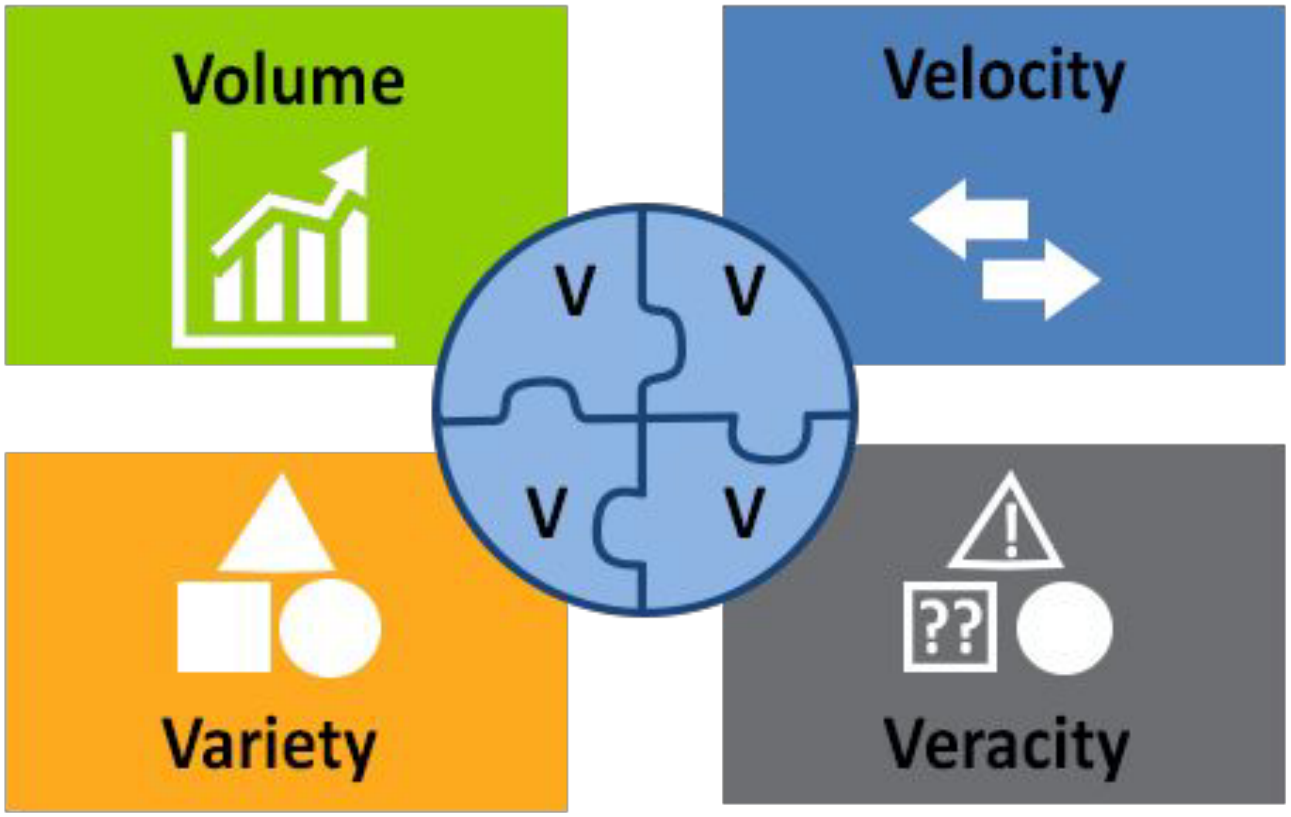
Big Data

- **Big data is dataset that having the ability to capture, manage& process the data in elapsed time.**
- **Big data includes the unstructured data, semi structured data, & structured data but it mainly focus on unstructured data.**
- **Big data size is vary from 30-50 terabytes(1012 or 1000 gigabytes per terabyte) to multiple petabytes (1015 or 1000 terabytes per petabyte).**

5 Vs of Big Data :



Value

5 Vs of Big Data :

- Big Data is the combination of these three factors; High-volume, High-Velocity ,High-Variety, High Veracity, High Value

5 Vs of Big Data :

1.Volume

- The name ‘Big Data’ itself is related to a size which is enormous.
- Volume is a huge amount of data.
- To determine the value of data, size of data plays a very crucial role. If the volume of data is very large then it is actually considered as a ‘Big Data’. This means whether a particular data can actually be considered as a Big Data or not, is dependent upon the volume of data.
- Hence while dealing with Big Data it is necessary to consider a characteristic ‘Volume’.
- *Example:* In the year 2016, the estimated global mobile traffic was 6.2 Exabytes(6.2 billion GB) per month. Also, by the year 2020 we will have almost 40000 ExaBytes of data.

