HBase:

* non-relational column-oriented distributed database
* it is a NoSQL open source database that stores data in rows and columns
* Two in-memory data structure [ memstore, blockcache ]
* Two disk data structure [ WAL, HFile ]
* Writes data to WAL on disk and also to memstore in memory
* If memstore reached threshold level it will flush into the HFiles on disk
* Data is read from HFile block into block storage in memory
* If it’s required it will sync the latest data from memstore.

Tuning:

* **Row keys** always salt your keys by adding random numbers or characters to the row key prefix
* **ColumnFamily** name as small as possible, preferably one character.
* **Row key** as short as possible
* Use **batch put’s** : construct list of puts then call HTableInterface.put()
* While use batching don’t forget to increase no of connections to HBase
* **Bucket Caching** to improve your read performance, this setting enabled on newer HBase cluster
* Avoid **major compaction**, it will read all the store file from region and write into the single store file.
* Avoid HBase cluster for other Hadoop Applications
* Improve read performance by increasing block cache : in settings file
  + hfile.block.cache.size = 0.6
  + hbase.regionserver.global.memstore.size = 0.2

Hive:

* it is an open source data warehouse
* its data warehouse framework for querying and analysis of datasets is stored in HDFS
* To handle massive volumes of data
* To bring familiar db concepts (SQL) to the hadoop world (HQL)
* Not for real time query, only for analysis (tool to query data) over a period of time

|  |  |
| --- | --- |
| Hbase | Hive |
| Support NoSQL DB | Is not DB |
| Support real time data streaming That is OLAP (Online Transaction Processing system) | Support batch processing. That is OLAP (Online Analytical Processing system) |
| Schema free | Schema model |
| DataBase model is a relational DBMS | DataBase model is wide column store |
| uses HQL(Hive query language) | does not use SQL |
| uses sharding method for partition | uses sharding method for partition |
| While we do not want to write complex MapReduce code, we use Apache Hive. | While we want to have random access to read and write a large amount of data, we use HBase. |