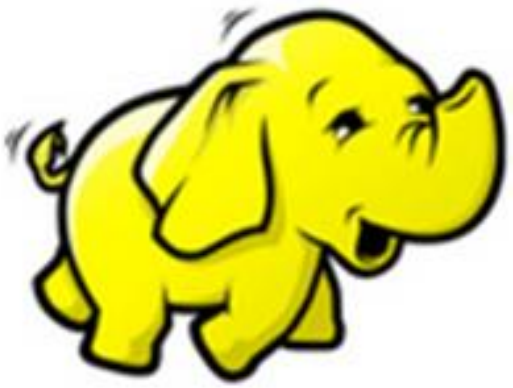

git
APACHE
Spark
IJ
HIVE
HDFS



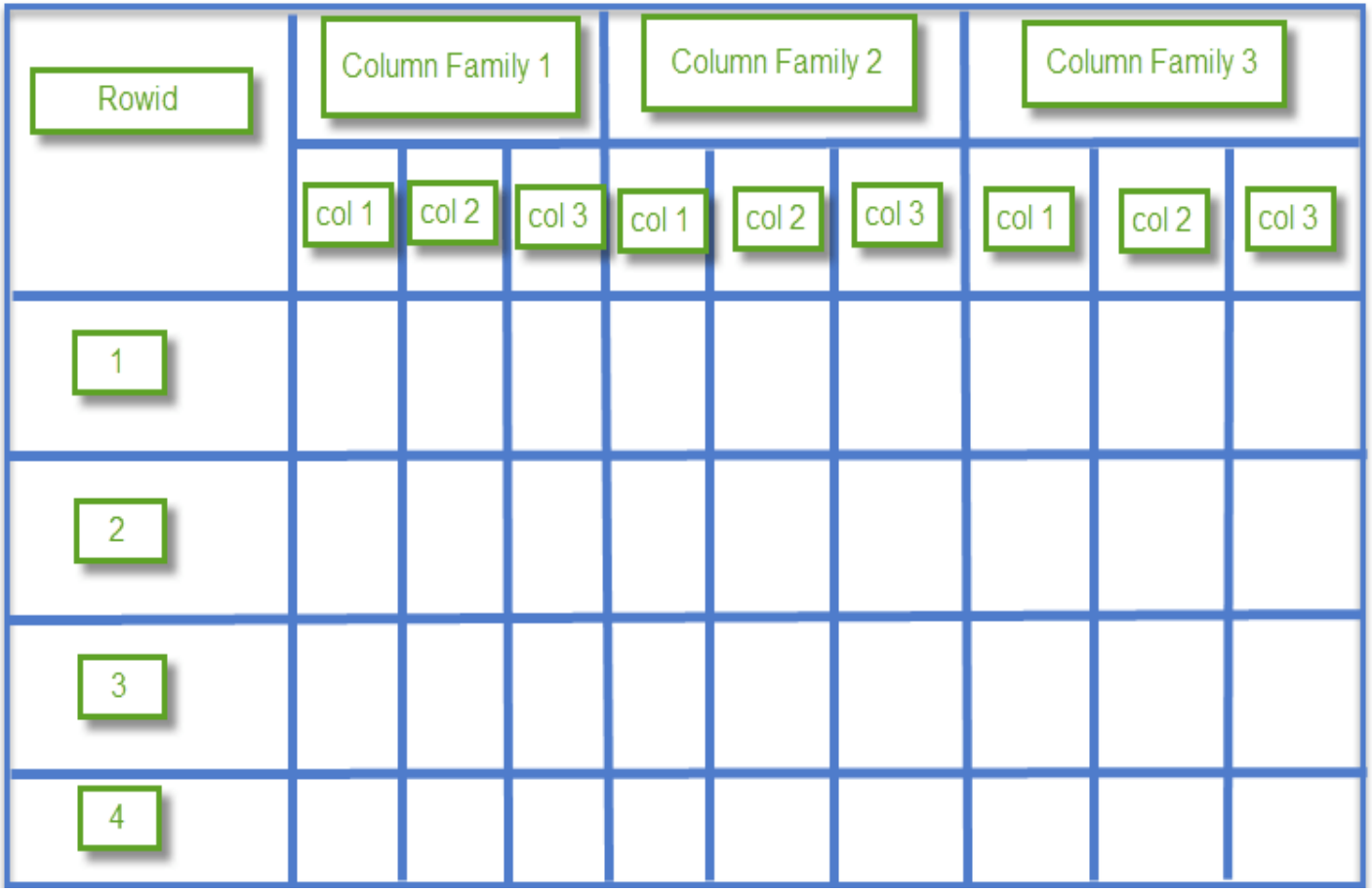

Spark

Are we missing something?

