Recap

- Pig
- Hive
- Impala

Agenda for today

- Sqoop
- Hbase

Introduction

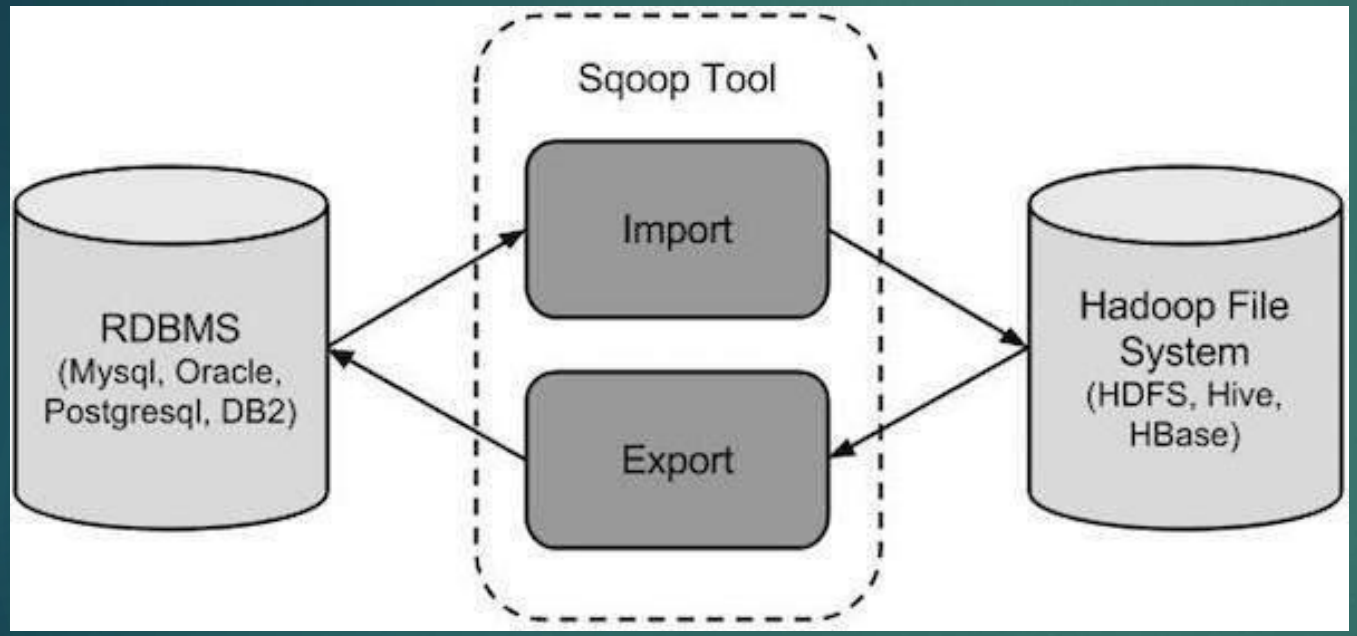


Image Ref: Tutorialspoint

Export

Parameter	Description
--table	Target table name
--export-dir	HDFS source dir name
--fields-terminated-by	Field delimiter
-m, --num-mappers	#mappers to launch
--staging-table	Staging table for temp storage
--jar-file	Use mentioned jar file to export
--update-key	Update data in RDBMS based on mentioned key

- sqoop export --connect jdbc:mysql://localhost :3306/retail_db --username retail_dba --password cloudera --table test --fields-terminated-by ',' --export-dir <HDFS DIRCTORY NAME>

Import

Parameter	Description
--table	Source table name
--target-dir	HDFS target dir name
--fields-terminated-by	Field delimiter
-m, --num-mappers	#mappers to launch
--split-by	Unique column name
--delete-target-dir	Delete target HDFS dir if exists
--where	Condition to apply while fetching data from RDBMS

- sqoop import --connect jdbc:mysql://localhost :3306/<DATABASE NAME> --username root **-p** --table <TABLE NAME> --m 1 --target-dir <HDFS DIRCTORY NAME>

Jobs

- Compile sqoop jobs for regular execution
- Create Job

  sqoop job --create myjob -- import --connect jdbc:mysql:// localhost :3306/retail_db --username retail_dba --password cloudera --table departments --target-dir <HDFS DIRCTORY NAME>
- List all created jobs

  sqoop job --list
- Show details of one specific job

  sqoop job --show myjob
- Execute created job

  sqoop job --exec myjob

Codegen

- Generate java code for sqoop commands

  sqoop codegen --connect jdbc:mysql:// localhost :3306/retail_db --username retail_dba --password cloudera --table departments

- What could be the use case of codegen tool?

Eval

- Evaluate a single command on RDBMS

  sqoop eval -- connect jdbc:mysql:// localhost :3306/retail_db --username retail_dba -- password cloudera -e "INSERT INTO Test VALUES(999, 'name999')“

- What could be the use case of eval tool?

Others

- sqoop-import-all-tables
- sqoop-import-mainframe
- Validation
- sqoop-metastore
- sqoop-merge

