.
    - Algorithms and processing defects.
    - Control, logic and sequence defects.
    - Data defects.
    - Module interface description defects.
    - Functional description defects.
    - External interface description defects.
7. Explain the process in coding defects.
    - Algorithms and processing defects.
    - Control, logic and sequence defects.
    - Initialization defects.
    - Data flow defects.
    - Data defects.
    - Code documentation defects.
    - External hardware and software interface defects.

8. How does a developer or tester support for developing a defect repository.
    - Test planning and test case development
    - Controlling and monitoring
    - Defect prevention
    - Quality evaluation and control
    - Test measurement
    - Test process improvements

9. Explain the elements of the engineering disciplines.
    - Basic principles
    - Processes.
    - Standards
    - Measurements
    - Tools and methods
    - Best practices.
    - Code of others
    - Body of knowledge.

10. Discuss in detail the internal structure of capability Maturity Level.
    - Initialization level
    - Repeatable level.
    - Defined level
    - Managed level
    - Optimizing level.

UNIT-II

1. Explain the Equivalence class partitioning of Black Box testing with example.
   - Requirements
   - Documents
   - Specification
   - Domain knowledge
   - Defect analysis
   - Data
   - Once the test cases are executed , test results can be used to collect metrics such as
   - Total number of test cases passed and failed
   - Total number of defects in requirements.
   - Number requirements completed and pending.

2. Explain the Boundary value analysis of Black Box testing with example.
   - Boundary value analysis method is useful for arriving at tests that are effective in catching defects that happen at boundaries.
   - There are four possible cases to be tested.
   - All buffers free for use
   - After inserting two buffers and still having free buffers
   - After inserting the last available buffer, no free buffers.
   - No free buffers and new buffer coming in. First buffer needs freeing.

3. Explain Random Testing.
   - Each software module or system has an input domain from which test input data is selected. If a tester randomly selects inputs from the domain, this is called random testing.
   - Some of the issues in random testing are
   - Are the three values adequate to show that the module meets its specification when the tests are run?
   - Should additional or fewer values be used to make the most effective use of resources.
   - Should any values outside the valid domain be used as test input?
   - Are there any input values, other than those selected, more quickly to reveal defects.
4. Discuss the cause and effect graphing of black box testing.
   - Cause and effect graphing technique can be used to combine conditions and derive an effective set of test cases that may disclose inconsistencies in a specification.
   - The specification must be transformed into a graph that resembled a digital logic circuit.
   - The tester should have knowledge of Boolean logic. The graph itself must be expressed in a graphical language.
   - Here causes are placed in left side and effects are placed in right side.
   - Logical relationships are expressed using standard logical operators such as AND, OR, NOT and are associated with arcs.

5. Write a note on Compatibility testing.
   - Horizontal combination
   - Intelligent sampling

   - Backward compatibility testing
   - Forward compatibility testing

6. Explain how to evaluate test adequacy criteria in white box test approach.
   - Applicability property
   - No exhaustive applicability property
   - Monotonicity property.
   - Inadequate empty set.
   - Anti extensionality property
   - General multiple change property.
   - Ant composition property.
   - Renaming property.
   - Complexity property
   - Statement coverage property.

7. Write a note on following Loop Testing
   - Zero iteration of the loop
   - One iteration of the loop
   - Two iteration of the loop
   - K iteration of the loop where k<n
   - N-1 iteration of the loop
   - N+1 iteration of the loop

8. Explain briefly about path and cyclomatic complexity.
   - The cyclomatic complexity attribute is very useful to a tester. The complexity value is useful calculated from the control flow graph (G) by the formula. V(G)=E-N+2
     Where E=number of edges in the control flow graph
     N=number of nodes.
9. Explain in detail about Static testing and structural testing.

   1. Static Testing:
   - Desk checking of the code
   - Code walkthrough
   - Code review
   - Code inspection
   - Formal inspection

   2. Structural testing:
   - Unit/Code functional testing
   - Code coverage testing
   - Code complexity testing

UNIT-III

1. List and explain types of system test.
   - Unit testing
   - Integration Testing
   - System Testing
   - Acceptance Testing
2. Discuss the needs for various levels of testing.
   - Each of the level may consist of one or more sublevels or phases. At each level there are specific testing goals.
   - At unit test a single component is tested. A principal goal is to detect functional and structural defects in the unit.
   - At integration level several components are tested as a group and the tester investigates component interactions.
   - At the system level the system as a whole is tested and principal goal is to evaluate attributes such as usability, reliability and performance.

3. Explain the planning of unit tests.
   - Phase1: Describe unit test approach and risks
   - Phase2: Identify unit features to be tested.
   - Phase3: Add levels of detail to the plan.

4. Explain the design process in unit test.
   - The test cases.
   - Test Procedures.
   - Test Harness
   - Running the unit test and recording results.

5. Explain the Execution process of unit test.
   - Unit test can begin when,
   - The unit becomes available from the developers.
   - The test cases have been designed and reviewed.
   - The test harness and any other supplemental supporting tools are available.
   - The causes of the failure should be recorded in a test summary report.
   - The test summary report is a valuable document for the groups responsible for integration and system tests.
6. Describe the Integration strategies for procedures and functions.
   - **Top down approach**
     - ✓ Depth first search
     - ✓ Breadth first search
   - **Bottom up approach**
   - Bottom up integration of the modules begins with testing the lowest modules, those at the bottom of the structure chart.

