CS249 Distributed Systems **Siddharth Kumar**
ID No. - 011455470

Homework #1

Q1: Compare and contrast cloud computing with traditional Client-Server Computing? What is novel about cloud computing as a concept?

Answer : With traditional computing :

- Machines have to be maintained and administered.
- Need to purchase and setup hardware infrastructure
- Maintenance is time consuming, compute intensive and expensive
- Data replication and backup has to be done to avoid data loss,

With cloud computing :

- The cloud service providers (CSP) maintains an upgrades the infrastructure.
- No special hardware has to be purchased.
- Maintenance is also taken care of by the CSP.
- No need to explicit data replication as it is already performed by the cloud service.

Moreover, cloud computing requires little maintenance and upgradation and the compute capacity can be scaled up and down in very little time.

Q2: Suppose that the operations of the BLOB object are separated into two categories – public operations that are available to all users and protected operations that are available only to certain named users. State all of the problems involved in ensuring that only the named users can use a protected operation. Suppose that access to a protected operation provides information that should not be revealed to all users, what further problems arise.

Answer : All the access requests to the protected operations have to include the identity of the user making the request. Some of the problem are :

- Defining the identities of the users. These identities can will be used to to define a list of users that can access the protected implementation of the BLOB object.
- Making sure that the identity information comes from a legitimate user and not from someone pretending to be that user.
- Preventing the tampering of the access requests made by the legitimate users.
- Furthermore, the information resulting from protected operations should be hidden from unauthorized users. This can be achieved by encrypting the message.

Q3: Consider a simple server that carries out client requests without accessing other servers. Explain why it is generally not possible to set a limit on the time taken by such a server to respond to a client request. What would need to be done to make the server able to execute requests within a bounded time? Is this a practical option ?

Answer : It is not possible to predict the rate of client requests. If the server executes the requests concurrently by the use of threads, it may not be able to allocate sufficient resources to a particular request in a given time limit.
If the server queues the request and servers them one at at time, there may be starvation for some requests as they can remain in the queue for very long amount of time.

To be able to execute requests in a bounded time, we need to limit the number of clients according to the capacity. To deal with more clients, we can use a server with more compute capacity or replicate the service using more servers.
This solution would be expensive and would require compute resources to create replica of services.

Q4: Define the integrity property of reliable communication and list all the possible threats to integrity from users and from system components. What measures can be taken to ensure the integrity property in the face of each of these sources of threats ?

Answer :
Integrity means that a message received is identical to the one sent and no messages are delivered twice.
Threats to message integrity from users:

- Inserting false messages.
- Replaying old messages.
- Altering messages during transmission.

These threats may be avoided by using access list to authorise users to access protected implementation. Also, encrypting the messages would prevent illegitimate users from reading and altering messages they may intercept.

Threats to integrity from components:

- Message corruption during transmission.
- Message duplication during retransmission.

To prevent threats from components, checksums can be used to detect corrupted messages. Also, we can add sequence numbers to messages to detect duplicate messages.