Hbase

Introduction

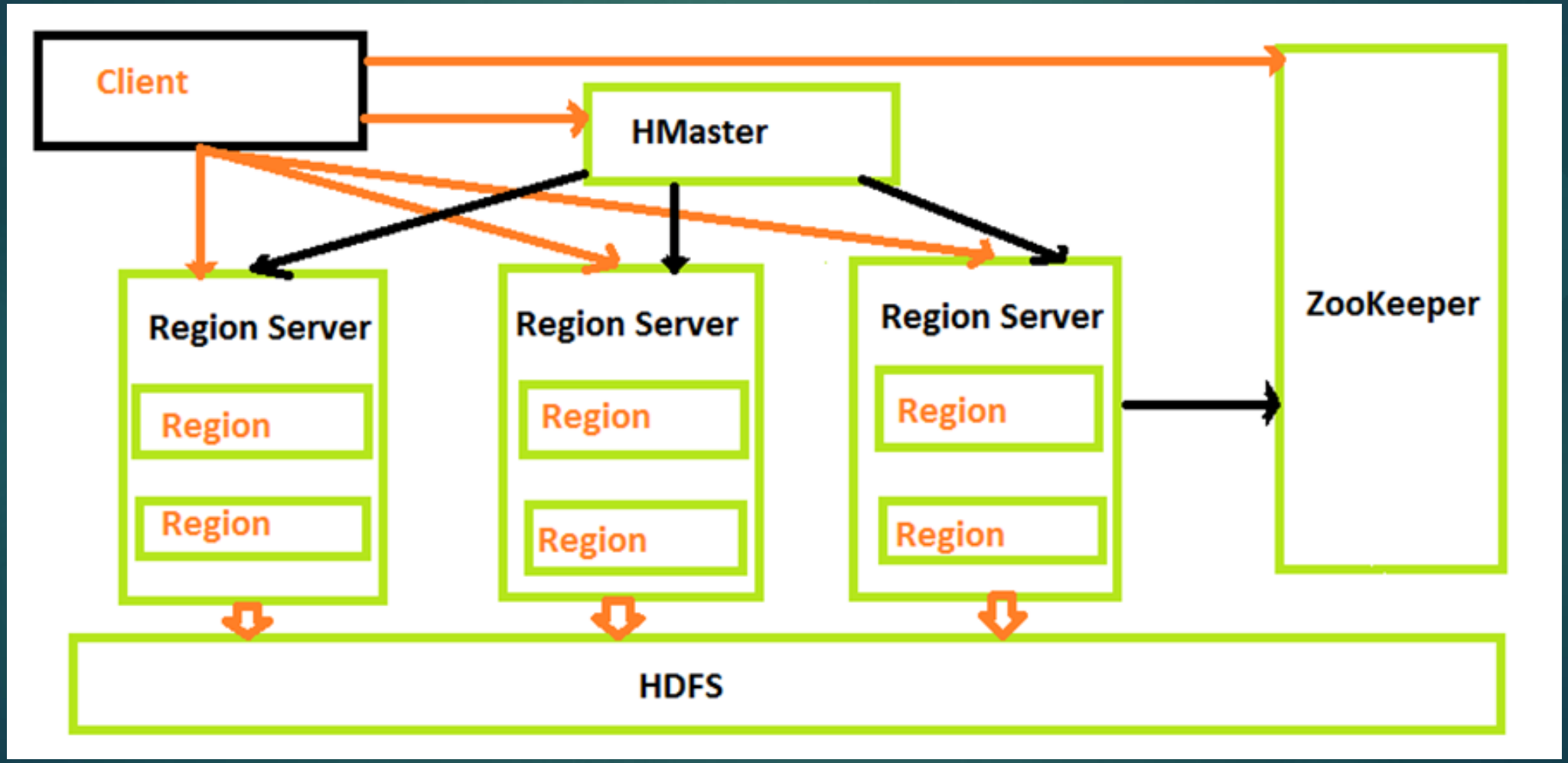
- Column-oriented database built on top of HDFS
- Horizontally scalable
- Built for low latency operations
- Random read and write
- Strictly consistent
- Support for Java API for client access
- Compatibility with MapReduce jobs

Data structure

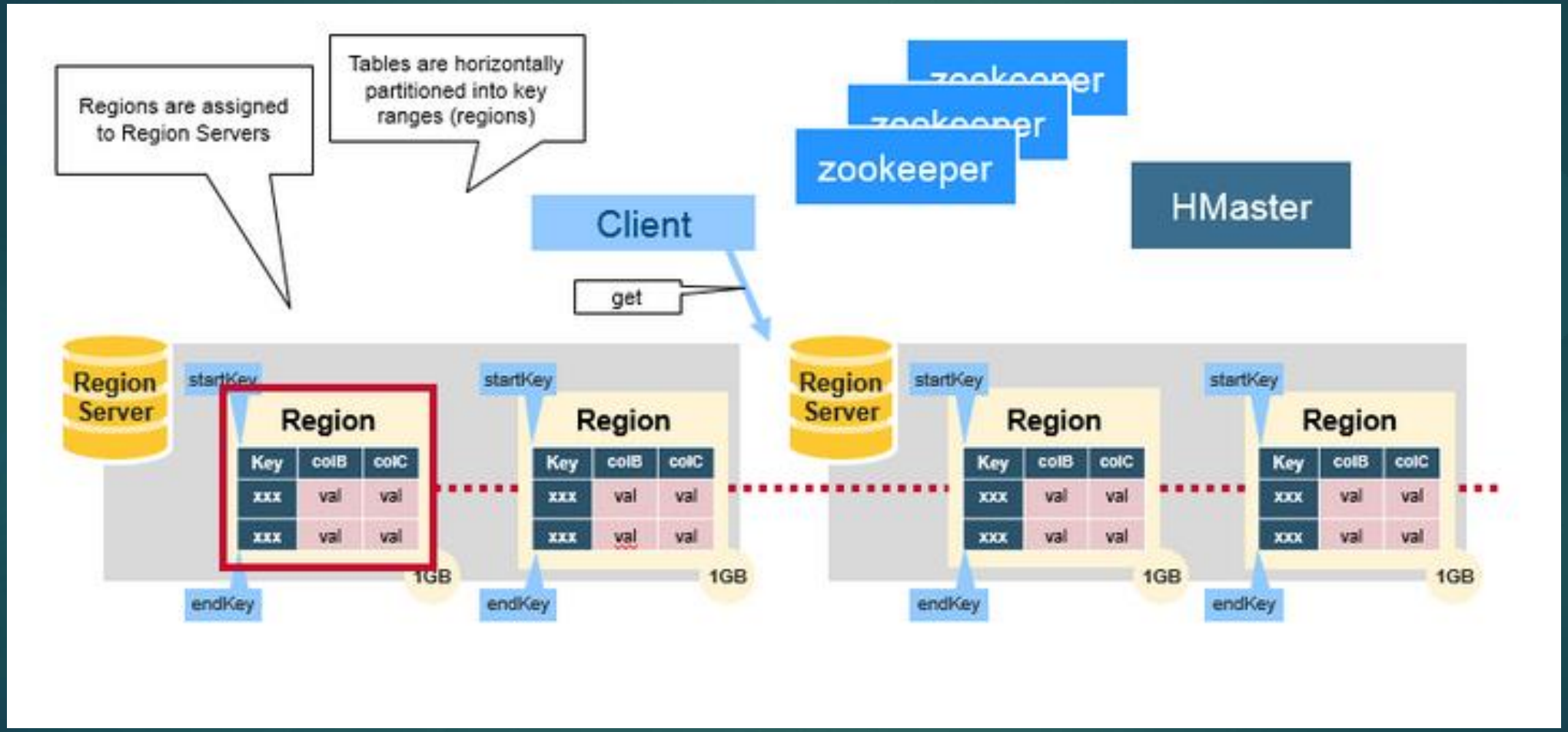
Data structure: Cont…

- Table: Collection of rows present
- Row: Collection of column families
- Column Family: Collection of columns
- Column: Collection of key-value pairs
- Namespace: Logical grouping of tables
- Cell: A {row, column, version} tuple exactly specifies a cell definition in HBase