5 Vs of Big Data :

2.Velocity

- Velocity refers to the high speed of accumulation of data.
- In Big Data velocity data flows in from sources like machines, networks, social media, mobile phones etc.
- There is a massive and continuous flow of data. This determines the potential of data that how fast the data is generated and processed to meet the demands.
- Sampling data can help in dealing with the issue like ‘velocity’.
- *Example:* There are more than 3.5 billion searches per day are made on Google. Also, FaceBook users are increasing by 22%(Approx.) year by year.

3 Variety:

- It refers to nature of data that is structured, semi-structured and unstructured data.
- It also refers to heterogeneous sources.
- Variety is basically the arrival of data from new sources that are both inside and outside of an enterprise. It can be structured, semi-structured and unstructured.
  - **Structured data**: This data is basically an organized data. It generally refers to data that has defined the length and format of data.
  - **Semi- Structured data**: This data is basically a semi-organised data. It is generally a form of data that do not conform to the formal structure of data. Log files are the examples of this type of data.
  - **Unstructured data**: This data basically refers to unorganized data. It generally refers to data that doesn't fit neatly into the traditional row and column structure of the relational database. Texts, pictures, videos etc. are the examples of unstructured data which can't be stored in the form of rows and columns.

4. Veracity:

- It refers to inconsistencies and uncertainty in data, that is data which is available can sometimes get messy and quality and accuracy are difficult to control.
- Big Data is also variable because of the multitude of data dimensions resulting from multiple disparate data types and sources.
- *Example:* Data in bulk could create confusion whereas less amount of data could convey half or Incomplete Information.

5. Value:

- After having the 4 V's into account there comes one more V which stands for Value!. The bulk of Data having no Value is of no good to the company, unless you turn it into something useful.
- Data in itself is of no use or importance but it needs to be converted into something valuable to extract Information. Hence, you can state that Value! is the most important V of all the 5V's.
-

Benefits of Big Data

- **Cost Savings :**
- Some tools of Big Data like Hadoop and Cloud-Based Analytics can bring cost advantages to business when large amounts of data are to be stored and these tools also help in identifying more efficient ways of doing business.

Benefits of Big Data

- **Time Reductions :**
- The high speed of tools like Hadoop and in-memory analytics can easily identify new sources of data which helps businesses analyzing data immediately and make quick decisions based on the learnings.

Benefits of Big Data

- **New Product Development :**
- By knowing the trends of customer needs and satisfaction through analytics you can create products according to the wants of customers.

Benefits of Big Data

- **Understand the market conditions :**
- By analyzing big data you can get a better understanding of current market conditions. For example, by analyzing customers' purchasing behaviors, a company can find out the products that are sold the most and produce products according to this trend. By this, it can get ahead of its competitors.

Benefits of Big Data

- **Control online reputation:**
- Big data tools can do sentiment analysis. Therefore, you can get feedback about who is saying what about your company. If you want to monitor and improve the online presence of your business, then, big data tools can help in all this.

- Since 1970, RDBMS is the solution for data storage and maintenance related problems. After the advent of big data, companies realized the benefit of processing big data and started opting for solutions like **Hadoop.**
- **Hadoop** uses distributed file system for storing big data, and MapReduce to process it.
- **Hadoop** excels in storing and processing of huge data of various formats such as arbitrary, semi-, or even unstructured.

Limitations of Hadoop

Hadoop can perform only batch processing, and data will be accessed only in a sequential manner. That means one has to search the entire dataset even for the simplest of jobs.

A huge dataset when processed results in another huge data set, which should also be processed sequentially. At this point, a new solution is needed to access any point of data in a single unit of time *randomaccess*.

