USN | | | | | | | | | | |

Internal Assessment Test 1 – April 2019

Answer Solution

Sub:	Cloud Computing					Sub Code:	18SCS23	Branch:	CSE
Date:	16/04/19	Duration:	90 min's	Max Marks:	50	Sem / Sec:	M. Tech (CSE)/ II SEM		OBE

Answer any FIVE FULL Questions

		MARKS	CO	RBT
1.	a) **Define Cloud Computing? List out the 4 reasons for success and obstacles of cloud computing.**	**[07]**	**CO1**	**L1**

Cloud Computing: Is a model for enabling ubiquitous, on demand, convenient network access to a shared pool of configurable computing resources (network, server storage, applications) that can be rapidly provisioned and released with minimum management effort or service providers interaction.

A non-exhaustive list of reasons for the success of cloud computing includes:

- Cloud computing is in a better position to exploit recent advances in software, networking, torage, and processor technologies. Cloud computing is promoted by large IT companies where these new technological developments take place and these companies have a vested interest to promote the new technologies.
- A cloud consists of a homogeneous set of hardware and software resources in a single administrative domain. In this setup security, resource management, fault-tolerance, and quality of service are less challenging than in a heterogeneous environment with resources in multiple administrative domains.
- Cloud computing is focused on enterprise computing; its adoption by industrial organizations, financial institutions, healthcare organizations and so on, has a potentially huge impact on the economy.
- A cloud provides the illusion of infinite computing resources; its elasticity frees the applications designers from the confinement of a single system.
- A cloud eliminates the need for up-front financial commitment and it is based on a pay-as-you-go approach; this has the potential to attract new

applications and new users for existing applications fomenting a new era of industry-wide technological advancements.

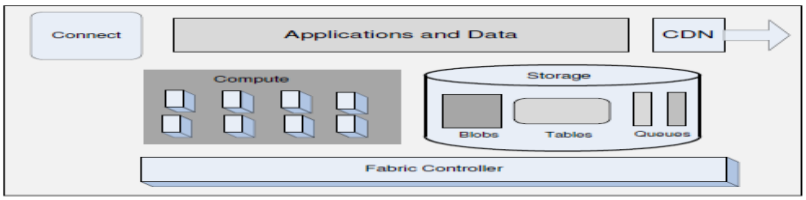
Obstacles in cloud computing:

- **Availability of service**; what happens when the service provider cannot deliver? Can a large company such as GM move its IT to the cloud and have assurances that its activity will not be negatively affected by cloud overload? A partial answer to this question is provided by Service Level Agreements (SLA)s6. A temporary fix with negative economical implications is overprovisioning, i.e., having enough resources to satisfy the largest projected demand.
- **Vendor lock-in**; once a customer is hooked to one provider it is hard to move to another. The standardization efforts at NIST attempt to address this problem.
- **Data confidentiality and auditability**; this is indeed a serious problem.
- **Data transfer bottlenecks**; many applications are data-intensive. A very important strategy is to store the data as close to the site where it is needed as possible. Transferring 1 TB of data on a 1 Mbps network takes 8 000 000 seconds or about 10 days; it is faster and cheaper to use courier service and send data recoded on some media than to send it over the network. Very high speed networks will alleviate this problem in the future, e.g., a 1 Gbps network would reduce this time to 8, 000 seconds, or slightly more than 2 hours.
- **Performance unpredictability**; this is one of the consequences of resource sharing.
- **Elasticity**, the ability to scale up and down quickly. New algorithms for controlling resource allocation and workload placement are necessary. Autonomic computing based on self-organization and self-management seems to be a promising avenue.