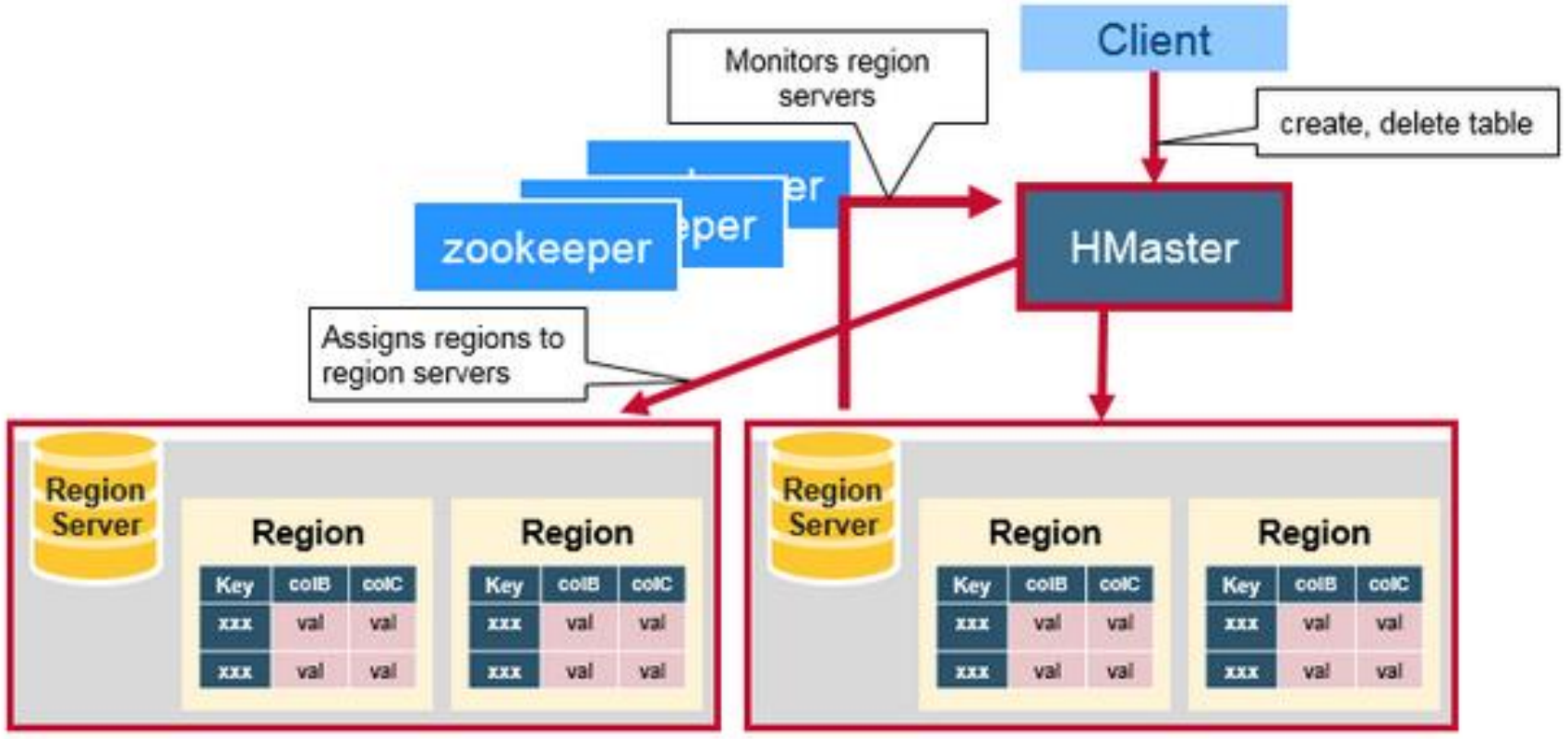
Architecture



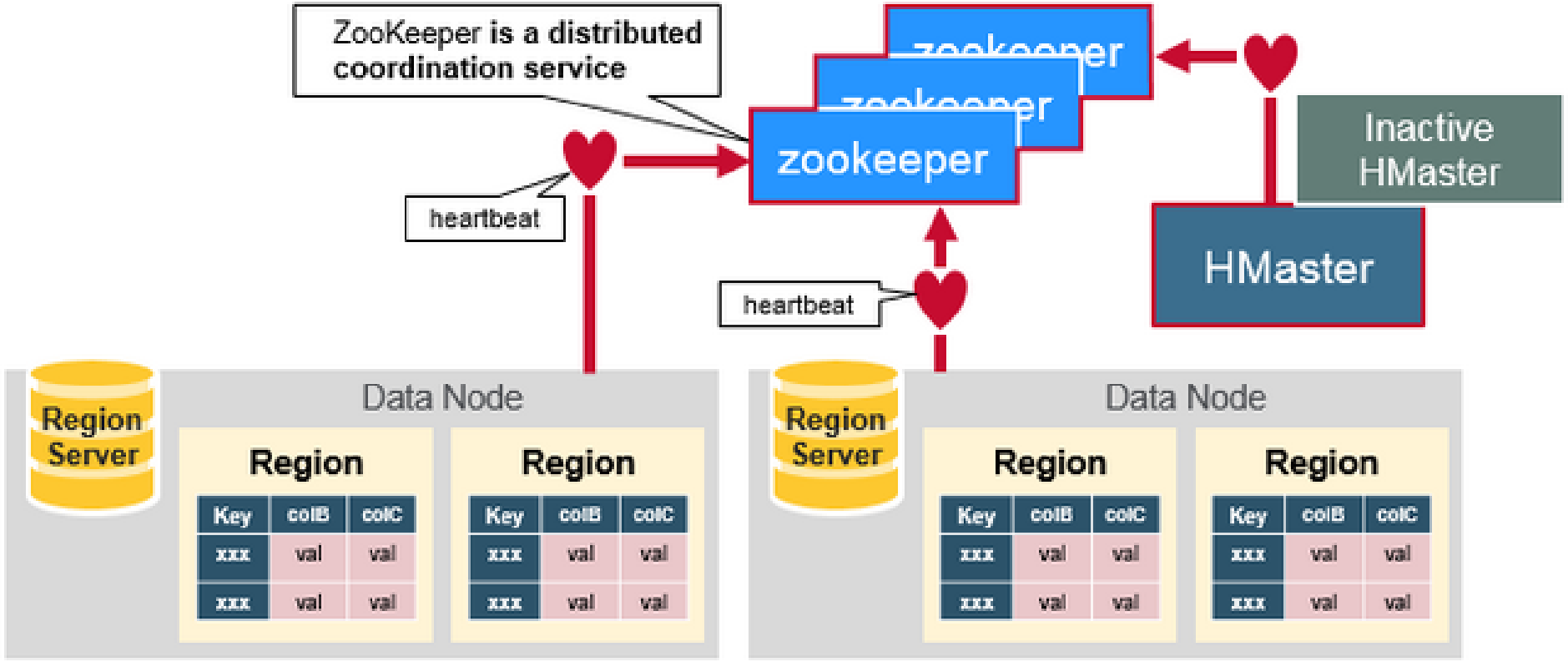
Architecture: Region Server



Architecture: HMaster



Architecture: Zookeeper