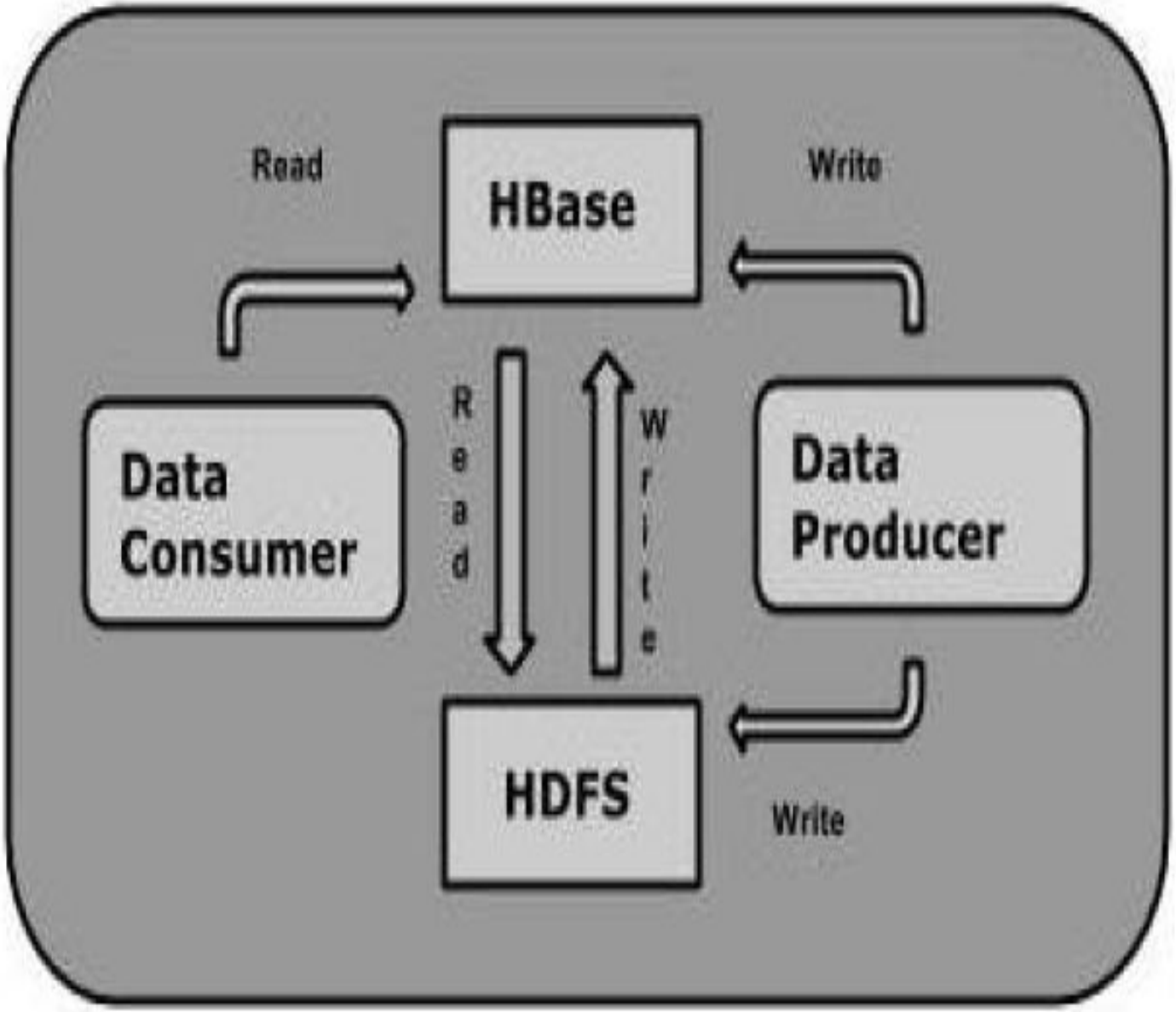
Hadoop Random Access Databases

- Applications such as **HBase**,
- **Cassandra, couchDB, Dynamo, and MongoDB** are some of the databases that store huge amounts of data and access the data in a random manner.

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.


Read
HBase
Write
Data Consumer
Read
Write
Data Producer
HDFS
Write

HBASE	HDFS
Hbase is a database built on top of HDFS	HDFS is a Distributed file system suitable for Storing large files
Build for Low Latency Operations Provide access to single rows from billion records Randomacess.	Build for High Latency batch processing Operations Data Access through Map Reduce.
Random reads and writes	Write once Read many times
Accessed through shell commands, client API in java, REST, Avro or Thrift	Primarly accessed through MR (Map Reduce) jobs
Storage and process both can be perform	It's only for storage areas
It provides fast lookups for large tables.	It does not support fast individual record lookups.
Hbase internally uses Hash tables and provides random access and it stores data in indexed in HDFS files for faster lookups	It provides only sequential access of data.