b) Explain the cloud vulnerabilities **[03]** CO1 L2

- Clouds are affected by malicious attacks and failures of the infrastructure, e.g., power failures. Such events can affect the Internet domain name servers and prevent access to a cloud or can directly affect the clouds. For example, an attack at Akamai on June 15, 2004 caused a domain name outage and a major blackout that affected Google, Yahoo, and many other sites.
- In May 2009, Google was the target of a serious denial of service attack which took down services like Google News and Gmail for several days. Lightning caused a prolonged down time at Amazon on June 29 − 30, 2012;
- The AWS cloud in the East region of the US which consists of ten data centers across four availability zones, was initially troubled by utility power fluctuations, probably caused by an electrical storm. A June 29, 2012 storm on the East Coast took down some of Virginia based Amazon facilities and affected companies using systems exclusively in this region.
- Instagram, a photo sharing service, was one of the victim of this outage according to http://mashable.com/2012/06/30/aws-instagram/. The recovery from the failure took a very long time and exposed a range of problems.
- For example, one of the ten centers failed to switch to backup generators before exhausting the power that could be supplied by UPS units.
- AWS uses "control planes" to allow users to switch to resources in a different region and this software component also failed.
- The booting process was faulty and extended the time to restart EC2 and EBS services.
- Another critical problem was a bug in the Elastic Load Balancer (ELB), which is used to route traffic to servers with available capacity.
- Clustering the resources in datacenters located in different geographical area is one of the means used today to lower the probability of catastrophic failures.

2. **a)With a neat sketch explain Microsoft windows Azure and its online services** **[05]** CO2 L3



- *Azure* and *Online Services* are PaaS (Platforms-as-a-Service) and, respectively, SaaS (Software as a Service) cloud platforms from Microsoft.
- *Windows Azure* is an operating system, *SQL Azure* is a cloud-based version of the SQL Server, and *Azure AppFabric* (formerly .NET Services) is a collection of services for cloud applications.
- *Windows Azure* has three core components: *Compute* which provides a computation environment, *Storage* for scalable storage, and *Fabric Controller* which deploys, manages, and monitors applications; it interconnects nodes consisting of servers, high-speed connections, and switches.
- The Content Delivery Network (CDN) maintains cache copies of data to speed up computations. The Connect subsystem supports IP connections between the users and their applications running on Windows Azure.
- The API interface to *Windows Azure* is built on REST, HTTP and XML.
- The platform includes five services: *Live Services*, *SQL Azure*, *AppFabric*, *SharePoint* and *Dynamics CRM*.
- A client library and tools are also provided for developing cloud applications in Visual Studio.
- The computations carried out by an application are implemented as one or more *roles*; an application typically runs multiple *instances of a role*. One distinguishes: (i) web role instances used to create web applications; (ii) Worker role instances used to run Window based code; and (iii) VM role instances which run a user-provided Windows Server 2008 R2 image.
- Scaling, load balancing, memory management, and reliability are ensured by a *fabric controller*, a distributed application replicated across a group of machines which owns all of the resources in its environment: computers, switches, load balancing.
- Blobs, tables, queue, and drives are used as scalable storage. A blob contains binary data, a container consists of one or more blobs.

b) What is SLA and its objectives **[05]** CO1 L2

- A Service Level Agreement (SLA) is a negotiated contract between two parties, the customer the service provider; the agreement can be legally binding or informal and specifies the services that the customer receives, rather than how the service provider delivers the services.

Objectives of the agreement are:

- Identify and define the customers' needs and constraints including the level of resources, security, timing, and quality of service.
- Provide a framework for understanding; a critical aspect of this framework is a clear definition of classes of service and the costs.
- Simplify complex issues; for example, clarify the boundaries between the responsibilities of the clients and those of the provider of service in case of failures.
- Reduce areas of conflict.
- Encourage dialog in the event of disputes.
- Eliminate unrealistic expectations.
- An SLA records a common understanding in several areas: (i) services, (ii) priorities, (iii) responsibilities, (iv) guarantees, and (v) warranties.

3. **a) Explain the any one open source platform for private clouds** **[05]** CO2 L2

- Open-source cloud computing platforms such as Eucalyptus, OpenNebula, and Nimbus can be used as a control infrastructure for a private cloud.
- Schematically, a cloud infrastructure carries out the following steps to run an application:
    1. retrieves the user input from the front-end;
    2. retrieves the disk image of a VM (Virtual Machine) from a repository;
    3. locates a system and requests the VMM (Virtual Machine Monitor) running on that system to setup a VM;

4. invokes the DHCP20 and the IP bridging software to set up a MAC and IP address for the VM.

