

HBase



BY
PROF. PRAKASH PATEL
ASST.PROF.-IT DEPT.
GIT

What is HBase?



- HBase is a distributed column-oriented database built on top of the Hadoop file system.
- It is an open-source project and is horizontally scalable.
- HBase is a data model that is similar to Google's big table designed to provide quick random access to huge amounts of structured data.
- It is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System.

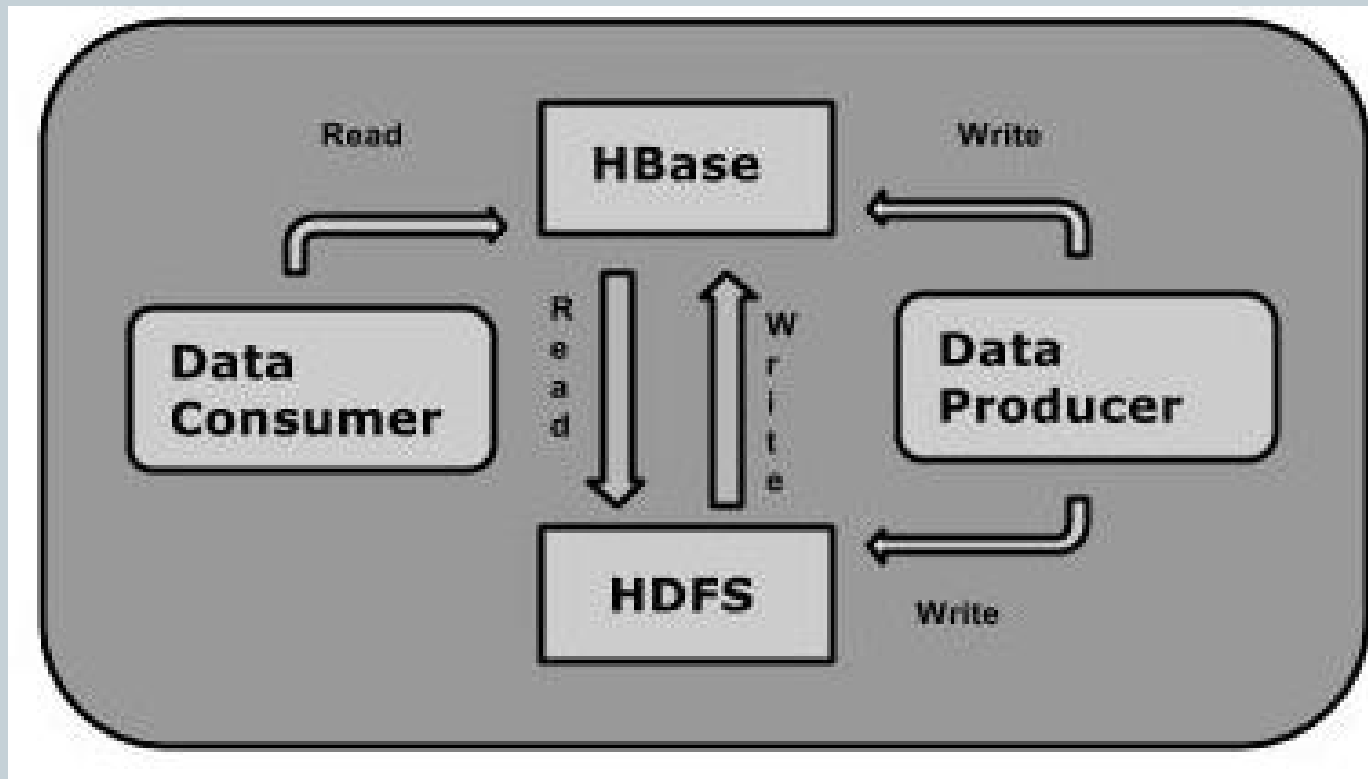


- One can store the data in HDFS either directly or through HBase.
- Data consumer reads/accesses the data in HDFS randomly using HBase.
- HBase sits on top of the Hadoop File System and provides read and write access.

Architecture of HBase



- The following component diagram depicts the architecture of HBase:



HBase and HDFS



HDFS	HBase
HDFS is a distributed file system suitable for storing large files.	HBase is a database built on top of the HDFS.
HDFS does not support fast individual record lookups.	HBase provides fast lookups for larger tables.
It provides high latency batch processing; no concept of batch processing.	It provides low latency access to single rows from billions of records (Random access).
It provides only sequential access of data.	HBase internally uses Hash tables and provides random access, and it stores the data in indexed HDFS files for faster lookups.