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. A table have multiple column families and each column family can have any number of columns. Subsequent column values are stored contiguously on the disk. Each cell value of the table has a timestamp. In short, in an HBase:

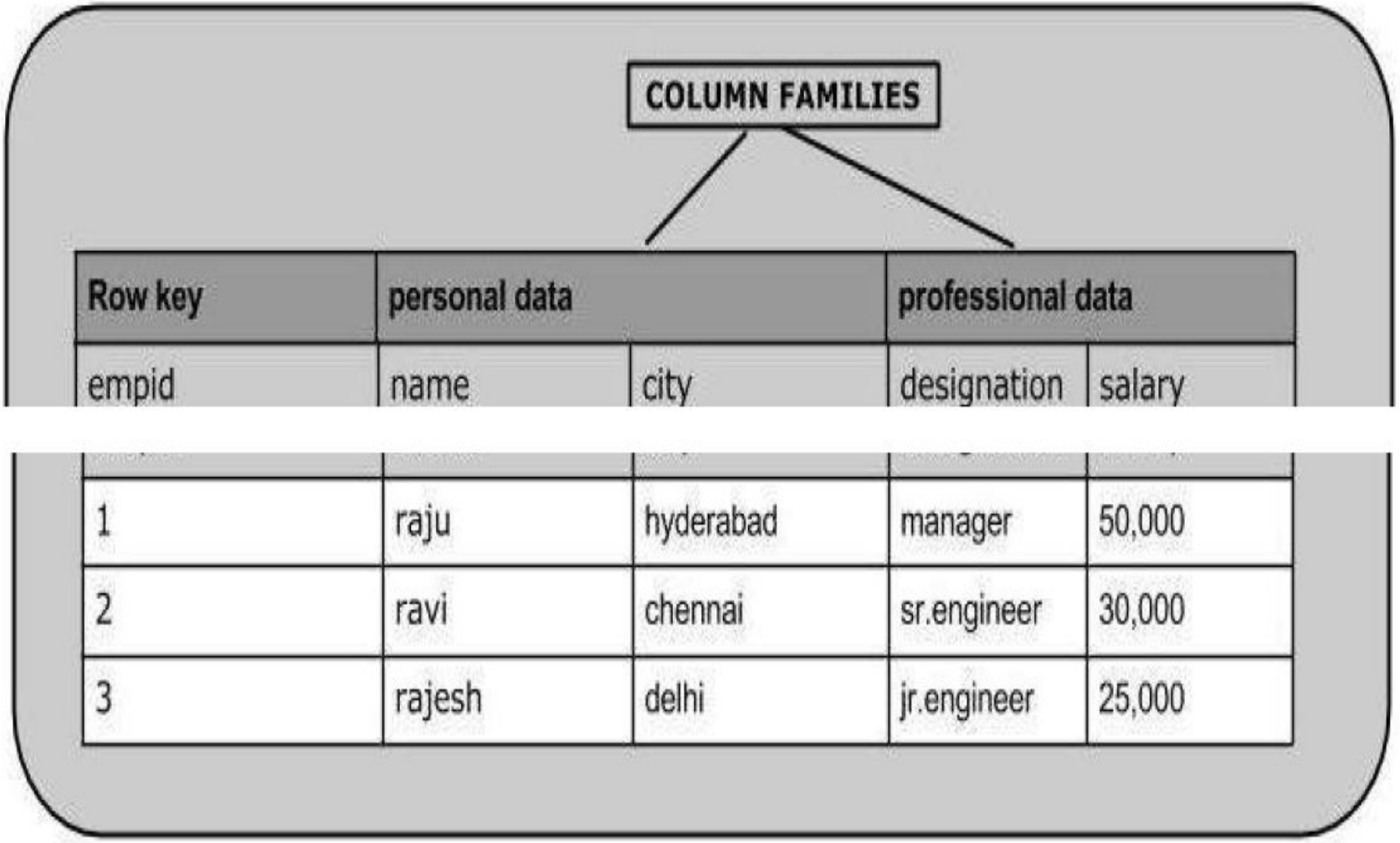
In Hbase

- 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.

example schema of table in HBase.

Rowid	Column Family			Column Family			Column Family			Column Family		
	col1	col2	col3	col1	col2	col3	col1	col2	col3	col1	col2	col3
1												
2												
3												

column families in a column-oriented database:



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

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.