- **Eucalyptus (http://www.eucalyptus.com/)** can be regarded as an open-source counterpart of Amazon's EC2.
- The components of the system are:

  • *Virtual Machine*. Run under several VMMs including Xen, KVM, and Vmware.

  • *Node Controller.* Runs on every server/node designated to host a VM and controls the activities of the node. Reports to a cluster controller.

  • *Cluster Controller*. Controls a number of servers. Interacts with the the node controller on each server to schedule requests on that node. Cluster controllers are managed by cloud controller.

  • *Cloud Controller*. Provides the cloud access to end-users, developers, and administrators. It is accessible through command line tools compatible with EC2 and through aweb-based Dashboard. Manages cloud resources, makes high-level scheduling decisions and interacts with cluster controllers.

  • *Storage Controller*. Provides persistent virtual hard drives to applications. It is the correspondent of EBS. Users can create snapshots from EBS volumes. Snapshots are stored in Walrus and made available across availability zones.

  • *Storage Service (Walrus*). Provides persistent storage and, similarly, to S3 allows users to store objects in buckets.

- The procedure to construct a virtual machine is based on the generic one as below:

  • the euca2ools front-end is used to request a VM;

  • the VM disk image is transferred to a compute node;

  • this disk image is modified for use by the VMM on the compute node;

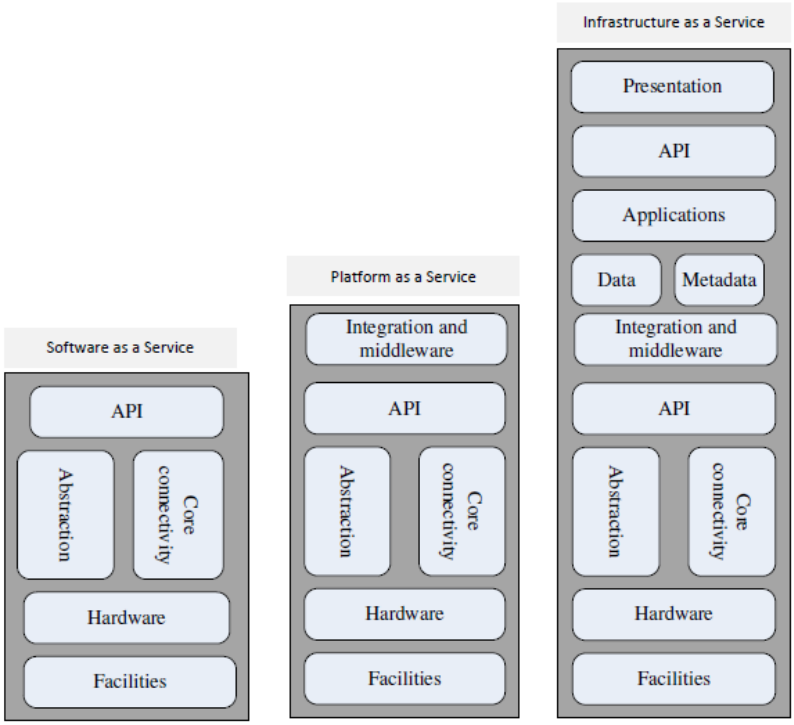
  • the compute node sets up network bridging to provide a virtual NIC with a virtual and MAC address.

- **Open-Nebula (http://www.opennebula.org/)** is a private cloud with users actually logging into the head node to access cloud functions. The system is centralized and its default configuration uses the NFS filesystem.
- The procedure to construct a virtual machine consists of several steps: (i) a user signs in to the head node using ssh; (ii) next, it uses the onevm command to request a VM; (iii) the VM template disk image is transformed to fit the correct size and configuration within the NFS directory on the head node; (iv) the oned daemon on the head node uses ssh to log into a compute node; (v) the compute node sets up network bridging to provide a virtual NIC with a virtual MAC; (vi) the files needed by the VMM are transferred to the compute node via the NFS; (vii)) the VMM on the compute node starts the VM; (viii) the user is able to ssh directly to the VM on the compute node.
- The system is best suited for an operation involving a small to medium size group of trusted and knowledgeable users who are able to configure this versatile system based on their needs.
- **Nimbus (http://www.nimbusproject.org/)** is a cloud solution for scientific applications based on the Globus software.
- The system inherits from Globus the image storage, the credentials for user authentication, and the requirement that a running Nimbus process can ssh into all compute nodes. Customization in this system can only be done by the system administrators.