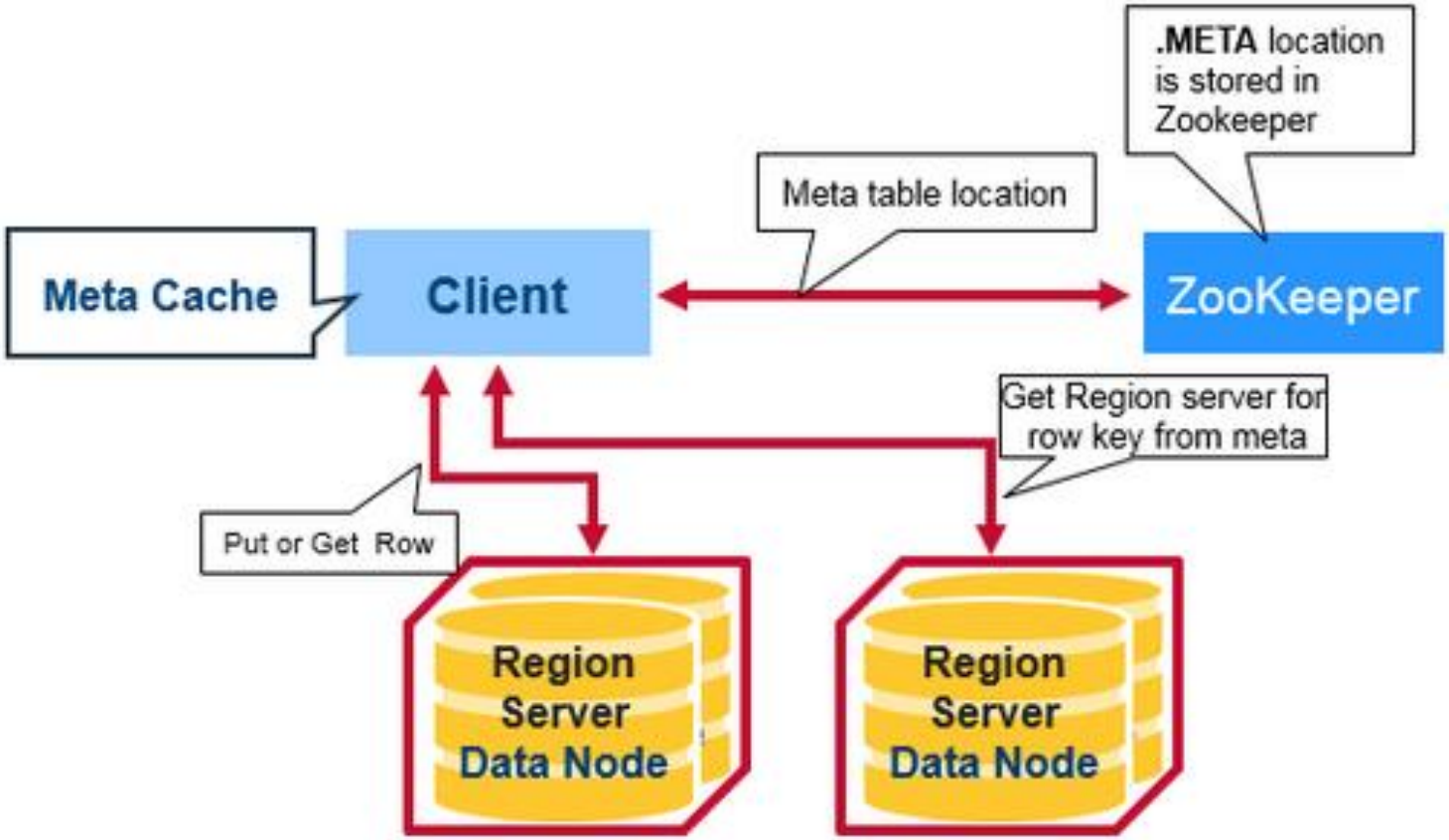
META table

- Keeps a list of all regions in the system
- Structure:
  - Key: region start key,region id
  - Values: RegionServer