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

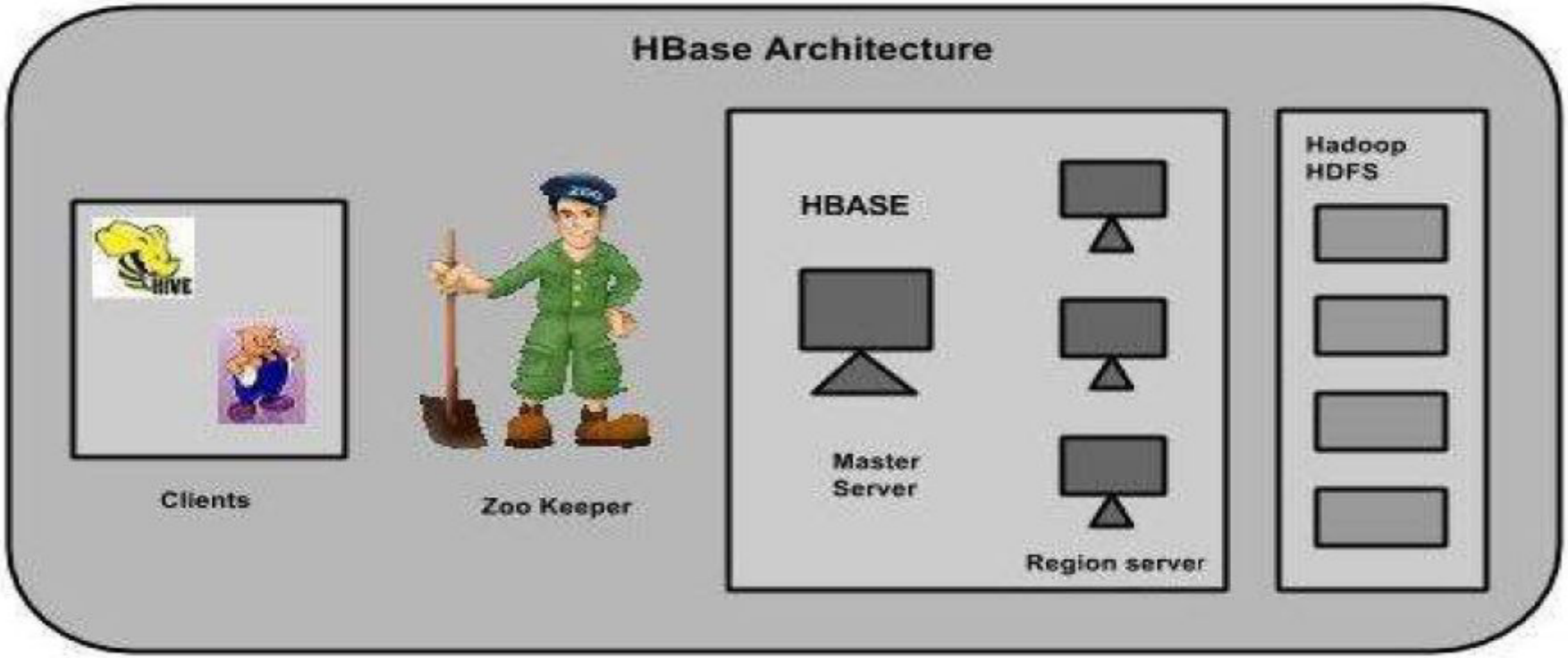
- HBase is linearly scalable.
- It has automatic failure support.
- It provides consistent read and writes.
- It integrates with Hadoop, both as a source and a destination.
- It has easy java API for client.
- It provides data replication across clusters.

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. divided by column families into “Stores”. Stores are saved as files in HDFS.



Client

- The client communicates in a bi-directional way with both HMaster and ZooKeeper for read and write operations, it directly contacts with HRegion servers. HMaster assigns regions to region servers and in turn check the health status of region servers.

HMaster:

- HMaster is the implementation of Master server in HBase architecture. It acts like monitoring agent to monitor all Region Server instances present in the cluster and acts as an interface for all the metadata changes. In a distributed cluster environment, Master runs on NameNode. Master runs several background threads.

HBase has three major components: the client library, a master server, and region servers. Region servers can be added or removed as per requirement.

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.

Regions

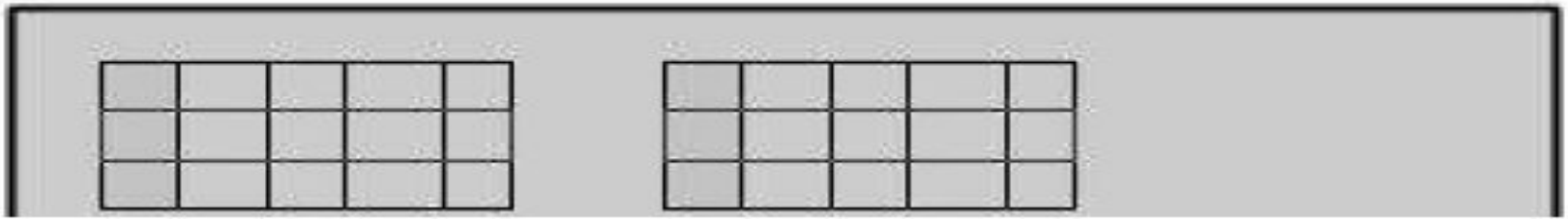
Regions are nothing but tables that are split up and spread across the region servers.