b) Mention the structure of three delivery models in cloud **[05]** CO2 L2

- The structure of the three delivery models, SaaS, PaaS, and IaaS, according to the Cloud Security Alliance as below shown:
- The structure of the three delivery models, SaaS, PaaS, and IaaS. SaaS gives the users capability to use applications supplied by the service provider but allows no control of the platform or the infrastructure.
- PaaS gives the capability to deploy consumer-created or acquired applications using programming languages and tools supported by the provider.

- IaaS allows the user to deploy and run arbitrary software, which can include operating systems and applications.



- **Software as a Service (SaaS)** - the capability to use applications supplied by the service provider in a cloud infrastructure.
- The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email).
- The user does not manage or control the underlying cloud infrastructure including networks, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.
- Services offered include:

  (a) Enterprise services such as: workflow management, group-ware and collaborative, supply chain, communications, digital signature, customer relationship management (CRM), desktop software, financial management, geo-spatial, and search.

  (b) Web 2.0 applications such as: metadata management, social networking, blogs, wiki services, and portal services.

- The SaaS is not suitable for applications which require real-time response or those where data is not allowed to be hosted externally; the most likely candidates for SaaS are applications when:
    - Many competitors use the same product, such Email;
    - Periodically there is a significant peak in demand, such as billing and payroll;
    - There is a need for the web or mobile access, such as mobile sales management software;
    - There is only a short-term need, such as collaborative software for a project.

b. Platform as a Service (PaaS) gives the capability to deploy consumer-created or acquired applications using programming languages and tools supported by the provider.

- The user does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage. The user has control over the deployed applications and, possibly, application hosting environment configurations.
- Such services include:

    session management, device integration, sandboxes, instrumentation and testing, contents management, knowledge management, and Universal Description, Discovery and Integration (UDDI), a platform-independent, Extensible Markup Language (XML)-based registry providing a mechanism to register and locate web service applications.
- PaaS is not particulary useful when the application must be portable, when proprietary programming languages are used, or when the underlaying hardware and software must be customized to improve the performance of the application.
- Its major application areas are in software development when multiple developers and users collaborate and the deployment and testing services should be automated.

c. Infrastructure as a Service (IaaS) - the capability to provision processing, storage, networks, and other fundamental computing resources; the consumer is able to deploy and run arbitrary software, which can include operating systems and applications.

- The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of some networking components, e.g., host firewalls.
- Services offered by this delivery model include: server hosting,web servers, storage, computing hardware, operating systems, virtual instances, load balancing, Internet access, and bandwidth provisioning.
- The IaaS cloud computing delivery model has a number of characteristics such as: the resources are distributed and support dynamic scaling, it is based on a utility pricing model and variable cost, and the hardware is shared among multiple users.
- This cloud computing model is particulary useful when the demand is volatile and a new business needs computing resources and it does not want to invest in a computing infrastructure or when an organization is expanding rapidly.

4.	**What is workflows? Explain the basic workflow patterns**	**[10]**	**CO2**	**L2**

- Many cloud applications require the completion of multiple interdependent tasks; the description of a complex activity involving such an ensemble of tasks is known as a **workflow**.
- **Workflow pattern** refers to the temporal relationship among the tasks of a process.
- The workflow description languages and the mechanisms to control the enactment of a case must have provisions to support these temporal relationships.
- Workflow patterns are classified in several categories: basic, advanced branching and synchronization, structural, state-based, cancellation, and patterns involving multiple instances.
- The basic workflow patterns illustrated as below:

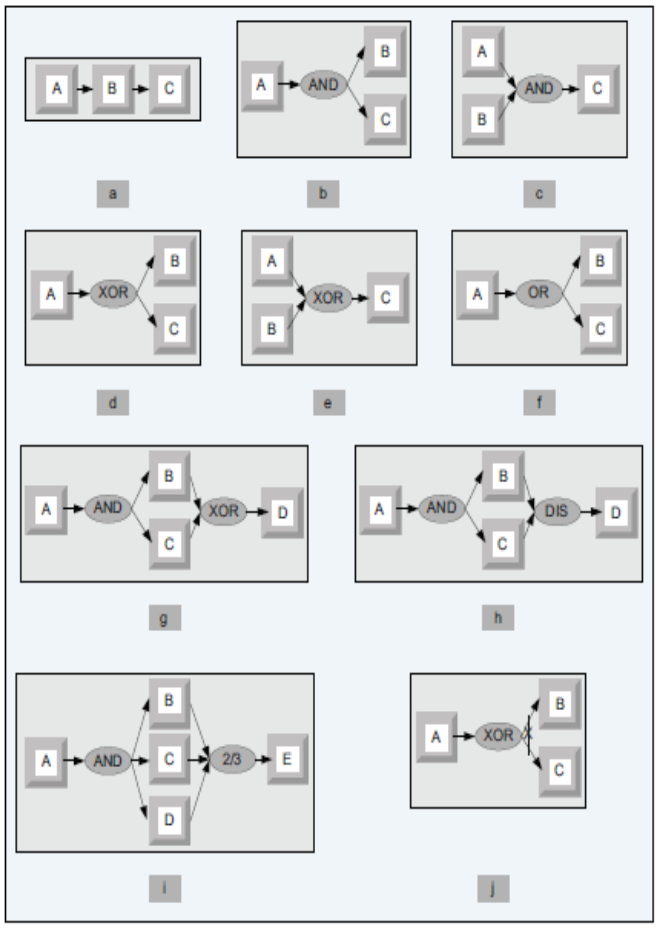


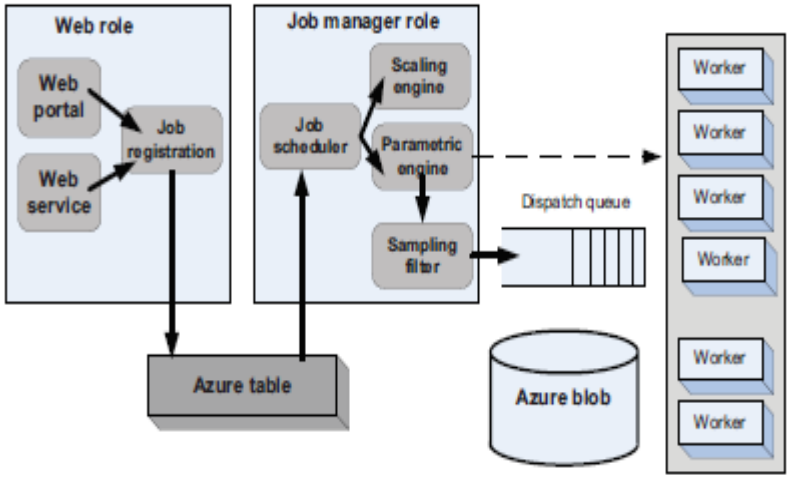
Figure 36: Basic workflow patterns. (a) Sequence; (b) AND split; (c) Synchronization; (d) XOR split; (e) XOR merge; (f) OR split; (g) Multiple Merge; (h) Discriminator; (i) N out of M join; (j) Deferred Choice.

- **The sequence pattern** occurs when several tasks have to be scheduled one after the completion of the other, Figure 36(a).
- The **AND split** pattern requires several tasks to be executed concurrently. Both tasks B and C are activated when task A terminates, Figure 36(b).
- In case of an **explicit AND split** the activity graph has a routing node and all activities connected to the routing node are activated as soon as the flow of control reaches the routing node.
- In the case of **an implicit AND split,** activities are connected directly and conditions can be associated with branches linking an activity with the next ones.