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

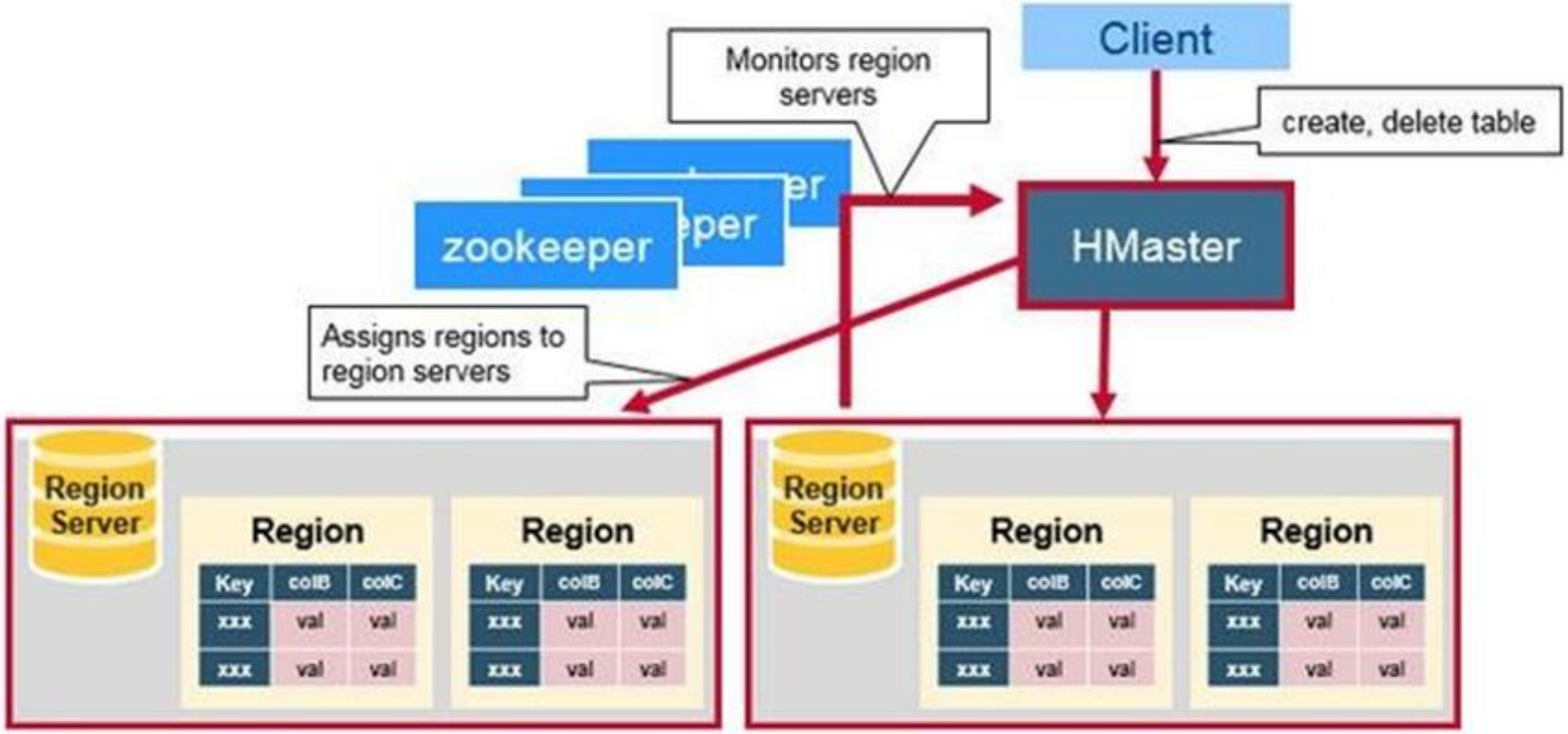
Data structure



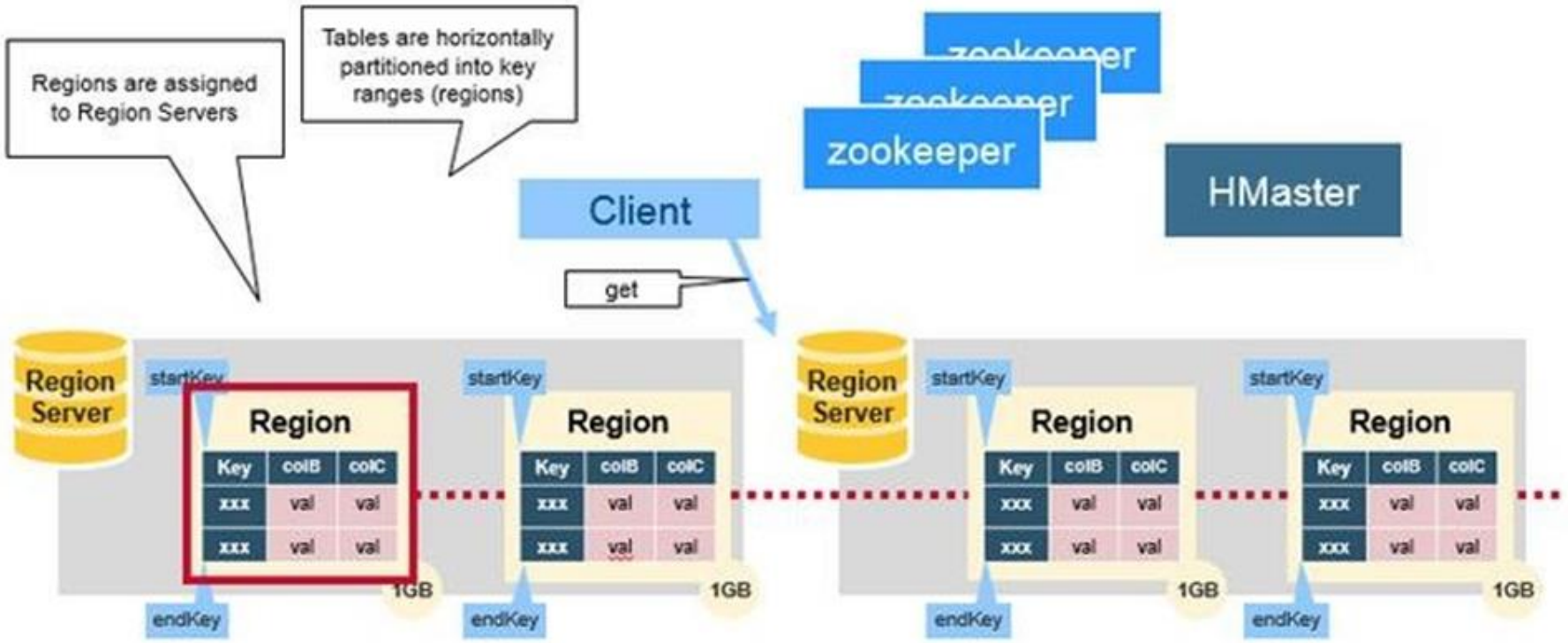
Data structure: cont...

- Namespace: Logical grouping of tables
- Table: Collection of rows present
- Row: Collection of column families
- Column Family: Collection of columns
- Column: Collection of key-value pairs
- Rowid: Unique ID to define a row
- Cell: {rowid, column family, column, version}