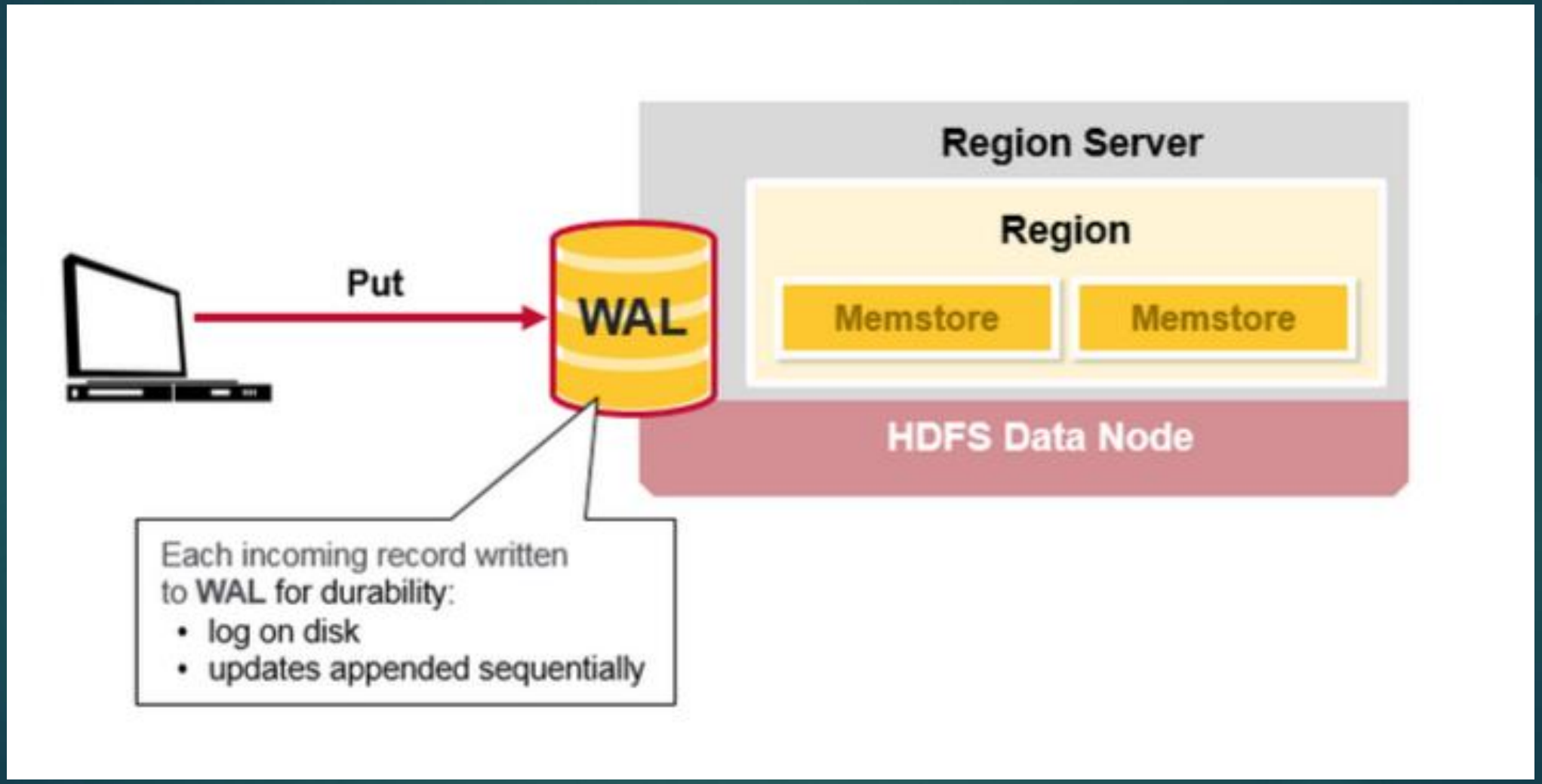
Region Server Components

- WAL: Write Ahead Log is a file on distributed file system
- BlockCache: is the read cache
- MemStore: is the write cache
- Hfiles store the rows as sorted KeyValues on disk.