- Only when the conditions associated with a branch are true are the tasks activated.
- The **synchronization pattern** requires several concurrent activities to terminate before an activity can start; in our example, task C can only start after both tasks A and B terminate, Figure 36(c).
- The **XOR split** requires a decision; after the completion of task A, either B or C can be activated, Figure 36(d).
- The **XOR join;** several alternatives are merged into one, in our example task C is enabled when either A or B terminates, Figure 36(e).
- The **OR split** pattern is a construct to choose multiple alternatives out of a set. In our example, after completion of task A, one could activate either B or C, or both, Figure 36(f).
- The **multiple merge** construct allows multiple activations of a task and does not require synchronization after the execution of concurrent tasks. Once A terminates, tasks B and C execute concurrently, Figure 36(g). When the first of them, say B, terminates, then task D is activated; then, when C terminates, D is activated again.
- The **discriminator pattern** waits for a number of incoming branches to complete before activating the subsequent activity, Figure 36(h); then it waits for the remaining branches to finish without taking any action until all of them have terminated. Next, it resets itself.
- The **N out of M join** construct provides a barrier synchronization. Assuming that M > N tasks run concurrently, N of them have to reach the barrier before the next task is enabled; in our example, any two out of the three tasks A, B, and C have to finish before E is enabled, Figure 36(i).
- The **deferred choice** pattern is similar to the XOR split but this time the choice is not made explicitly and the run-time environment decides what branch to take, Figure36(j).

5. Describe the concepts of cloud computing in: a)Biology Research **[05]** CO2 L2

- Molecular dynamics computations are CPU-intensive while protein alignment is data-intensive.
- An experiment carried out by a group from Microsoft Research illustrates protein database using *AzureBLAST*, a version of the BLAST56 program running on the Azure platform.
- *Azure* offers VM with four levels of computing power depending on the number of cores: small (1 core), medium (2 cores), large (8 cores), and extra large (> 8 cores).



Cirrus, a general platform for executing legacy Windows applications

- The job is added to a table called *job registry*.
- The execution of each job is controlled by a *job manager instance* which first scales the size of the worker based on the job configuration, then, the parametric engine starts exploring the parameter space;
- If this is a test-run, the parameter sweeping result is sent to the sampling filter.
- Each task is associated with a record in the task table and this state record is updated periodically by the worker instance running the task; the progress of the task is monitored by the manager.
- The dispatch queue feeds into a set of worker instances.
- A worker periodically updates the task state in the task table and listens for any control signals from the manager.

5. b) Social computing ,Digital content **[05]** CO2 L2

- Social networks play an increasingly important role in people's lives; they have expanded in terms of the size of the population involved and in terms of the function performed.
- Social intelligence is another area where social and cloud computing intersect. Indeed, the process of knowledge discovery and techniques based on pattern recognition demand high performance computing and resources that can be provided by computing clouds.
- Case-based reasoning (CBR), the process of solving new problems based on the solutions of similar past problems, is used by context-aware recommendation systems; it requires similarity-based retrieval.
- In the past, social networks have been constructed for a specific application domain e.g.,MyExperiment and nanoHub for biology and nanoscience, respectively; these networks enable researchers to share data and provide a virtual environment supporting remote execution of workflows.
- Another form of social computing is the volunteer computing when a large population of users donate resources such as CPU cycles and storage space for a specific project; for example, the Mersenne Prime Search initiated in 1996, followed in the late 1990s by the SETI@Home, the Folding@home, and the Storage@Home a project to back up and share huge data sets from scientific research. Information about these projects.
- Such platforms cannot be used for an environment where users require some level of accountability as there are no SLAs.
- The PlanetLab project is a credit based system in which users earn credits by contributing resources and then spend these credits when using other resources.
- The Facebook Markup Language (FBML) is a subset of HTML with proprietary extensions and the Facebook JavaScript (FBJS) is a version of JavaScript parsed when a page is loaded to create a virtual application scope.
- The new technologies supported by cloud computing favor the creation of digital content. Data mashups or composite services combine data extracted by different sources; event driven mashups, also called Svc, interact through events rather than the request-response traditional method.

6. With a neat diagram explain the ZooKeeper coordination service **[10]**

CO2	L2

- Proxy is a single point of failure, A solution for the proxy coordination is problem is ZooKeeper.
- ZooKeeper is a distributed coordination service based on this model. The high throughput and low latency service is used for coordination in large-scale distributed systems.
- The open-source software is written in Java and has bindings for Java and C.
- The ZooKeeper software must first be downloaded and installed on several servers; then clients can connect to any one of these servers and access the coordination service.