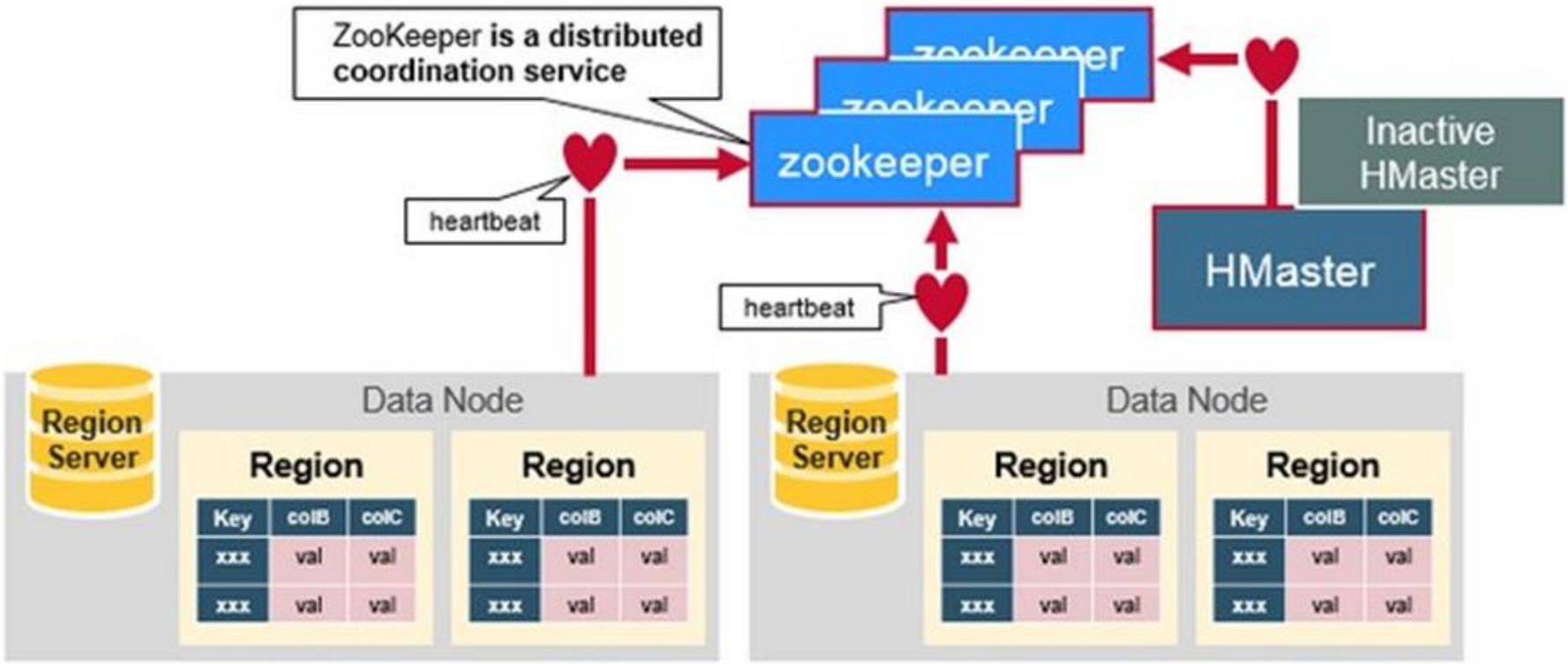
Architecture: HMaster



Architecture: Region Server



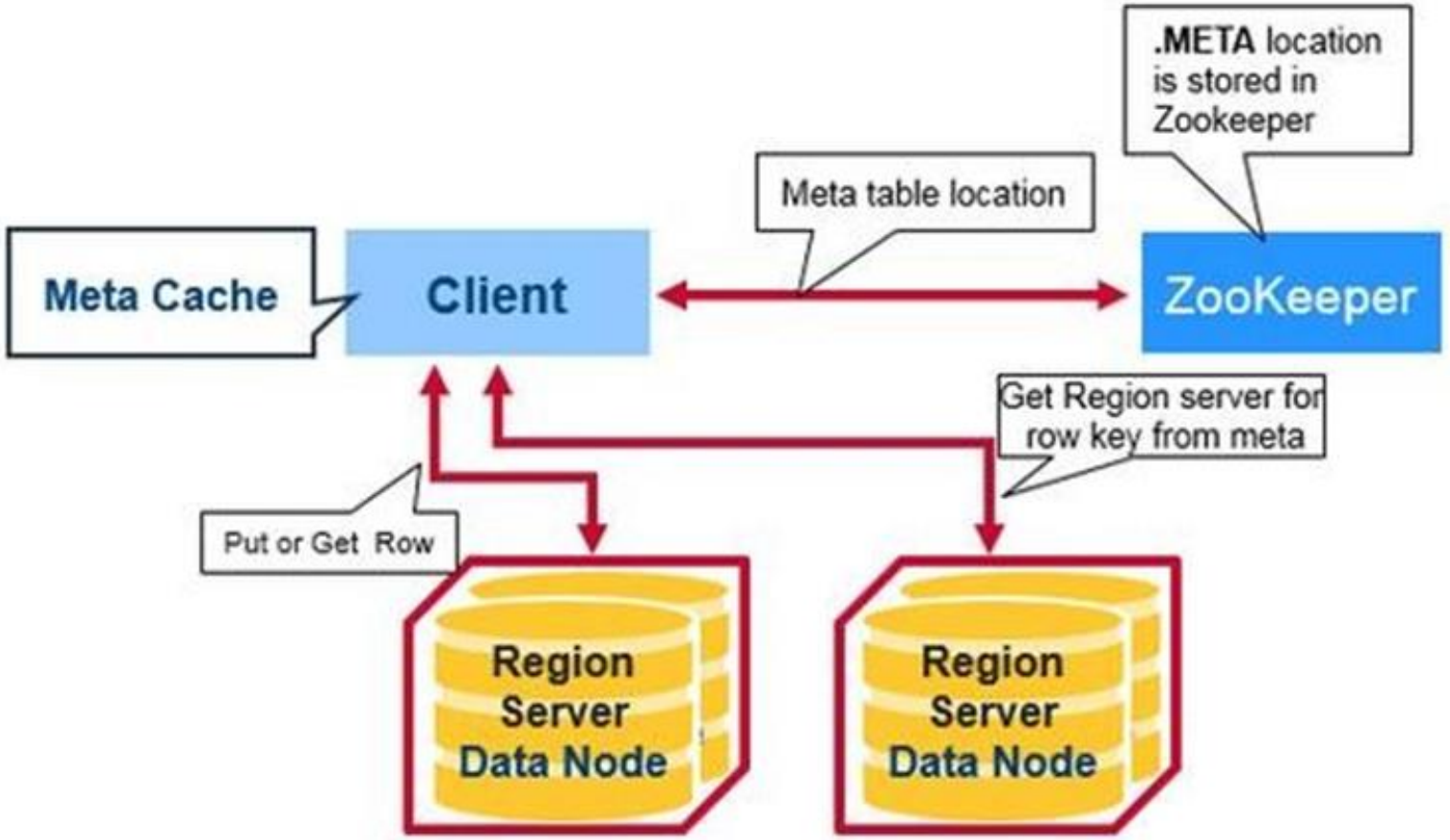
