SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with ‘A’ Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Artificial Intelligence and Data Science

Course Name – Big Data Analytics
III Year / V Semester

Unit 3 – DATA ANALYTICAL FRAMEWORKS

Topic - HBASE

HBASE

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 modeled after Google's Bigtable, which is a distributed storage system for structured data. Apache HBase provides Bigtable-like capabilities on top of Hadoop and HDFS.

- Some of the companies that use HBase as their core program are Facebook, Netflix, Yahoo, Adobe, and Twitter. The goal of HBase is to host large tables with billions of rows and millions of columns on top of clusters of commodity hardware.

- It is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System.

HBASE

Features of HBase

HBase has a number of features like:

- Scalable: HBase allows data to be scaled across various nodes as it is stored in HDFS.

- Automatic failure support: Write ahead Log across clusters are present that provides automatic support against failure.

- Consistent read and write: HBase provides consistent read and write of data.

- JAVA API for client access: HBase provides easy to use JAVA API for clients.

- Block cache and Bloom filters: It supports block cache and bloom filters for high volume query optimization.

HBASE

HBASE Architecture

- It has the following components

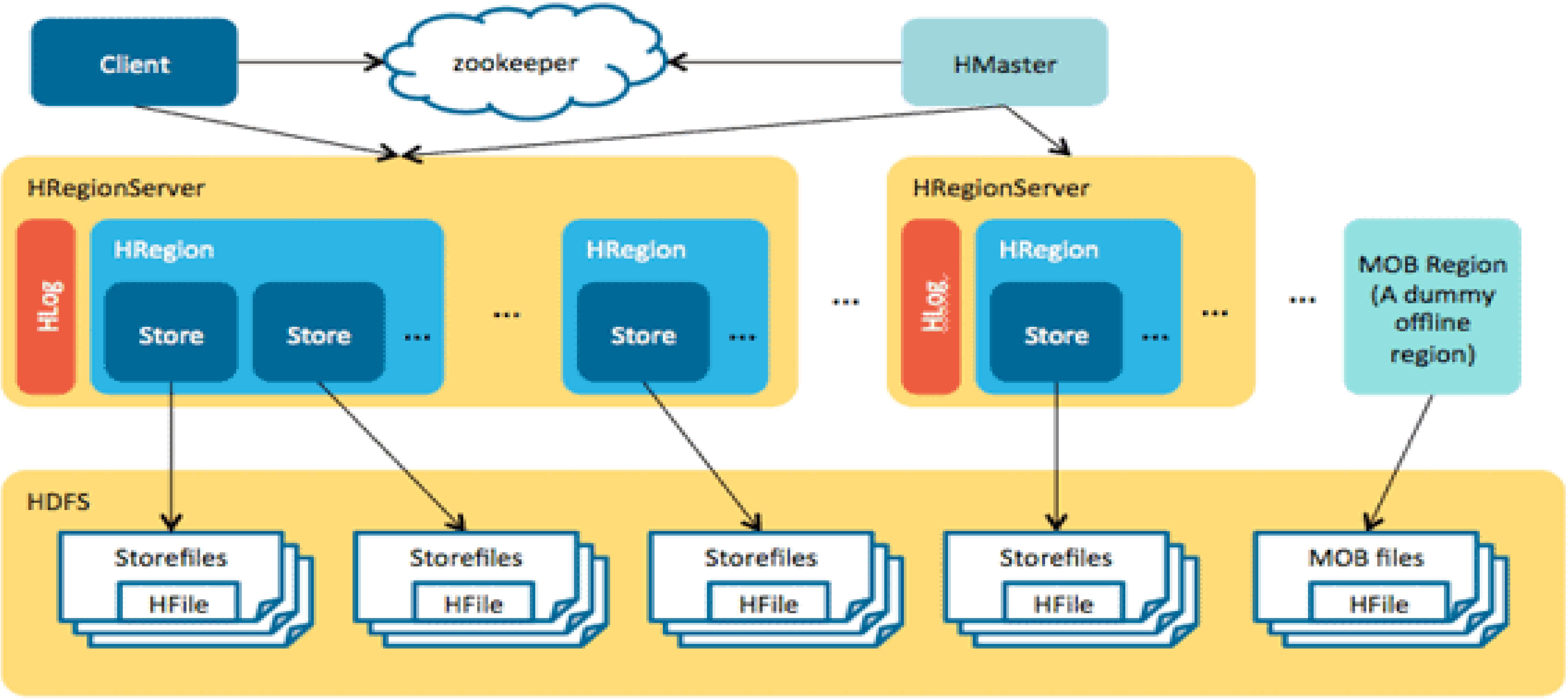
- The Apache Zookeeper - monitors the system, and

- HBase Master - assigns regions and load balancing.