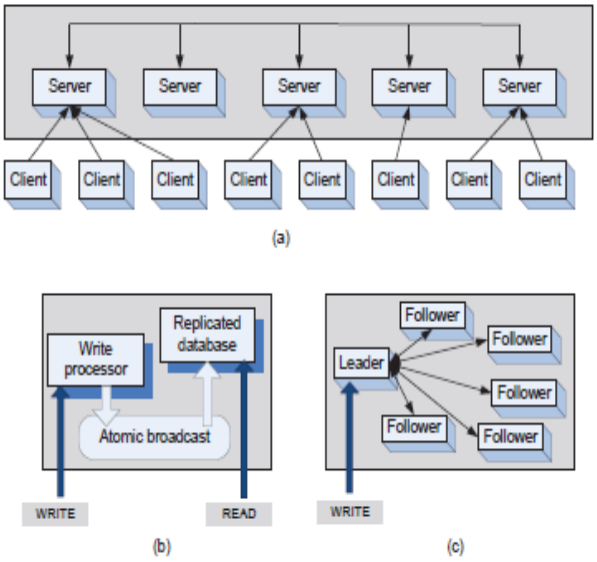


Figure 37: The *ZooKeeper* coordination service. (a) The service provides a single syst image, clients can connect to any server in the pack. (b) Functional model of the *ZooKee* service; the replicated database is accessed directly by *READ* commands, while *WRI* commands involve a more intricate processing based on atomic broadcast. (c) Processin *WRITE* command: (1) a server receiving the command from a client, forwards the comma to the *leader*; (2) the *leader* uses atomic broadcast to reach consensus among all *followe*

- The data stored in each node is read and written atomically, a READ returns all the data stored in a znode, while a WRITE replaces all the data in the znode. Unlike a file system Zookeeper data, the image of the state, is stored in the server memory.

- Updates are logged to disk for recoverability, and WRITEs are serialized to disk before they are applied to the in-memory database which contains the entire tree.
- The ZooKeeper service guarantees:

  1. Atomicity - a transaction either completes or fails.

  2. Sequential consistency of updates - updates are applied strictly in the order they are received.

  3. Single system image for the clients - a client receives the same response regardless of the server it connects to/

  4. Persistence of updates - once applied, an update persists until it is overwritten by a client.

  5. Reliability - the system is guaranteed to function correctly as long as the majority of servers function correctly.
- To reduce the response time READ requests are serviced from the local replica of the server the connected to the client. When the leader receives a WRITE request it determines the state of the system when the WRITE will to be applied and then it transforms the state into a transaction that captures this new state.
- The messaging layer is responsible for the election of a new leader when the current leader fails. The messaging protocols uses: packets - sequence of bytes sent through a FIFO channel, proposals - units of agreement, and messages - sequence of bytes atomically broadcast to all servers. A message is included into a proposal and it is agreed upon before it is delivered. Proposals are agreed upon by exchanging packets with a quorum of servers

  as required by the Paxos algorithm.

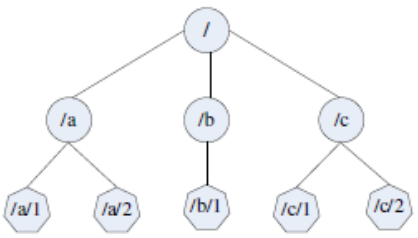


Figure 38: The *ZooKeeper* is organized as a shared hierarchical namespace; a name is a sequence of path elements separated by a backslash.

