Solution Assignment 1

- **What do you mean by Horizontal Scalability?**

  Maintaining the throughput (transactions processed per second) or response time taken to process one transaction -- within SLA range --as load grows by adding the additional node(s) in the cluster.

- **Is transaction rollback possible in Cassandra? Explain in brief?**

  In order to provide higher performance and availability, Cassandra doesn't have concept of transaction as implemented in RDBMS. Transaction is transformation from one state to another. In case of Cassandra, each new value is written in the column along with the timestamp as opposed to RDBMS which updates/or rollbacks the new value.

  Cassandra doesn't have concept of transaction controller/coordinator which is used to handle commit/rollback in RDBMS in local and distributed transaction which uses 2PC.

  Only way to roll back the transaction is to repair – in other words update the new value with the old value.

- **Why adding a cache layer with RDBMS is not enough to handle the large amount of data and ensure fast access?**

  - There is limit to how much RAM can be accessed by machine based on the data bus and the type of operating system used. In other words, there is a limit how much data can be made available in the cache.

  - Secondly, if a record, which is cached, is updated, then the cache needs to refresh, otherwise the client will read the old record. This

is not an issue; Hibernate provides second level cache where it is configured to refresh the cache if the record is updated.

4. Drawbacks of Vertical Scalability?

There is a limit to how much a machine can be scaled in various dimensions: such as increasing RAM, adding new hard disk or increasing hard disk size, adding dual/ quad processor, increasing clock speed, or Disk I/O.

- **Which type of consistency is involved in e-commerce sites?**

  Because e-commerce sites involve transaction (insert/delete/update), ideally it should allow strong/immediate read and write consistency. Which means if a record is inserted/updated/deleted -- all the replicas of the record would have the same value and the timestamp. On the same token if the record is read, the record from all the replicas must have same value and timestamp. Any read of the record after the transaction will get the same value (and timestamp) irrespective of the node it is read from.

- **How Cassandra ensures availability if any node is removed?**

  Cassandra architecture is decentralized – there is no master-slave concept as opposed to HBase. It uses peer-to-peer and gossip protocol to have the knowledge of what's going on the other node. If a node is removed, then the other node can take the pending job of the other node to process request.

  In addition, there is replication of records on different nodes based on the replication factor and replication strategy. This also helps in maintaining the availability as a node goes down due to any reason.

- **What is elastic scalability?**

  Elastic scalability relates to increasing the capacity of cluster as load increases; conversely decreasing the cluster capacity as load decreases in order to maintain the-- non-functional requirements-- such as throughput, availability, reliability of the system with in SLA.

  The way it works: in the cluster one node acts as a seed node. This seed node helps in adding new node in cluster as the new nodes gets bootstrap configuration from the seed node. Node can be added or removed based on the load on the cluster. An alarm can configured which fires in help add/remove node based on different factors, which affect runtime performance.