Architecture: Zookeeper



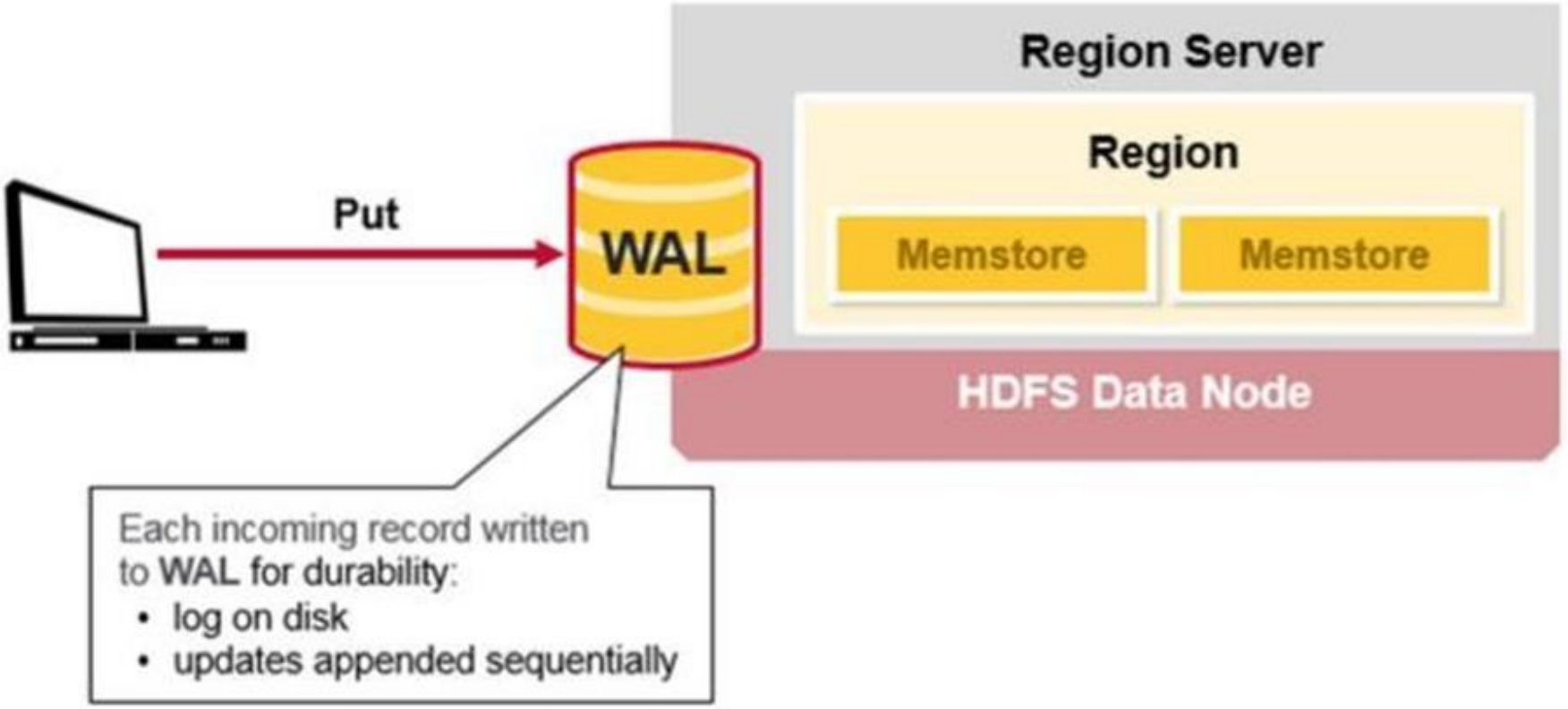
Processing

- 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