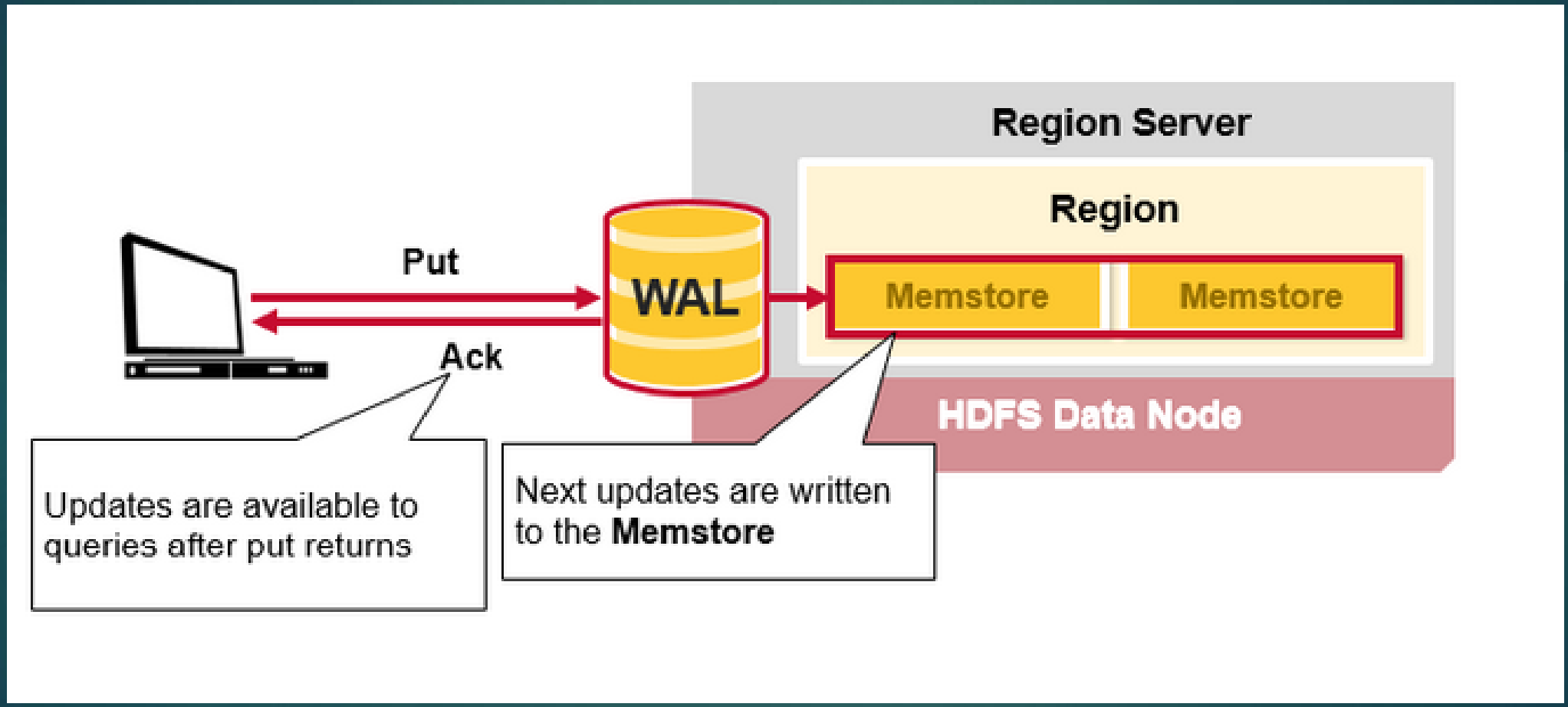
Write Operation



Write Operation: Cont…

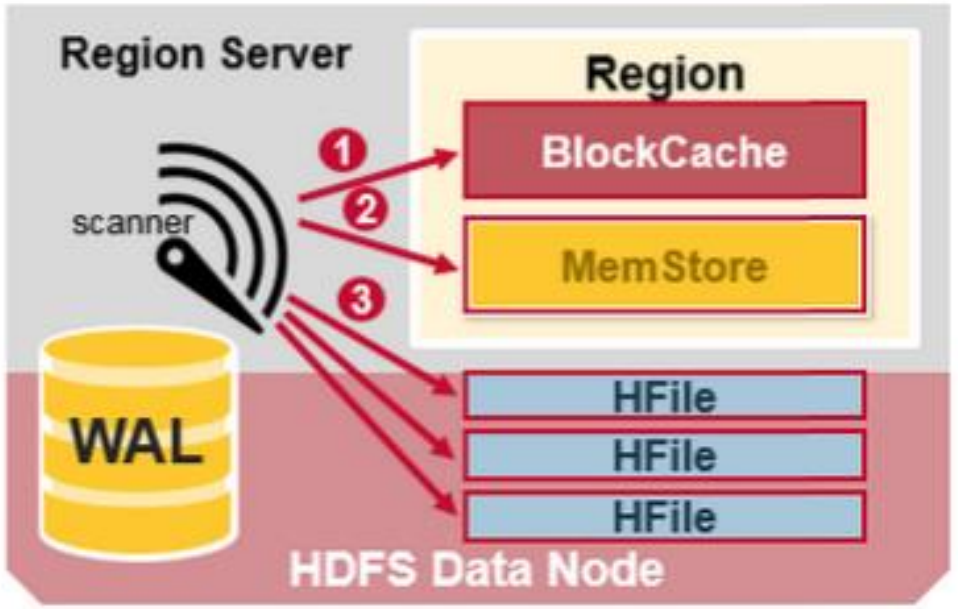


Write Operation: Cont…

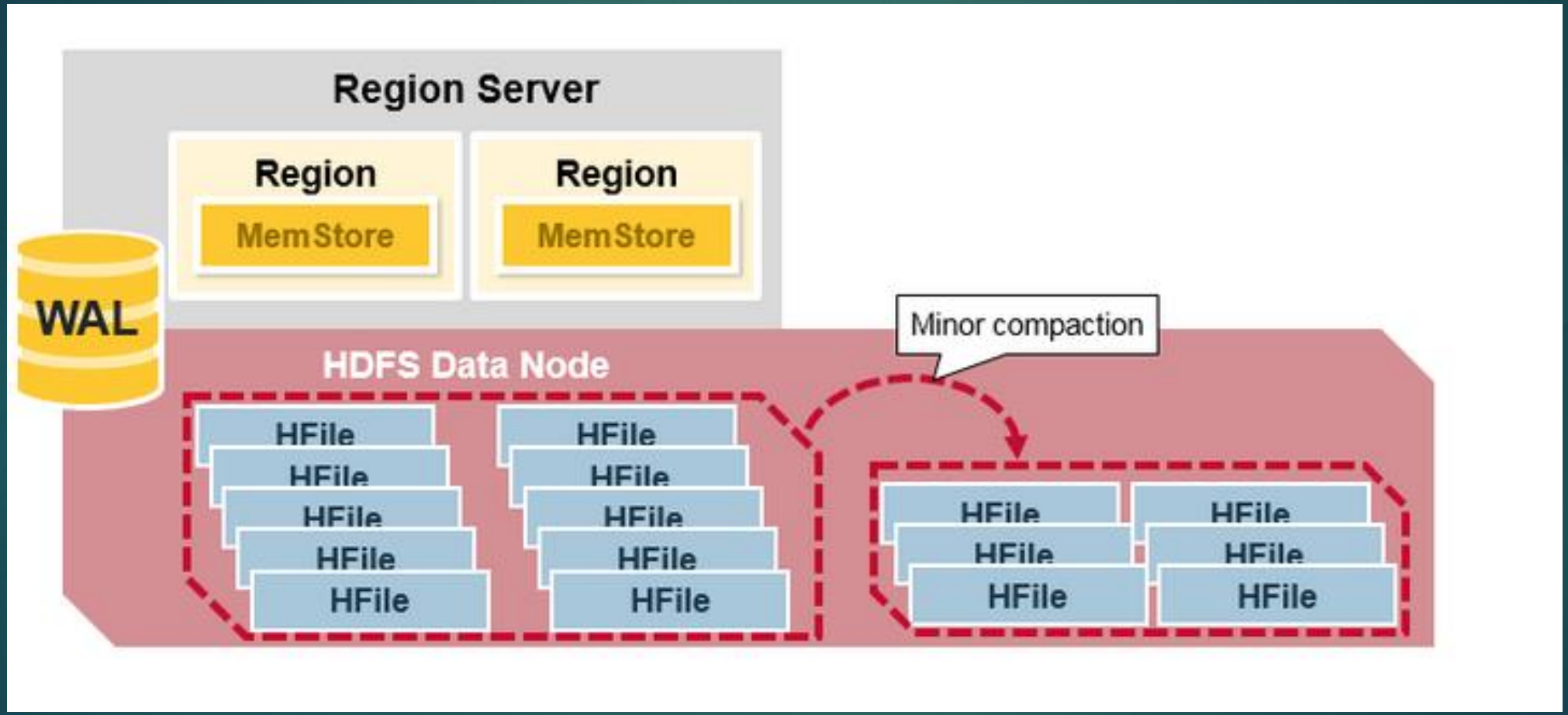


Read Operation