- The Region server - serves data to read and write. The Region Server is all the different computers in the Hadoop cluster. It consists of Region, HLog, Store, MemoryStore, and different files.

- All this is a part of the HDFS storage system.

HBASE Architecture



HBASE

Regions

- HBase tables are divided horizontally by row key range into “Regions”. Regions are assigned to the nodes in the cluster, called “Region Servers”. These servers serve data for reading and writing.

- A table can be divided into a number of regions. A Region is a sorted range of rows storing data between a start key and an end key.

- A Region has a default size of 256MB which can be configured according to the need.

- A Group of regions is served to the clients by a Region Server.

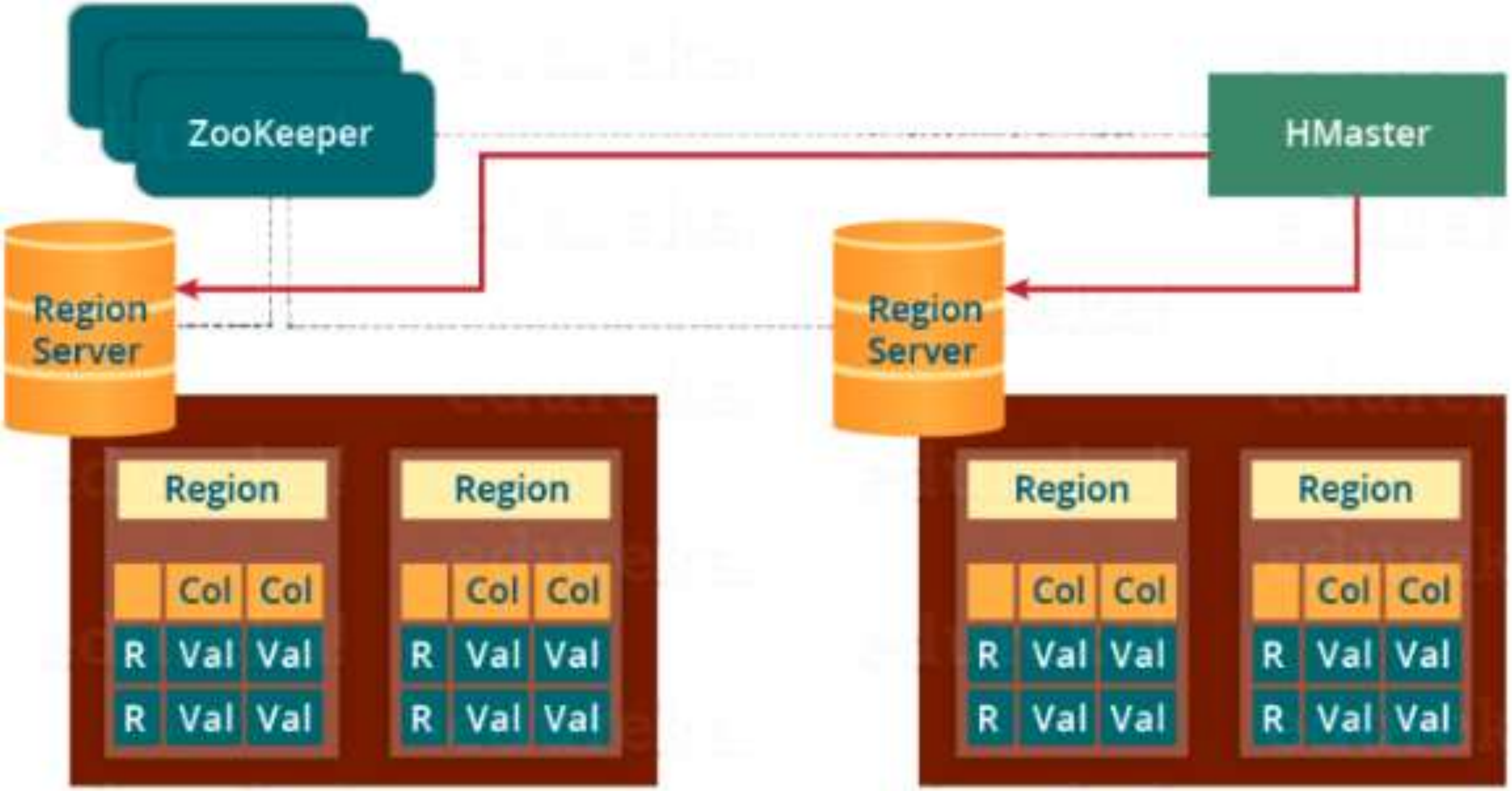
- A Region Server can serve approximately 1000 regions to the client.