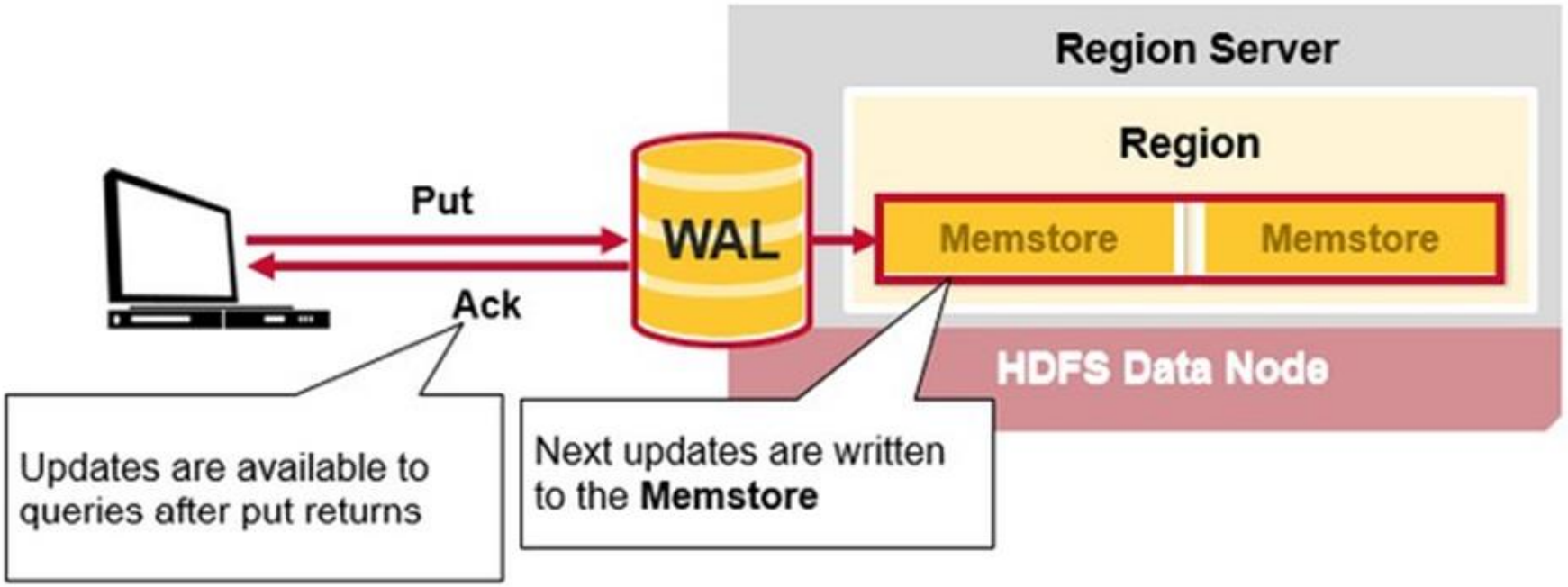
Processing: Locate a row



Processing: Write operation

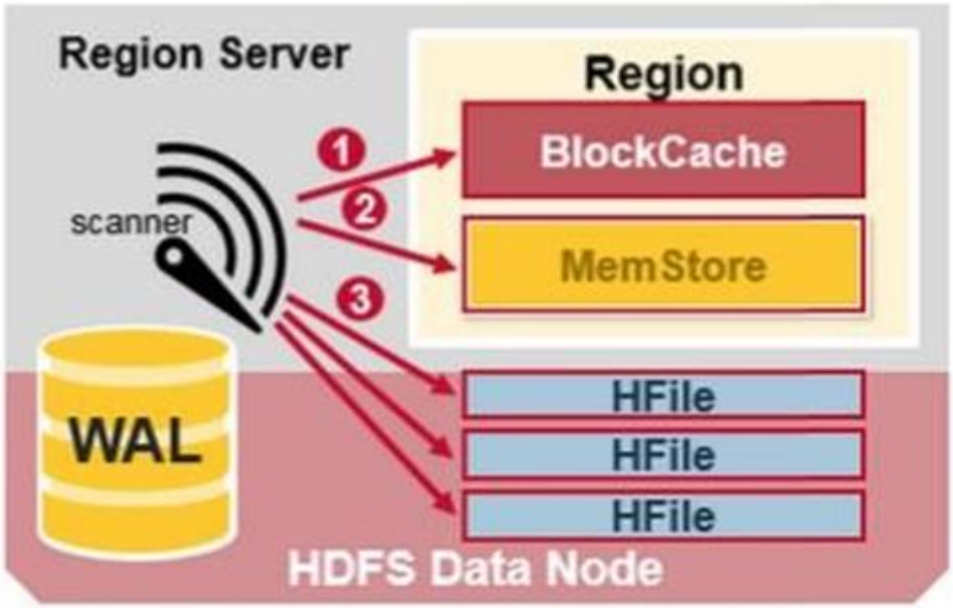