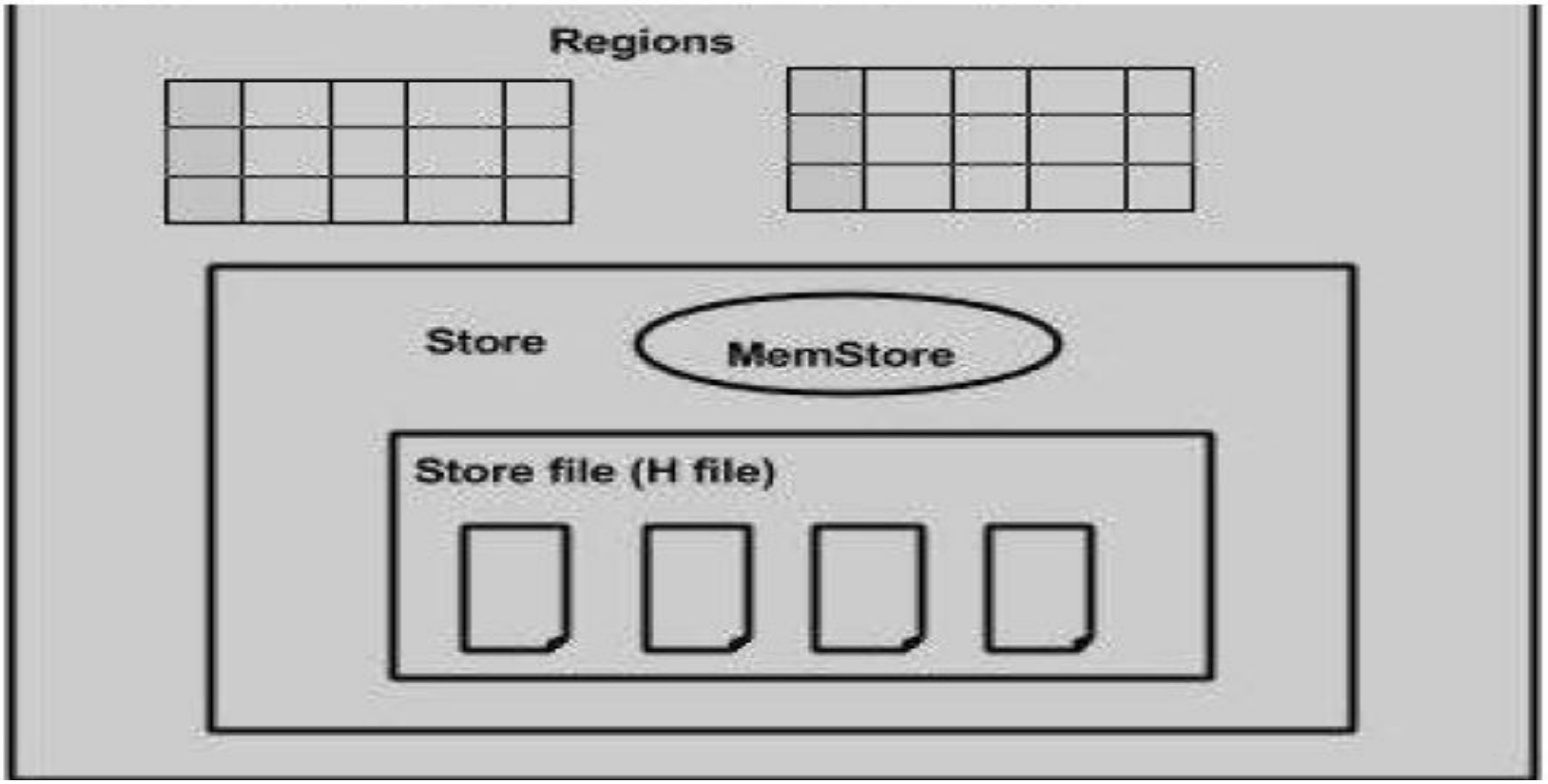
Region server

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.

