**Hive vs.HBase–Different Technologies**

HBase and Hive are two hadoop based big data technologies that serve different purposes. For instance, when you login to Facebook, you see multiple things like your friend list, you news feed, friend suggestions, people who liked your statuses, etc. With 1.79 billion monthly active users on Facebook and the profile page loading at lightning fast speed, can you think of a single big data technology like Hadoop or Hive or HBase doing all this at the backend? All these technologies work together to render an awesome experience for all Facebook users. The complexity of big data systems requires that every technology needs to be used in conjunction with the other.

Let’s consider the friend recommendations feature on Facebook, it is something that does not change every second or minute. Thus, recommendations can be pre-computed for all Facebook users. However, high throughput is required to pre-compute friend recommendations but latency is just fine. This is when Hadoop MapReduce or HIVE is helpful. Your Facebook profile data or news feed is something that keeps changing and there is need for a NoSQL database faster than the traditional RDBMS’s. HBase plays a critical role of that database. In this case, the analytical use case can be accomplished using apache hive and results of analytics need to be stored in HBase for random access.

Hive and HBase are both data stores for storing unstructured data. HBase is a NoSQL database used for real-time data streaming whereas Hive is not ideally a database but a mapreduce based SQL engine that runs on top of hadoop. Ideally comparing Hive vs. HBase might not be right because HBase is a database and Hive is a SQL engine for batch processing of big data. Instead of understanding Hive vs. HBase- what is the difference between Hive and HBase, let’s try to understand what hive and HBase do and when and how to use Hive and HBase together to build fault tolerant big data applications.

## ****Apache Hive****

Hive is a SQL engine on top of hadoop designed for SQL savvy people to run mapreduce jobs through SQL like queries. Hive allows developers to impose a logical relational schema on various file formats and physical storage mechanisms within or outside the hadoop cluster. SQL like queries are run against those schemas as Hadoop MapReduce jobs. With limited write capabilities and interactivity, Hive is meant for the execution of batch transformations and large analytical queries.

### ****When to use Hive****

RDBMS professionals love apache hive as they can simply map HDFS files to Hive tables and query the data. Even the HBase tables can be mapped and Hive can be used to operate on that data. Apache Hive should be used for data warehousing requirements and when the programmers do not want to write complex mapreduce code. However, all problems can be solved using apache hive. For big data applications that require complex and fine grained processing, Hadoop MapReduce is the best choice.

### ****Companies Using Apache Hive – Hive Use Cases****

Apache Hive has approximately 0.3% of the market share i.e. 1902 companies are already using Apache Hive in production.

* Scribd uses Hive for ad-hoc querying, data mining and for user facing analytics.
* Hive is an integral part of the Hadoop pipeline at Hubspot for near real-time web analytics.
* Chitika, the popular online advertising network uses Hive for data mining and analysis of its 435 million global user base.

## ****HBase – The NoSQL Hadoop Database****

Apache Hadoop does not provide random access capabilities and this is when the Hadoop database HBase comes to the rescue. HBase is high scalable (scales horizontally using off the shelf region servers), highly available, consistent and low latency NoSQL database.  With flexible data models, cost effectiveness and no Sharding (automatic Sharding), HBase works well with sparse data. Before choosing HBase for your applications, do ask these questions –

* Do you have sufficient hardware?
* Does your applications require those additional features that RDBMS does not provide?
* Do you have enough data?

### ****When to use HBase****

Apache Hadoop is not a perfect big data framework for real-time analytics and this is when HBase can be used i.e.  For real-time querying of data. HBase is an ideal big data solution if the application requires random read or random write operations or both. If the application requires to access some data in real-time then it can be stored in a NoSQL database. HBase has its own set of wonderful API’s that can be used to pull or push data. HBase can also be integrated perfectly with Hadoop MapReduce for bulk operations like analytics, indexing, etc. The best way to use HBase is to make Hadoop the repository for static data and HBase the data store for data that is going to change in real-time after some processing.

HBase should be used when –

* There is large amount of data.
* ACID properties are not mandatory but just required.
* Data model schema is sparse.
* When your applications needs to scale gracefully.

### ****Companies Using HBase - HBase Use Cases****

In the big data category, HBase has a market share of about 9/1% i.e. approximately 6190 companies use HBase. Companies use HBase for time series analysis or for click stream data storage and analysis.

* Original HBase use case was at Google which wanted to store massive databases for the internet and its users.
* Facebook uses HBase for real-time analytics, counting Facebook likes and for messaging.
* FINRA Financial Industry Regulatory Authority uses HBase to store all the trading graphs.
* Pinterest uses HBase to store the graph data.
* Flipboard uses HBase to personalize the content feed for its users.

## ****Hive vs. HBase - Difference between Hive and HBase****

* Hive is query engine that whereas HBase is a data storage particularly for unstructured data.
* Apache Hive is mainly used for batch processing i.e. OLAP but HBase is extensively used for transactional processing wherein the response time of the query is not highly interactive i.e. OLTP.
* Unlike Hive, operations in HBase are run in real-time on the database instead of transforming into mapreduce jobs.
* HBase is to real-time querying and Hive is to analytical queries.

## ****Hive and HBase –Better Together****

Hive has some limitations of high latency and HBase does not have analytical capabilities, integrating the two technologies together is the best solution. Often, people working with big data have this question in mind on –“How to use HBase from Hive? How well does using hive and HBase together work and what is the best way to use them?

Commonly HBase and Hive are used together on the same Hadoop cluster. Hive can be used as an ETL tool for batch inserts into HBase or to execute queries that join data present in HBase tables with the data present in HDFS files or in external data stores.

It is possible to write HiveQL queries over HBase tables so that HBase can make the best use of Hive’s grammar and parser, query execution engine, query planner, etc. Apache Hive has an additional library for interacting with HBase where the middle layer between Hive and HBase is implemented. When accessing HBase from Hive queries, there is a primary interface called HBaseStorageHandler that needs to be implemented. The application can also interact with HBase tables directly through input and output format but the handler is easy to implement and works well with most of the use cases. The interface between Hive and HBase is still in its maturing phase but has a great potential. The only issue integrating hive with HBase is the impedance mismatch between HBase’s sparse and untyped schema over Hive’s dense and typed schema.