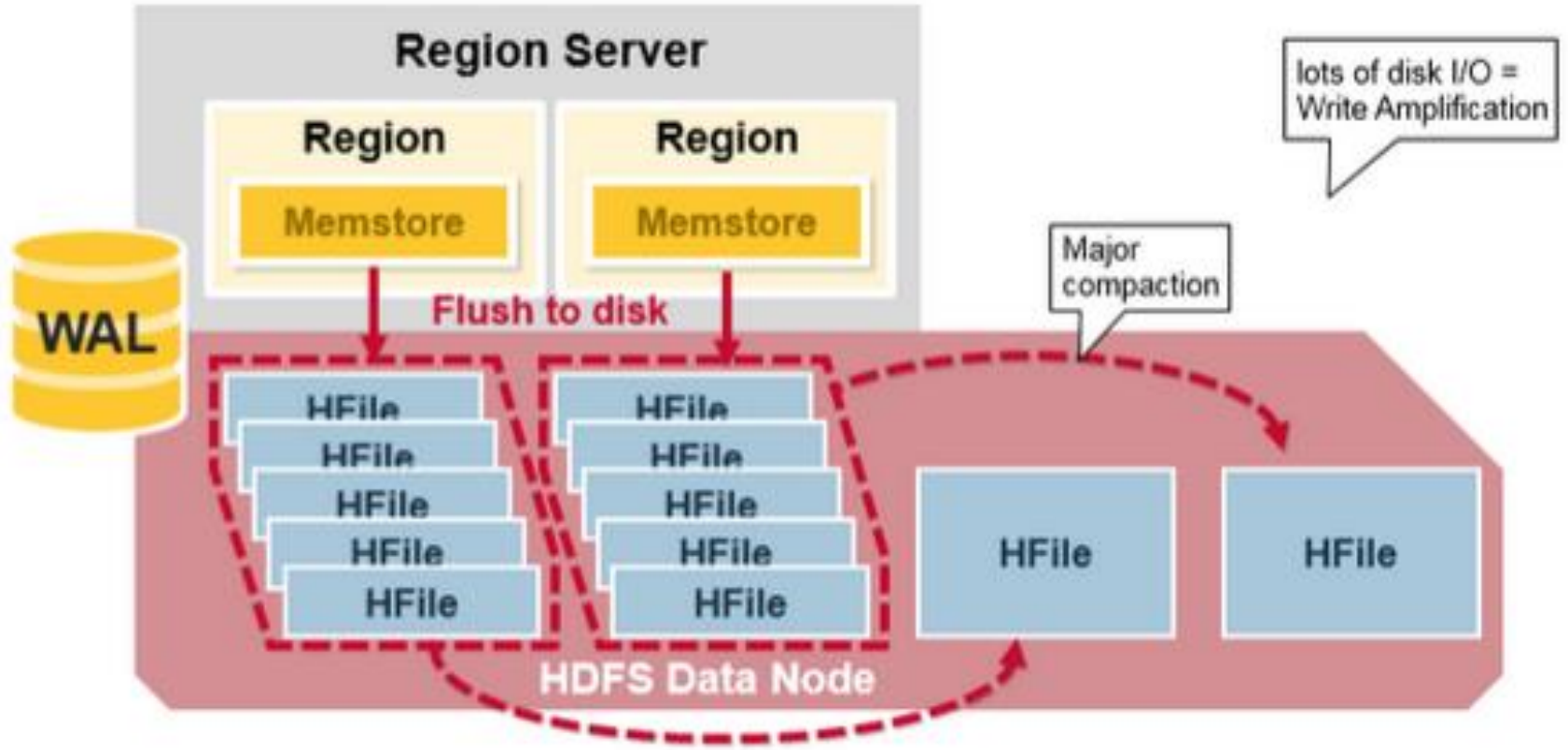
1. First the scanner looks for the Row KeyValues in the Block cache
2. Next the scanner looks in the MemStore
3. If all row cells not in MemStore or blockCache, look in HFiles