When we take a deeper look into the region server, it contain regions and stores as shown below:



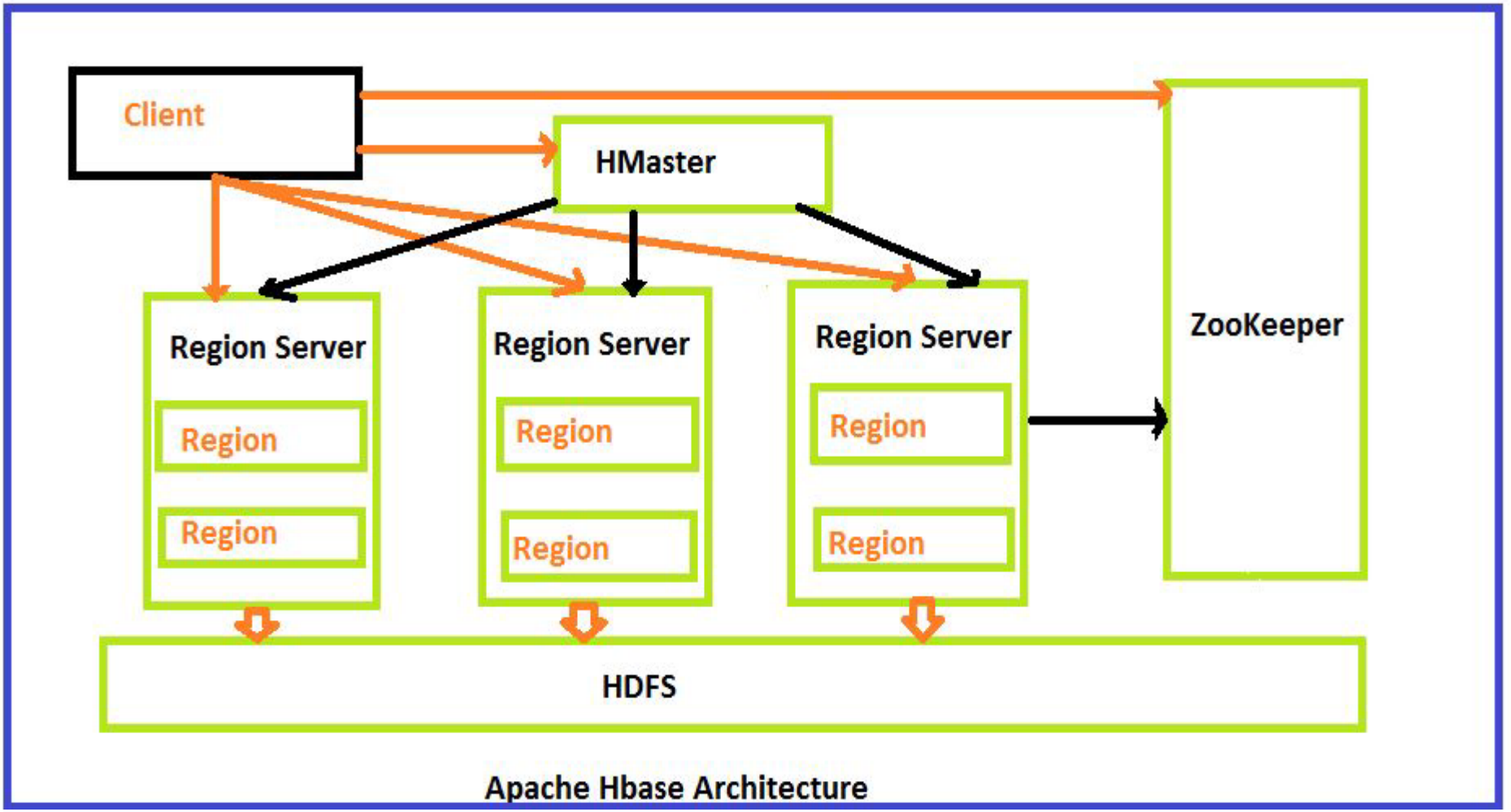

Regions
Store
MemStore
Store file (H file)

