**Session 17: Overview of HBase**

**Assignment 1**

Q 1. Give a brief difference between HBASE and HDFS.

**Answer:**

**HDFS is a distributed file system and has the following properties:**

1. It is optimized for streaming access of large files. You would typically store files that are in the 100s of MB upwards on HDFS and access them through MapReduce to process them in batch mode.  
2. HDFS is optimized for use cases where you write once and read many times like in the case of production logs. You can append to files in some of the recent versions but that is not a feature that is very commonly used. There is no concept of random writes.  
3. HDFS doesn’t do random reads very well.

**HBase** is a distributed column oriented database. The filesystem of choice typically is HDFS owing to the tight integration between HBase and HDFS. HBase provides following:  
1. It gives you the ability to do random read/writes on your data which HDFS doesnt allow you to.  
2. HBase stores data in the form of key value pairs in a columnar fashion. HBase provides a flexible data model.  
3. Fast scans across tables.  
4. Scale in terms of writes as well as total volume of data.

Q 2. List the main components of HBASE.

**Answer:** Main Components of HBASE are:

**Master:** is a master server which is used for monitoring the all region server in cluster. It allocates the regions (table) to the region servers and also handles the load balancing across multiple region server. Doesn't actually store or read data

**Region Server:** is a slave server which is responsible for serving and managing regions. Each Region Server has to serve a set of regions. HBase tables are partitioned into Regions with each table’s rows sorted based on row-ids. One Region can have only one Server

**Region:** Region stores the subset of table data. When a table becomes too big, the table is partitioned into multiple Regions.

**Zookeeper:** Zookeeper is an open-source project that provides services like maintaining configuration information, naming, providing distributed synchronization, etc. Zookeeper has ephemeral nodes representing different region servers. Master servers use these nodes to discover available servers.

**Catalog Tables:** The catalog tables -ROOT- and .META. exist as HBase tables. They are filtered out of the HBase shell's list command, but they are in fact tables just like any other.

Q 3. Does Hbase support sql?

**Answer:** Not really. SQL support for HBase via Hive is in development, however Hive is based on MapReduce which is not generally suitable for low-latency requests.

Q 4. When should we use HBASE, list some of the scenarios for the same.

**Answer:** First, make sure you have enough data. If you have hundreds of millions or billions of rows, then HBase is a good candidate. If you only have a few thousand/million rows, then using a traditional RDBMS might be a better choice due to the fact that all of your data might wind up on a single node (or two) and the rest of the cluster may be sitting idle.

Second, make sure you can live without all the extra features that an RDBMS provides (e.g., typed columns, secondary indexes, transactions, advanced query languages, etc.) An application built against an RDBMS cannot be ported to HBase by simply changing a JDBC driver, for example. Consider moving from an RDBMS to HBase as a complete redesign as opposed to a port.

Third, make sure you have enough hardware. Even HDFS doesnt do well with anything less than 5 DataNodes (due to things such as HDFS block replication which has a default of 3), plus a NameNode.

Q 5. What are the different modes in which Hbase can be run?

**Answer:**

**a). Standalone Mode:** This is the default mode. In standalone mode, HBase does not use HDFS , it uses the local filesystem instead -- and it runs all HBase daemons and a local ZooKeeper all up in the same JVM. Zookeeper binds to a well known port so clients may talk to HBase.

**b) Distributed Mode:**

**Pseudo – Distributed:** A pseudo-distributed mode is simply a distributed mode run on a single host.

**Fully-Distributed Mode:** where the daemons are spread across all nodes in the cluster

Q 6. Why is zookeeper needed in Hbase?

**Answer:** Zookeeper needed in Hbase for region assignment. [ZooKeeper](https://zookeeper.apache.org/) is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. All of these kinds of services are used in some form or another by distributed applications. Each time they are implemented there is a lot of work that goes into fixing the bugs and race conditions that are inevitable. Because of the difficulty of implementing these kinds of services, applications initially usually skimp on them ,which make them brittle in the presence of change and difficult to manage. Even when done correctly, different implementations of these services lead to management complexity when the applications are deployed.

Q 7. Hbase is a schema less database, what does it mean?

**Answer:** Scemaless means that no defined data structure , we can change. In Hbase we can change the structure as many times as we want. We can also put the logically related field together.

Q 8. What is the minimum number of column family every Hbase table should have?

**Answer:** Column Family is used for horizontal scalability. Minimum number of column family in Hbase table can be zero.

Q 10. What is the benefit of using connection pool in Hbase?

**Answer:**