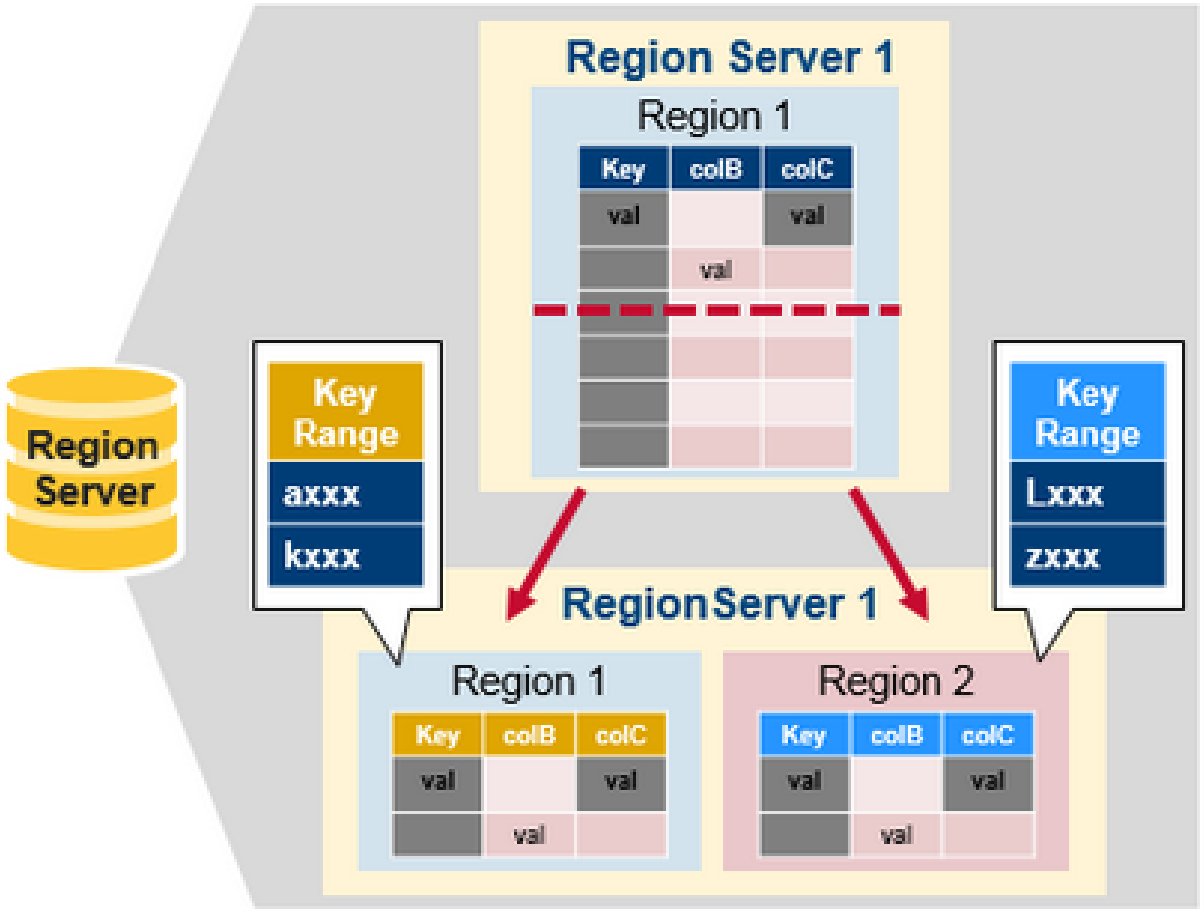
Minor Compaction



Major Compaction

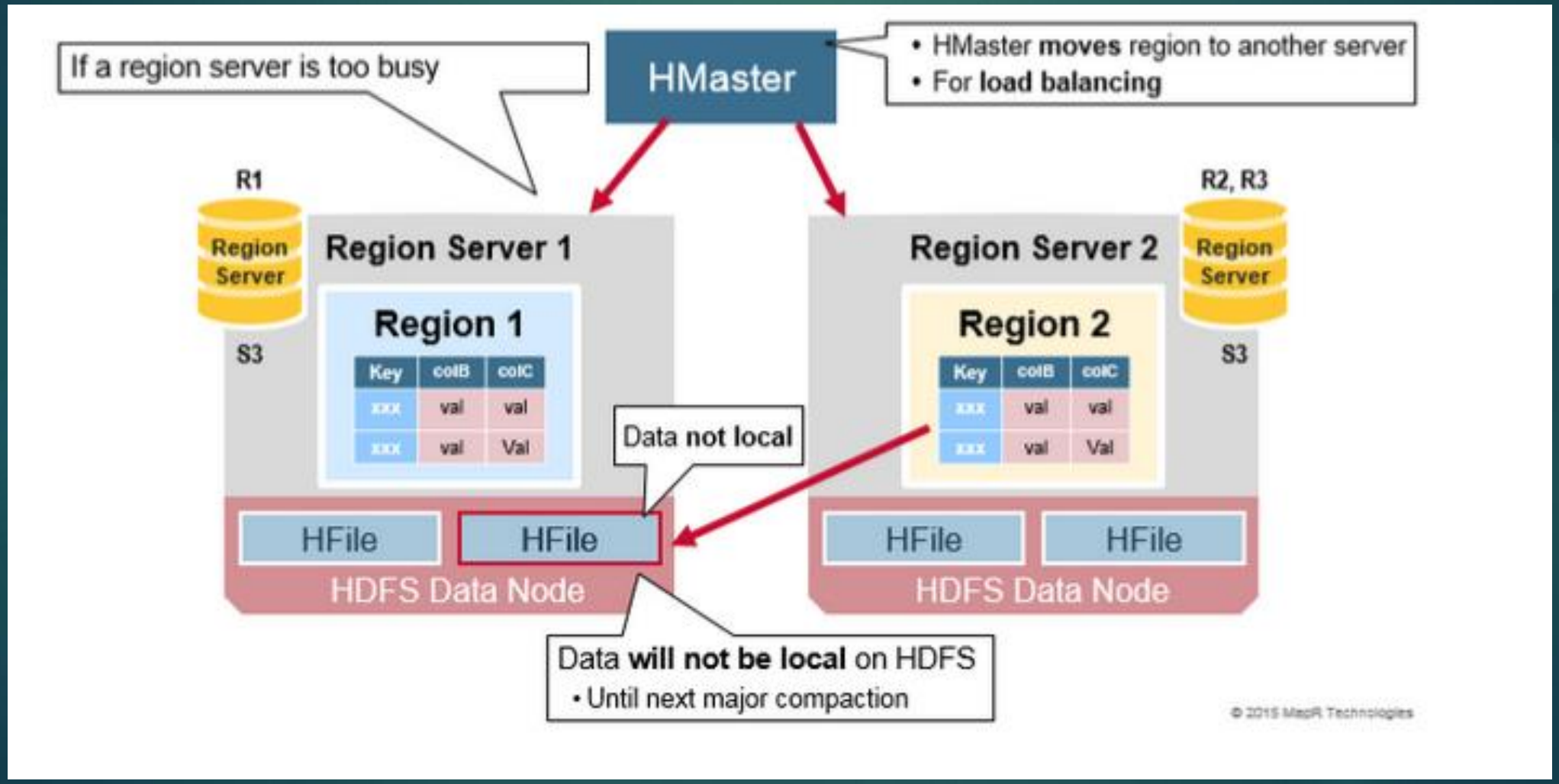


Region Split



when region size > hbase.hregion.max.filesize → split

Load balancing



Hbase shell Commands

Microsoft Word Document

References

- https://mapr.com/blog/in-depth-look-hbase-architecture/
- https://www.guru99.com/hbase-tutorials.html
- https://www.tutorialspoint.com
- Hadoop: the definitive guide 4th edition