HBASE

HMaster

- HMaster handles a collection of Region Server which resides on DataNode.
- HBase HMaster performs DDL operations (create and delete tables) and assigns regions to the Region servers as you can see in the above image.
- It coordinates and manages the Region Server (similar as NameNode manages DataNode in HDFS).
- It assigns regions to the Region Servers on startup and re-assigns regions to Region Servers during recovery and load balancing.
- It monitors all the Region Server’s instances in the cluster (with the help of Zookeeper) and performs recovery activities whenever any Region Server is down.
- It provides an interface for creating, deleting and updating tables.

HBASE


ZooKeeper
HMaster
Region Server
Region Server
Region
Region
Region
Region
Col
Col
R
Val
Val
R
Val
Val

HBASE

ZooKeeper – The Coordinator

- Zookeeper acts like a coordinator inside HBase distributed environment. It helps in maintaining server state inside the cluster by communicating through sessions.

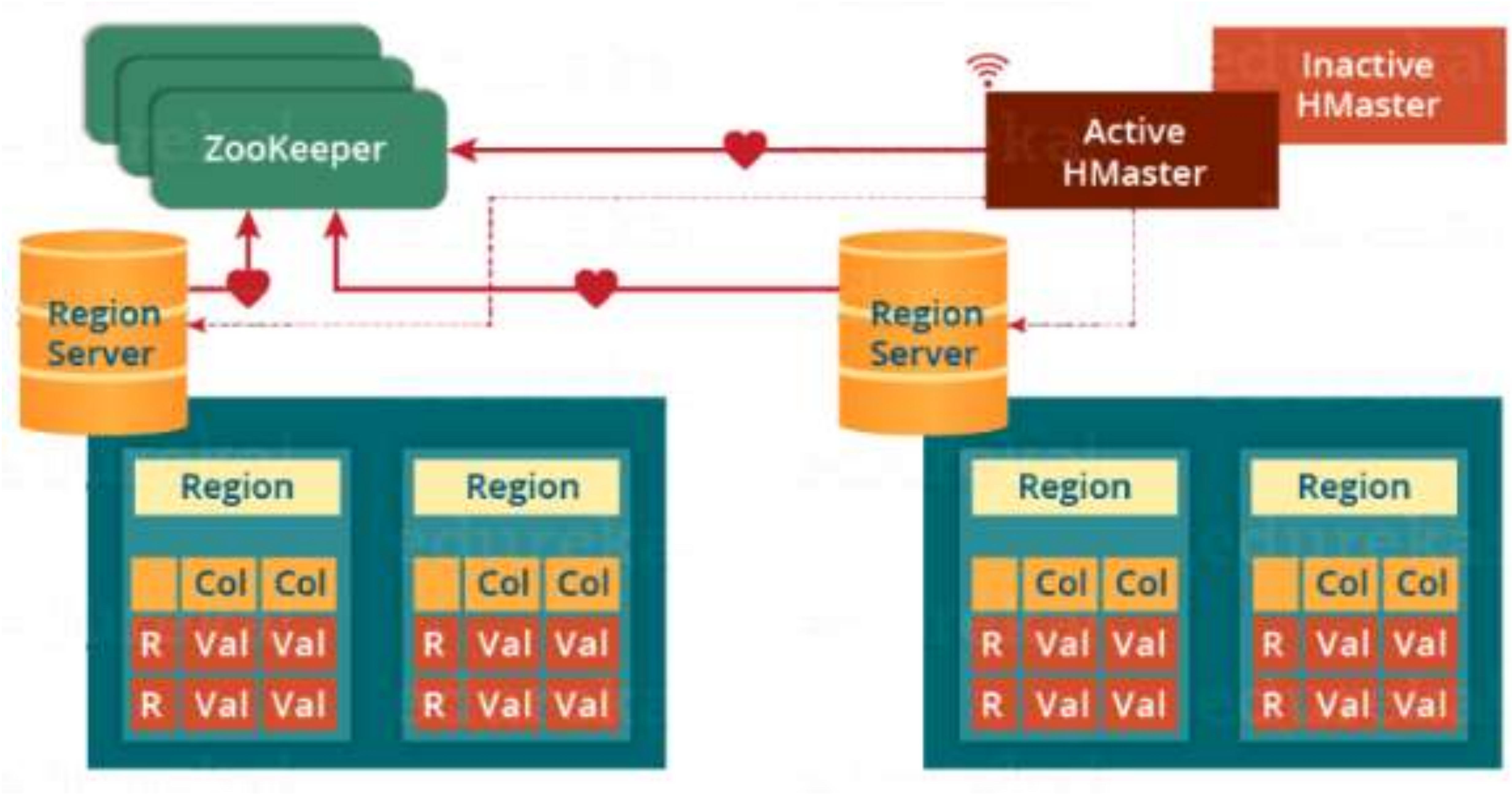
- Every Region Server along with HMaster Server sends continuous heartbeat at regular interval to Zookeeper and it checks which server is alive and available as mentioned in above image. It also provides server failure notifications so that, recovery measures can be executed.

