Homework -3

1. Outline an efficient implementation of globally unique identifiers.

A: It is a 128 bit number used for giving an unique identity for any entities such as Hardware, Software, Connections etc. One way to implement GUID is by using Time Based method where we append the local time to the network address along with a generated pseudo-random number. The possibility that an another machine having the same GUID is next to negligible.

2. Name at least three sources of delay that can be introduced between broadcasting the time over the network and the processors in a distributed system setting their internal clocks.

A:

1. Signal Propogation delay: The signal propogation speed is not constant(as signals slow down when entering atmosphere).
2. Packet Propogation delay: This propogation delay can occur on the LAN.
3. Once the packet arrives, there might be a delay, due to interrupt processing and internal queueing delays.

3. Consider the behavior of two machines in a distributed system. Both have clocks that are supposed to tick 1000 times per millisecond. One of them actually does, but the other ticks only 990 times per millisecond. If UTC updates come in once a minute, what is the maximum clock skew that will occur?

A: The Second clock ticks 990 times per millisecond i.e. 990,000 per second, hence giving an error 10 millisecond per second. For a minute this error becomes 600 milliseconds(10*60=600).

4. One of the modern devices that have (silently) crept into distributed systems are GPS receivers. Give examples of distributed applications that can make use of GPS information.

A: GPS is extensively used in Car navigation systems. It can also be used to construct car alarms to locate stolen cars. Other applications of GPS using distributed systems are aviation, GPS aides for the blind , astronomical telescope pointing, tracking of wild animals, military use, etc.

5. Consider a communication layer in which messages are delivered only in the order that they were sent. Give an example in which even this ordering is unnecessarily restrictive.

A. While downloading a large video from the Internet , the video is divided into consecutive blocks. Each block is identified by its position in the original video. Here, each incoming block is assigned back to its current position by the receiver. Hence, maintaining a order is not necessary.

6. Explain in your own words what the main reason is for actually considering weak consistency models. It is often argued that weak consistency models impose an extra burden for programmers. To what extent is this statement actually true?

A: By using weak consistency models, it turns out that many inconsistencies can be hidden in a relatively cheap way. Also, in strong consistency models in order to keep all copies consistent, it generally requires global synchronization, which is inherently costly in terms of performance. Hence, in order to increase performance ,we have to loosen up the consistency constraints. Hence, weak consistency models are used.
Programmers use lock or transaction to protect their shared data. Basic Idea behind this is that they require loose concurrency than one offered at read and write level operations. However they know synchronization variables stick to sequential consistency

7. What kind of consistency would you use to implement an electronic stock market? Explain your answer.

A: We can implement causal consistency model for electronic stock market. The stock values should be consistent across all the systems and most often it is seen that changes in stock value of different companies are independent of each other i.e. they are not causally related. In causal consistency all the processes agree on the order of causally related operations. Hence, causal consistency can be used for better performance in implementing of electronic stock market.

8. Consider a personal mailbox for a mobile user, implemented as part of a wide-area distributed database. What kind of client-centric consistency would be most appropriate?

A: Monotonic-read consistency would be suitable for distributed email database. Each users mailbox should be distributed and replicated across multiple machines. Suppose that a user reads his mail in California and later flies to Chicago and opens his mailbox again, monotonic-read consistency guarantees that the messages that were in the mailbox in California will also be in thhe mailbox when it is opened in Chicago.

9. We have stated that totally ordered multi-casting using Lamport's logical clocks does not scale. Explain why.

A: In totally ordered multi-casting using Lamport’s Logical clock it is necessary that all the servers are up and running. If any one of the server crashes or is slow, then the performance is reduced. The possibility of performance degradation is more if there are more number of servers. Hence it is seen that Lamport’s logical clocks does not scale.

10. For active replication to work in general, it is necessary that all operations be carried out in the same order at each replica. Is this ordering always necessary?

A: Yes, for active replication it is necessary that all operations need to be carried out in the same order at each replica. Consequently, a totally-ordered multicast mechanism is needed which can be implemented using Lamport's logical clocks.
However, the ordering isn't always necessary if we consider read operations that operate on nonmodified data or commutative write operations.