7. a) Mention in detail the Challenges of cloud computing? **[05]** CO1 L1

- The development of efficient cloud applications inherits the challenges posed by the natural imbalance between computing, I/O, and communication bandwidths of physical systems; these challenges are greatly amplified due to the scale of the system, its distributed nature, and by the fact that virtually all applications are data-intensive.
- Though **cloud computing infrastructures** attempt to automatically distribute and balance the load, the application developer is still left with the responsibility to place the data close to the processing site and to identify optimal storage for the data.
- One of the main advantages of cloud computing, the shared infrastructure, could also have a negative impact.
- **Performance isolation** is nearly impossible to reach in a real system, especially when the system is heavily loaded. The performance of virtual machines fluctuates based on the load, the infrastructure services, the environment including the other users. Security isolation is also challenging on multi-tenant systems.
- **Reliability** is also a major concern; node failures are to be expected whenever a large number of nodes cooperate for the computations. Choosing an optimal instance (in terms of performance isolation, reliability, and security) from those offered by the cloud infrastructure is another critical factor to be considered. Of course, cost considerations also play a role in the choice of the instance type.
- Many applications consist of multiple stages; in turn, each stage may involve multiple instances running in parallel on the systems of the cloud and communicating among them.
- Thus, efficiency, consistency, and communication scalability are major concerns for an application developer.
- Indeed, due to shared networks and unknown topology, cloud infrastructures exhibit inter-node latency and bandwidth fluctuations which affect the application performance.
- **Data storage** plays a critical role in the performance of any data-intensive application; the organization of the storage, the storage location, as well as, the storage bandwidth.
- Applications use metadata associated with individual data records;

- Another important consideration for the application developer is logging. Performance. Considerations limit the amount of data logging, while the ability to identify the source of unexpected results and errors is helped by frequent logging.
- **Logging** is typically done using instance storage preserved only for the lifetime of the instance thus, measures to preserve the logs for a post-mortem analysis must be taken.

b) Explain the architectural styles for cloud applications **[05]** CO2 L2

- Cloud computing is based on the client-server paradigm. The vast majority of cloud applications take advantage of request-response communication between clients and stateless servers.
- A ***stateless server*** does not require a client to first establish a connection to the server, instead it views a client request as an independent transaction and responds to it.
- For example, a basic web server is stateless; it responds to an HTTP request without maintaining a history of past interactions with the client. The client, a browser, is also stateless since it sends requests and waits for responses. The Hypertext Transfer Protocol (HTTP) used by a browser, to communicate with the web server is a request-response application protocol.
- **HTTP** uses the Transport Control Protocol (TCP), a connection oriented and reliable transport protocol; the use of TCP ensures reliable delivery of large objects, but exposes the web servers to denial of service attacks when malicious clients fake attempts to establish a TCP connection and force the server to allocate space for the connection.
- The **Common Object Request Broker Architecture (CORBA**) was developed in the early 1990s to allow networked applications developed in different programming languages and running on systems with different architecture and system software to work with one another.
- At the heart of the system is the Interface Definition Language (IDL) used to specify the interface of an object; the IDL representation is then mapped to the set of programming languages including: C, C++, Java, Smalltalk, Ruby, Lisp, and Python. Networked applications pass CORBA by reference and pass data by value.

- The **Simple Object Access Protocol (SOAP)** is an application protocol developed in 1998 for web applications; its message format is based on the Extensible Markup Language (XML). SOAP uses TCP and more recently UDP transport protocols; it can also be stacked above other application layer protocols such as HTTP, SMTP, or JMS.
- The processing model of SOAP is based on a network consisting of senders, receivers, intermediaries, message originators, ultimate receivers, and message paths. SOAP is an underlying layer of web Services.
- The **Web Services Description Language (WSDL**) was introduced in 2001 as an XML-based grammar to describe communication between end points of a networked application. The abstract definition of the elements involved include: services, collection of endpoints of communication; types, containers for data type definitions; operations, description of actions supported by a service; port types, operations supported by endpoints; bindings, protocols and data format supported by a particular port type; and port, an endpoint as a combination of a binding and a network address.
- These abstractions are mapped to concrete message formats and network protocols to define endpoints, and services.
- **Representational State Transfer (REST)** is a style of software architecture for distributed hypermedia systems. REST supports client communication with stateless servers, it is platform independent, language independent, supports data caching, and can be used in the presence of firewalls.
- REST almost always uses HTTP to support all four CRUD (Create/Read/Update/Delete) operations; it uses GET, PUT, and DELETE to read, write, delete the data, respectively.
- REST is a much easier to use alternative to RPC, CORBA, or Web Services such as SOAP or WSDL.
- The corresponding SOAP version of such a request consists of ten lines or more of XML. The REST server responds with the address of the individual.
- This justifies the statement that REST is a lightweight protocol. As far as usability is concerned, REST is easier to build from scratch and debug, but SOAP is supported by tools that use self-documentation, e.g., WSDL to generate the code to connect.