- Referring from the above image you can see, there is an inactive server, which acts as a backup for active server. If the active server fails, it comes for the rescue.

- The active HMaster sends heartbeats to the Zookeeper while the inactive HMaster listens for the notification send by active HMaster. If the active HMaster fails to send a heartbeat the session is deleted and the inactive HMaster becomes active.

- While if a Region Server fails to send a heartbeat, the session is expired and all listeners are notified about it. Then HMaster performs suitable recovery actions which we will discuss later in this blog.

HBASE


ZooKeeper
Inactive HMaster
Active HMaster
Region Server
Region Server
Region
Col
Col
R
Val
Val
R
Val
Val
Region
Col
Col
R
Val
Val
R
Val
Val
Region
Col
Col
R
Val
Val
R
Val
Val
Region
Col
Col
R
Val
Val
R
Val
Val

HBASE

Meta Table

- The META table is a special HBase catalog table. It maintains a list of all the Regions Servers in the HBase storage system, as you can see in the above image. Looking at the figure you can see, .

- META file maintains the table in form of keys and values. Key represents the start key of the region and its id whereas the value contains the path of the Region Server.

HBASE

Components of Region Server

- WAL: As you can conclude from the above image, Write Ahead Log (WAL) is a file attached to every Region Server inside the distributed environment. The WAL stores the new data that hasn't been persisted or committed to the permanent storage. It is used in case of failure to recover the data sets.

- Block Cache: From the above image, it is clearly visible that Block Cache resides in the top of Region Server. It stores the frequently read data in the memory. If the data in BlockCache is least recently used, then that data is removed from BlockCache.

- MemStore: It is the write cache. It stores all the incoming data before committing it to the disk or permanent memory. There is one MemStore for each column family in a region. As you can see in the image, there are multiple MemStores for a region because each region contains multiple column families. The data is sorted in lexicographical order before committing it to the disk.

- HFile: From the above figure you can see HFile is stored on HDFS. Thus it stores the actual cells on the disk. MemStore commits the data to HFile when the size of MemStore exceeds.