Storage Mechanism in HBase



- HBase is a **column-oriented database** and the tables in it are sorted by row.
- The table schema defines only column families, which are the key value pairs.
- Table is a collection of rows.
- Row is a collection of column families.
- Column family is a collection of columns.
- Column is a collection of key value pairs.

[illegible]

continue



5	Send Plan The compiler checks the requirement and resends the plan to the driver. Up to here, the parsing and compiling of a query is complete.
6	Execute Plan The driver sends the execute plan to the execution engine.
7	Execute Job Internally, the process of execution job is a MapReduce job. The execution engine sends the job to JobTracker, which is in Name node and it assigns this job to TaskTracker, which is in Data node. Here, the query executes MapReduce job.
7.1	Metadata Ops Meanwhile in execution, the execution engine can execute metadata operations with Metastore.
8	Fetch Result The execution engine receives the results from Data nodes.
9	Send Results The execution engine sends those resultant values to the driver.
10	Send Results The driver sends the results to Hive Interfaces.

HIVE Data Types



- All the data types in Hive are classified into four types, given as follows:
- 1. Column Types
- 2. Literals
- 3. Null Values
- 4. Complex Types

MapReduce I/O



- Input and Output types of a **MapReduce job**:
(Input) $\langle k1, v1 \rangle \rightarrow \text{map} \rightarrow \langle k2, v2 \rangle \rightarrow$
 $\text{reduce} \rightarrow \langle k3, v3 \rangle$ (Output).

	Input	Output
Map	$\langle k1, v1 \rangle$	list ($\langle k2, v2 \rangle$)
Reduce	$\langle k2, \text{list}(v2) \rangle$	list ($\langle k3, v3 \rangle$)

Column Versus Row



Column Oriented and Row Oriented

Column-oriented databases are those that store data tables as sections of columns of data, rather than as rows of data. Shortly, they will have column families.

Row-Oriented Database	Column-Oriented Database
It is suitable for Online Transaction Process (OLTP).	It is suitable for Online Analytical Processing (OLAP).
Such databases are designed for small number of rows and columns.	Column-oriented databases are designed for huge tables.

Column families



COLUMN FAMILIES

Row key	personal data		professional data	
empid	name	city	designation	salary
1	raju	hyderabad	manager	50,000
2	ravi	chennai	sr.engineer	30,000
3	rajesh	delhi	jr.engineer	25,000

HBase and RDBMS



HBase	RDBMS
HBase is schema-less, it doesn't have the concept of fixed columns schema; defines only column families.	An RDBMS is governed by its schema, which describes the whole structure of tables.
It is built for wide tables. HBase is horizontally scalable.	It is thin and built for small tables. Hard to scale.
No transactions are there in HBase.	RDBMS is transactional.
It has de-normalized data.	It will have normalized data.
It is good for semi-structured as well as structured data.	It is good for structured data.

Features of HBase



- HBase is linearly scalable.
- It has automatic failure support.
- It provides consistent read and writes.
- It integrates with Hadoop, both as a source and a destination.
- It has easy java API for client.
- It provides data replication across clusters.

Where to Use HBase



- Apache HBase is used to have random, real-time read/write access to Big Data.
- It hosts very large tables on top of clusters of commodity hardware.
- Apache HBase is a non-relational database modeled after Google's Bigtable. Bigtable acts up on Google File System, likewise Apache HBase works on top of Hadoop and HDFS.

