HBase

HBase runs on top of HDFS and provides BigTable like capabilities to Hadoop.

HBase	HDFS
Distributed systems that scale	Distributed systems that scale
Good for batch processing	Good for batch processing
Fast Record lookup	Not good for record lookup
Support for record level insertions	Not good for incremental addition of batches
Support for updates by creating new versions of values	Not good for updates

HBase has its own data storage on top of HDFS, and can store unstructured data.

HDFS	Hive	HBase
Unstructured data	Structured data	Unstructured data
Writes : No updates, only appends	No support for writes	Arbitrary writes
Reads entire file and analyses	Analytics via SQL	Analytics

The fundamental features of HBase are

1. Columnar storage - Each column is a separate file, one value per line
2. Unstructured data formats

Row storage is when data is stored as a single file with one row per line.

HBase Data Model

The HBase model uses

1. key,value pairs, with the value usually being a JSON object and the row key being the so-called primary key of the database.
    a. Indexed for fast lookup
2. Data is stored as byte arrays.
3. The table is sparse, as different rows may have different sets of attributes.
4. Different types of data are separated into different column families.
    a. Each column family has one or more columns
5. A single cell might have different values at different time stamps
    a. A version number can be user specified and is unique within each key

HBase Architecture

1. Master
2. Region Server
3. Client

A region is a subset of a table's rows, by horizontal range partitioning. This is automatically done by the HBase infrastructure.

The Region Server manages data regions, and serves data for reads and writes using logging.

The master is responsible for coordinating the slaves, and assigns regions and detects failures. It also coordinates load balancing and monitoring using zookeeper.

Region servers and the active HMaster connect with a session to the Zookeeper, which maintains ephemeral nodes for active sessions via heartbeats.

Each region server creates an ephemeral node, and the HMaster monitors these nodes to discover the available region servers and monitors these for server failures. Ephemeral nodes are used so that failure of the nodes can be offloaded to standby nodes.

A META table is stored in the region servers and stores the locations of the regions. It is maintained by the master and queried by the client.

This META table is stored in Zookeeper.

The automatic creation of regions is done by

1. Initially, each table is a single region
2. Master monitors load on region servers
3. If region is too loaded, region is split
4. Regions can also be migrated if the load on the server is too high.

Read in HBase

First time read/writes are done as

1. Get region server that stores META table
2. Query META server to get region server corresponding to key
3. Cache this region server
4. Get row from corresponding region server

Region Server Components

1. WAL - Write-Ahead-Log
    a. File on HDFS
    b. Used for recovery
    c. Shows new data which is not persisted
2. Block Cache
    a. Read cache
    b. Stores frequently accessed data
3. MemStore
    a. Write cache
    b. Stores data not yet written to disk
4. Hfiles
    a. Stores key-value pairs on disk as sorted rows

HBase Usage

1. Creation
    a. Create <table name>, <col-family name>
2. Adding entries
    a. Put <table name>, <row key>, <column name>, <value>
3. Retrieval
    a. Get <table name>, <row key>
    b. Scan <table name> for all rows

HBase is usually used when the application needs random reads and writes and the access patterns are known.