Store

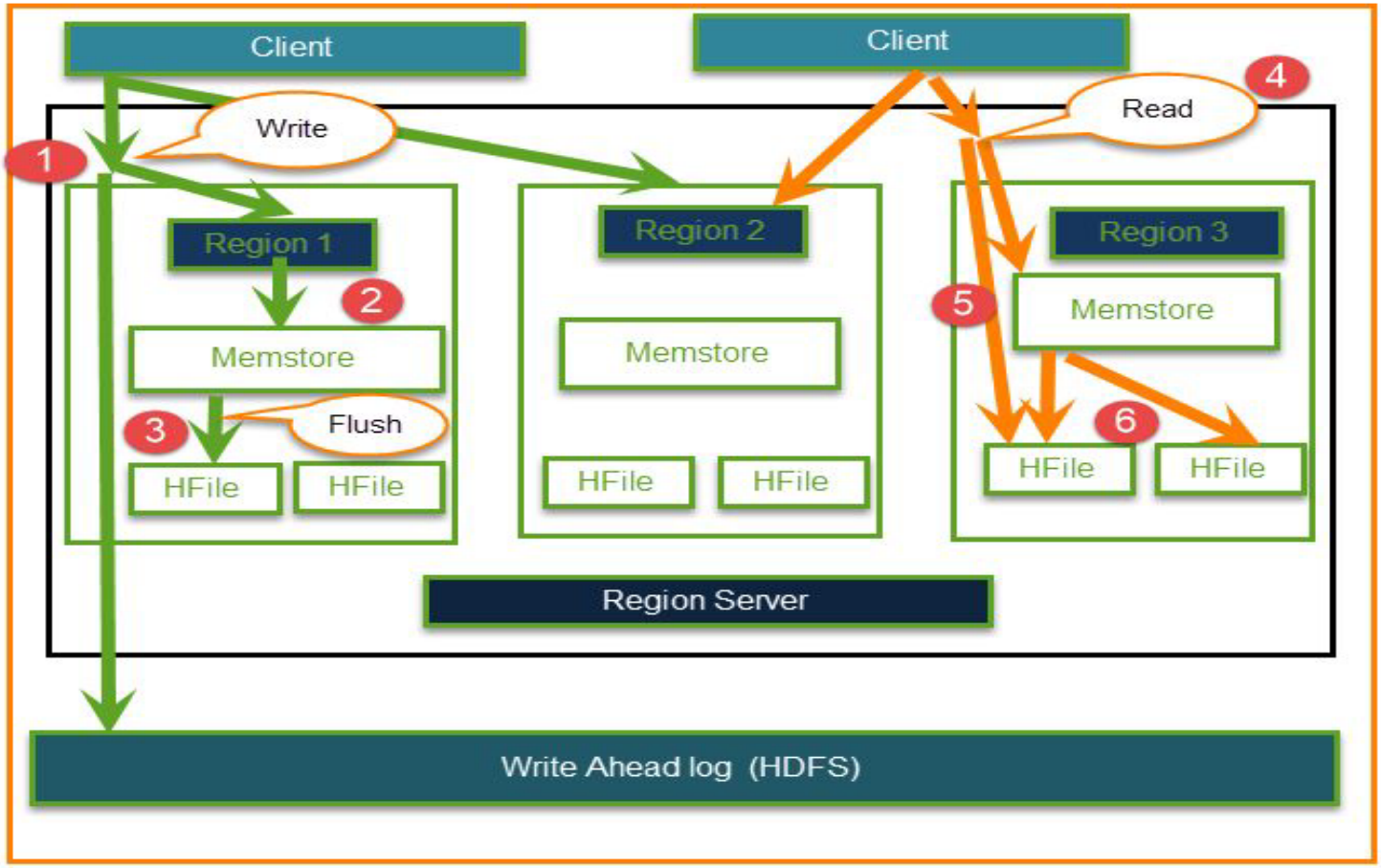
- The store contains memory store and HFiles.
- Memstore is just like a cache memory.
- Anything that is entered into the HBase is stored in Memstore initially.
- Later, the data is transferred and saved in Hfiles as blocks and the memstore is flushed.

Services provided by ZooKeeper

- Maintains Configuration information.
- Provides distributed synchronization.
- Client Communication establishment with region servers.
- To track server failure and network partitions.
- In pseudo and standalone modes, HBase itself will take care of zookeeper.



Apache Hbase Architecture


Client
Client
Write
Read
1
4
Region 1
Region 2
Region 3
2
5
Memstore
Memstore
Memstore
3
Flush
6
HFile
HFile
HFile
HFile
HFile
HFile
Region Server
Write Ahead log (HDFS)

Write and Read operations

The Read and Write operations from Client into Hfile can be shown in below diagram.

Step 1) Client wants to write data and in turn first communicates with Regions server and then regions

Step 2) Regions contacting memstore for storing associated with the column family

Step 3) First data stores into Memstore, where the data is sorted and after that it flushes into HFile. The main reason for using Memstore is to store data in Distributed file system based on Row Key. Memstore will be placed in Region server main memory while HFiles are written into HDFS.

Step 4) Client wants to read data from Regions

Step 5) In turn Client can have direct access to Mem store, and it can request for data.

Step 6) Client approaches HFiles to get the data. The data are fetched and retrieved by the Client.