HBASE

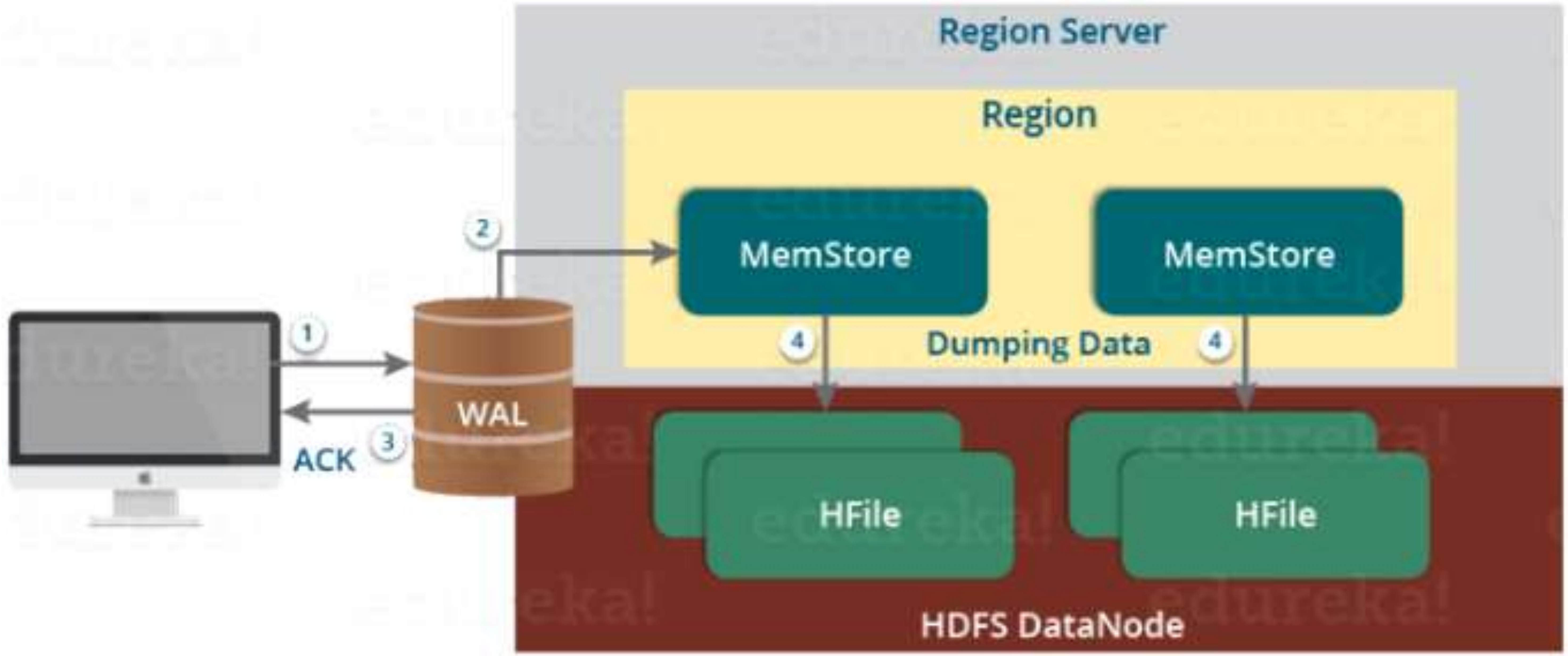
HBase Write Mechanism

The write mechanism goes through the following process sequentially

- Step 1: Whenever the client has a write request, the client writes the data to the WAL (Write Ahead Log).
- The edits are then appended at the end of the WAL file.
- This WAL file is maintained in every Region Server and Region Server uses it to recover data which is not committed to the disk.
- Step 2: Once data is written to the WAL, then it is copied to the MemStore.
- Step 3: Once the data is placed in MemStore, then the client receives the acknowledgment.
- Step 4: When the MemStore reaches the threshold, it dumps or commits the data into a HFile.

HBASE

HBase Write Mechanism



HBASE

Read Mechanism

- As discussed in our search mechanism, first the client retrieves the location of the Region Server from .META Server if the client does not have it in its cache memory. Then it goes through the sequential steps as follows:
- For reading the data, the scanner first looks for the Row cell in Block cache. Here all the recently read key value pairs are stored.
- If Scanner fails to find the required result, it moves to the MemStore, as we know this is the write cache memory. There, it searches for the most recently written files, which has not been dumped yet in HFile.
- At last, it will use bloom filters and block cache to load the data from HFile.

THANK YOU