**1.When should we use HBASE, list some of the scenarios for the same in real time.**

**Data volume:** The volume of data is the most common point to be considered. You should have peta bytes of data to be processed in a distributed environment. Otherwise, for a small amount of data, it will be stored and processed in a single node, keeping other nodes idle. So, it will be a misuse of technology framework.

**Application Types:**HBase is not suitable for transactional applications, large volume MapReduce jobs, relational analytics, etc. It is preferred when you have a variable schema with slightly different rows. It is also suitable when you are going for a key dependent access to your stored data.

**Hardware environment:** HBase runs on top of HDFS. And HDFS works efficiently with a large number of nodes (minimum 5). So, if you have good hardware support, then HBase can be a good selection.

**No requirement of relational features:** Your application should not have any requirement for RDBMS features like transaction, triggers, complex query, complex joins etc. If you can build your application without these features, then go for HBase.

**Quick access to data:**If you need a random and real time access to your data, then HBase is a suitable candidate. It is also a perfect fit for storing large tables with multi structured data. It gives ‘flashback’ support to queries, which makes it more suitable for fetching data in a particular instance of time.

Apart from the above points, HBase is also suitable when you need fault tolerant, fast and usable data management in a non-relational environment.

**2. What are the different modes in which Hbase can be run?**

**Solution:**

HBase has two run modes:

1. Standalone mode
2. Distributed mode

**Standalone mode:**

-This is the default mode.

-In standalone mode, HBase does not use HDFS .It uses the local filesystem.

- 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.

**Distributed mode:**

**-**Distributed mode can be subdivided into:

1) Pseudo-Distributed mode

2) Fully-Distributed mode

**Pseudo-Distributed mode:**

-All daemons run on a single node.

-A pseudo-distributed mode is simply a distributed mode run on a single host.

-This is used for prototyping HBASE .

- This should not be used for production and evaluating the HBASE performance.

**Pseudo-Distributed configuration file:**

<configuration>

...

<property>

<name>hbase.rootdir</name>

<value>hdfs://h-24-30.sfo.stumble.net:8020/hbase</value>

</property>

<property>

<name>hbase.cluster.distributed</name>

<value>true</value>

</property>

<property>

<name>hbase.zookeeper.quorum</name>

<value>h-24-30.sfo.stumble.net</value>

</property>

...

</configuration>

**Fully-Distributed mode:**

-Fully-distributed where the daemons are spread across all nodes in the cluster.

-This is the one which is used in the real time and for production of the Hadoop applications.

**Fully-Distributed configuration file:**

<configuration>

...

<property>

<name>hbase.rootdir</name>

<value>hdfs://namenode.example.org:8020/hbase</value>

<description>The directory shared by RegionServers.

</description>

</property>

<property>

<name>hbase.cluster.distributed</name>

<value>true</value>

<description>The mode the cluster will be in. Possible values are

false: standalone and pseudo-distributed setups with managed Zookeeper

true: fully-distributed with unmanaged Zookeeper Quorum (see hbase-env.sh)

</description>

</property>

...

</configuration>

**3. Need and working of zookeeper in Hbase?**

### Distributed HBase will depend on Zookeeper for the purpose of cluster configuration as well management.

### Apache HBase, ZooKeeper coordinates, communicates, and shares state will be between Master and Region Server.

### HBase will design policy using ZooKeeper only for transient data that is, for coordination and state communication.

### If the HBase’s ZooKeeper data is removed, only the transient operations will get affected

### data can be written and read to/from the HBase.

* Distributed Apache HBase (TM) installation will depend upon running the ZooKeeper cluster.
* Participating nodes and clients need to able to access the running ZooKeeper ensemble.
* Apache HBase by default will manage ZooKeeper "cluster".
* It will start and stop ZooKeeper ensemble as part of HBase start/stop process.
* This variable,by defaults be true,and will tell the HBase whether to start/stop the ZooKeeper ensemble servers as part of HBase start/stop.