Applications of HBase



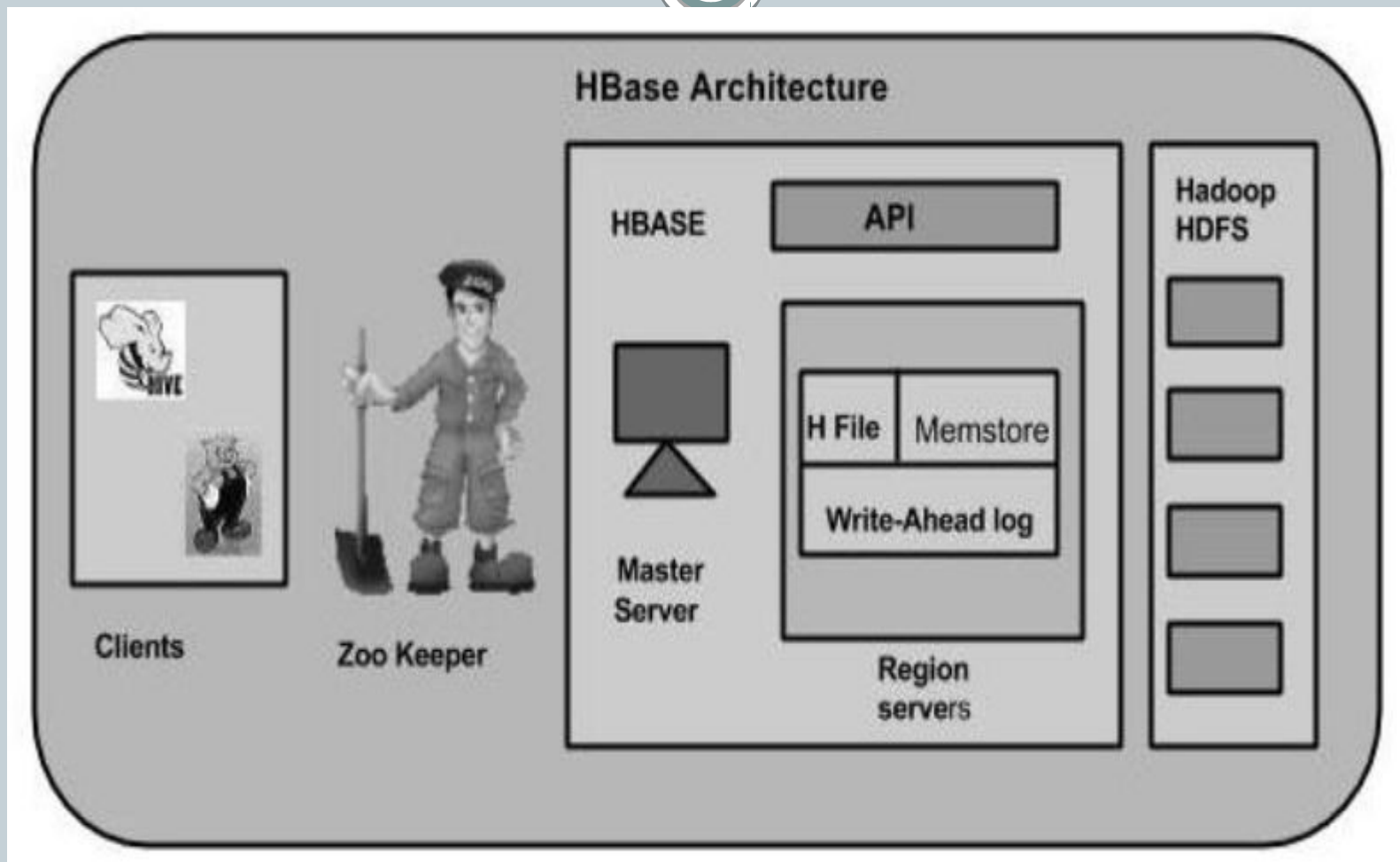
- It is used whenever there is a need to write heavy applications.
- HBase is used whenever we need to provide fast random access to available data.
- Companies such as Facebook, Twitter, Yahoo, and Adobe use HBase internally.

HBase Architecture



- In HBase, tables are split into regions and are served by the region servers.
- Regions are vertically divided by column families into “Stores”.
- Stores are saved as files in HDFS.
- **Note:** The term ‘store’ is used for regions to explain the storage structure.