7. Define Integration test with its design planning procedures along with its goals.

- Conventional system.
- Object oriented system
- Procedural oriented system.
- System scenarios
- Use case scenarios.

8. Explain types of testing in detail.
   - Functional testing
   - Design / Architecture verification
   - Business vertical testing
   - Deployment testing
   - Beta testing
   - Non Functional testing
   - Performance testing
   - Scalability / load testing
   - Reliability testing
   - Stress testing
   - Interoperability testing
   - Localization testing.
9. Explain the Performance / Load testing.
   - Collecting requirement.
   - Writing test cases.
   - Automating performance test cases.
   - Executing performance test cases.
   - Analyzing performance test cases.
   - Performance tuning
   - Performance benchmarking.

10. Write short notes on Stress testing.

The following guidelines can be used to select the tests for stress testing.

- Repetitive testing.
- Concurrency
- Magnitude
- Random variation.

UNIT-IV

1. Explain the steps in forming a test group.
    - Upper management support for test function
    - Establish test organization, carrier paths.
    - Define education and skill levels.
    - Develop job description
    - Interview candidates
    - Select test group members.

2. Explain in brief about approaches to test cost estimation.
    - COCOMO model
    - Use of test cost driver
    - Test tasks
    - Tester/developer ratios.
    - Expert judgment.

3. Explain elaborately about the basic test plan components as described in IEEE 829-1983.
    - Test plan identifier
    - Introduction
    - Items to be tested
    - Features to be tested
    - Approach
    - Pass/ Fail criteria.
    - Test deliverables.
    - Testing tasks.
    - Test environment
    - Responsibilities.
    - Scheduling
    - Risks.
    - Testing costs
    - Approvals.

4. Write the Work breakdown structure for testing in detail.
    - Project startup.
    - Management coordination
    - Tool selection
    - Test Planning
    - Test design
    - Test development
    - Test execution
    - Test measurement and monitoring
    - Test analysis and reporting.
    - Test process improvement.

5. What is the information present in the test item transmittal report or locating test items? Explain.
    - Version/revision number of the item.
    - Location of the item.
    - Person responsible for the item.
    - Status of the item.
    - Approvals-space for signatures of staff who approve the transmittal.
6. Explain the skills needed by a test specialist?
    - **Personal and managerial skills:**
    - Organizational and planning skills, work with others, resolve conflicts, mentor and train others, written / oral communication skills, think creatively.
    - **Technical Skills:**
    - General software engineering principles and practices, understanding of testing principles and practices, ability to plan, and execute test cases, knowledge of networks, database and operating system
7. Describe the role of three critical groups in test planning and test policy development.
    - **Managers:**
    - Task forces, policies, standards, planning resource allocation, support for education and training, interact with users and clients.
    - **Developers/Testers**:
    - Apply lack box and white box models, test at all levels, asset with test planning, Participate in task forces.
    - **Users/Clients:**
    - Specify requirement clearly, support with operational profile, and participate in acceptance test planning.
8. Explain the test plan components and attachments
    - Introduction
    - Items to be tested
    - Features to be tested
    - Approach
    - Pass/ fail criteria.
    - Suspension and resumption criteria.
    - Scheduling
    - Risks and contingencies.
    - Testing costs
    - Approvals.
9. Explain the activities in test term hierarchy.
    - Test Manager
    - Test leader
    - Test Engineer

- Junior Test engineer.

UNIT-V

1. Explain Skills needed foe test automation.
   - Scripting languages.
   - Programming languages.
   - Generic test requirement for multi products.
   - Design and architecture skills for framework creation.
2. Elaborate scope of automation in testing.
   - Identifying the types of testing amenable to automation.
   - Automating areas less prone to change.
   - Automate test that pertain to standards.
   - Management aspects in automation.

3. Explain the Components of test automation.
   - External modules.
   - Scenario and configuration file modules.
   - Test cases and test framework modules.
   - Tools and results modules.
   - Report generator and reports/ metrics modules.

4. Write short notes on test milestone meetings.
   - Test team
   - Test manager
   - Project manager
   - SQA, process group.
   - Test documents.
   - Test data
   - Miles tone report:
   - Activities
   - Problem
   - Test state.
   - Expenses
   - Plans for next meeting.
   -
5. Explain the steps involved in inspection process.
   - Inspection policies and plans
   - Entry criteria
   - Initiation
   - Preparation
   - Inspection meeting
   - Reporting results.
   - Rework and follow up
   - Exit.
6. Discuss the components of review plan.

- Review goals
- Items being reviewed.
- Preconditions for the review.
- Training requirements
- Checklists and other related documents to be distributed to participants.
- Time requirements.
- Reword and follow up.

7. Explain the benefits of defects analysis and prevention processes.
    - Defects analysis and prevention.
    - Reduces development and maintenance costs
    - Process is more predictable.
    - Improves software quality.
    - Allows focus on serious defects.
    - Support process improvement.
    - Encourage diverse groups to interact.

8. List and explain the members present in the review team.
    - Software quality assurance staff.
    - Testers.
    - Developers.
    - Users/ clients
    - Specialist.