HBase Architecture



Hbase Components

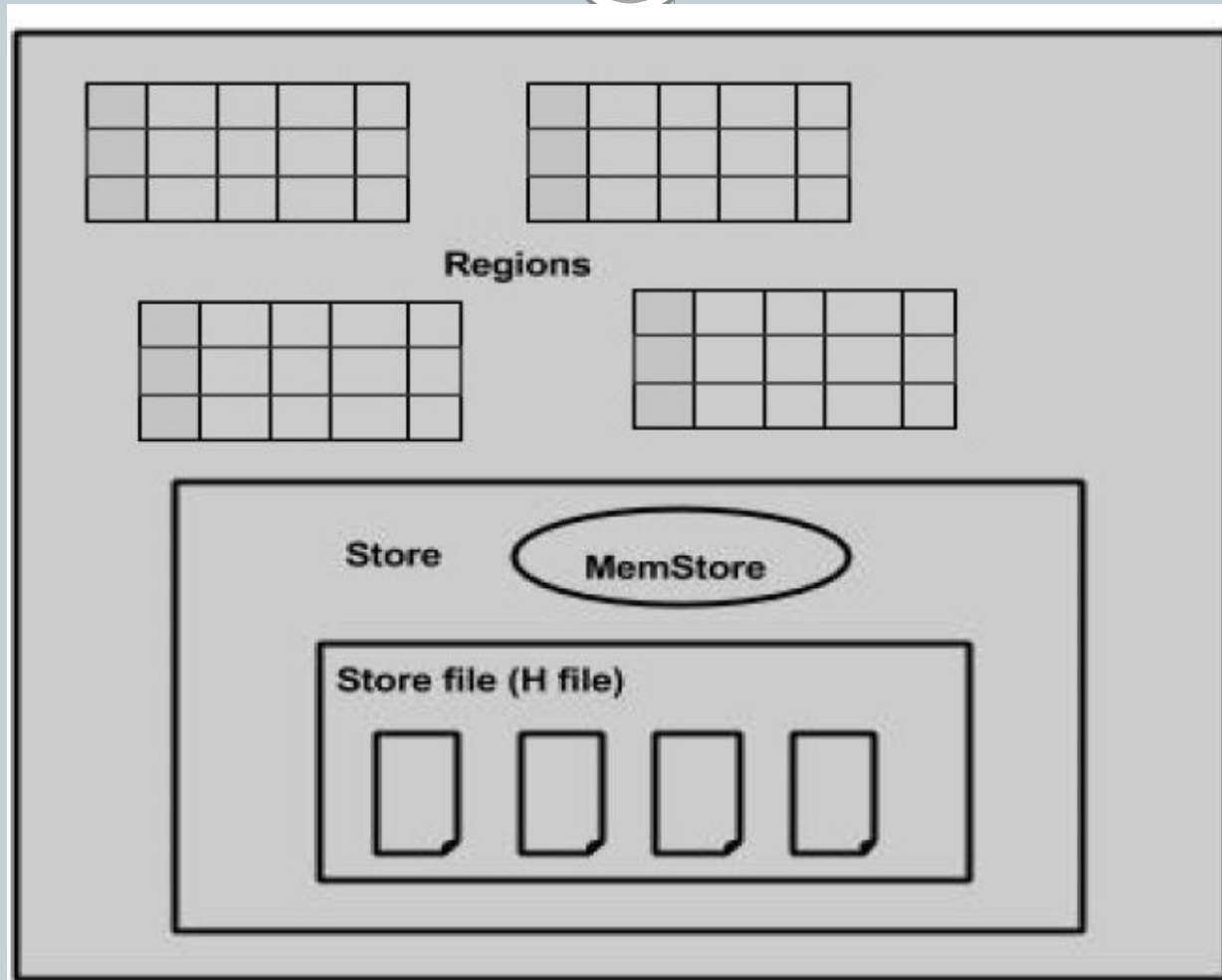


- **MasterServer**
- The master server -
- Assigns regions to the region servers and takes the help of Apache ZooKeeper for this task.
- Handles load balancing of the regions across region servers. It unloads the busy servers and shifts the regions to less occupied servers.
- Maintains the state of the cluster by negotiating the load balancing.
- Is responsible for schema changes and other metadata operations such as creation of tables and column families.



- **Region server**
- The region servers have regions that -
- Communicate with the client and handle data-related operations.
- Handle read and write requests for all the regions under it.
- Decide the size of the region by following the region size thresholds.
- The store contains memory store and HFiles.
- Memstore is just like a cache memory.

Region server



HBase Shell



- HBase contains a shell using which you can communicate with HBase.
- HBase uses the Hadoop File System to store its data.
- It will have a master server and region servers.
- The data storage will be in the form of regions (tables).
- These regions will be split up and stored in region servers.

Data Definition Language



- These are the commands that operate on the tables in HBase.
- **create:** Creates a table.
- **list:** Lists all the tables in HBase.
- **disable:** Disables a table.
- **is_disabled:** Verifies whether a table is disabled.
- **enable:** Enables a table.
- **is_enabled:** Verifies whether a table is enabled.
- **describe:** Provides the description of a table.
- **alter:** Alters a table.
- **exists:** Verifies whether a table exists.
- **drop:** Drops a table from HBase.

Data Manipulation Language



- **put:** Puts a cell value at a specified column in a specified row in a particular table.
- **get:** Fetches the contents of row or a cell.
- **delete:** Deletes a cell value in a table.
- **deleteall:** Deletes all the cells in a given row.
- **scan:** Scans and returns the table data.
- **count:** Counts and returns the number of rows in a table.
- **truncate:** Disables, drops, and recreates a specified table.