Processing: Write operation

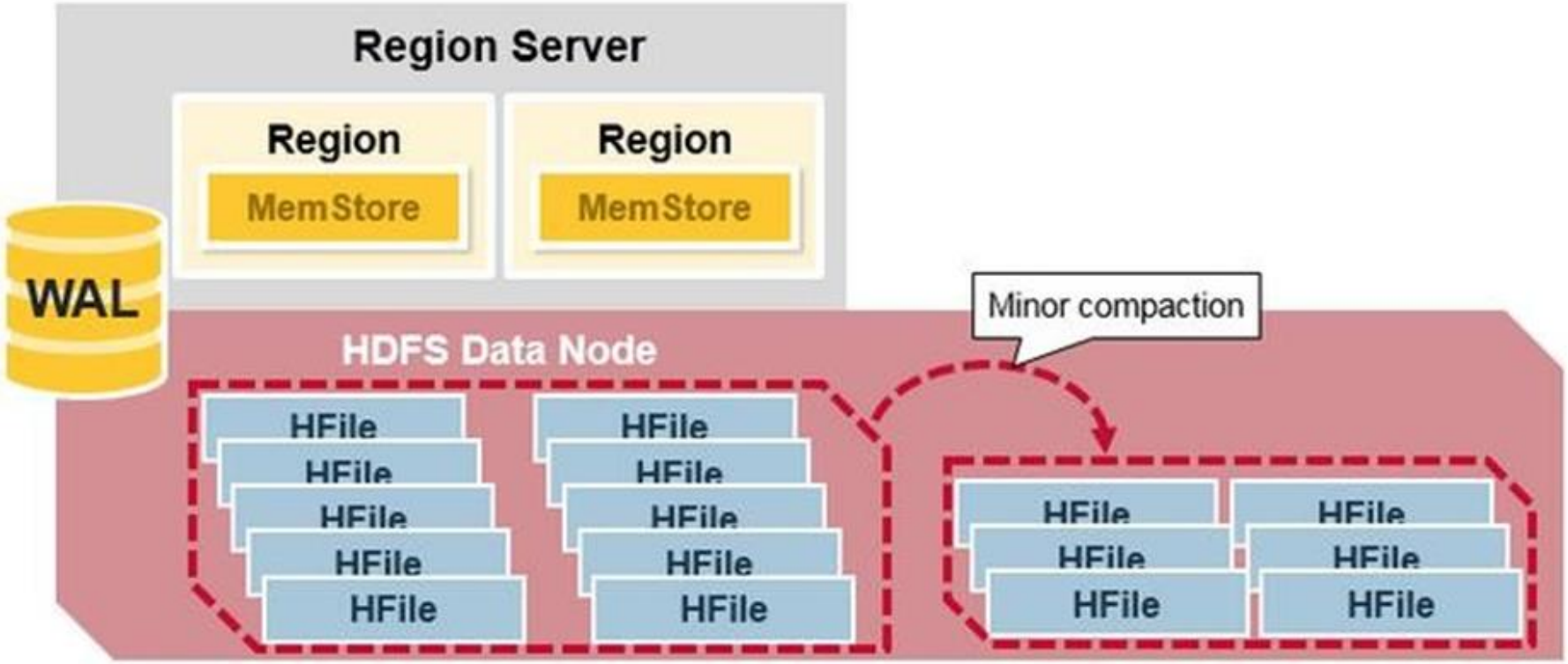


Processing: Read operation

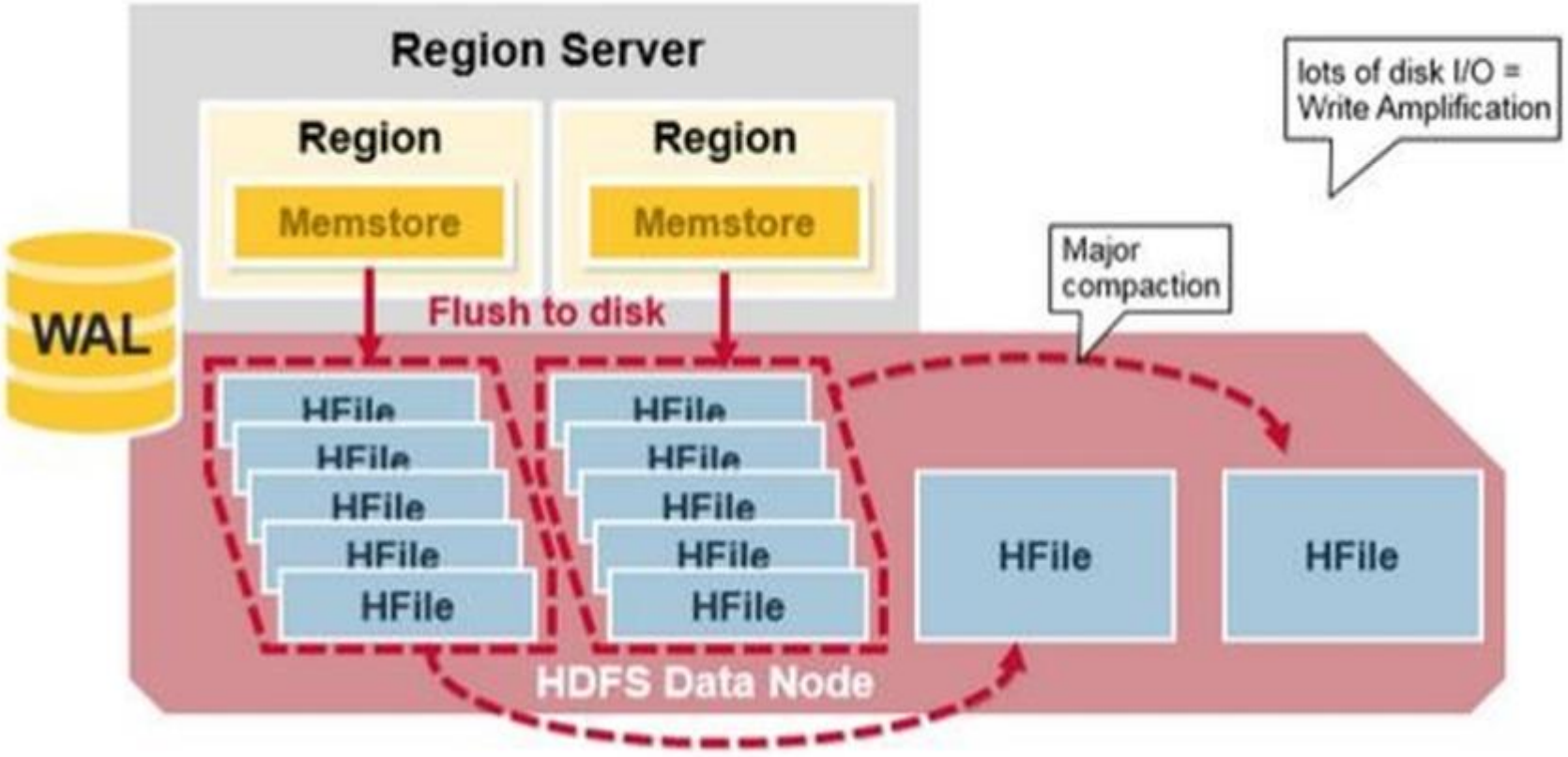
1. First the scanner looks for the Row KeyValues in the Block cache
2. Next the scanner looks in the MemStore
3. If all row cells not in MemStore or blockCache, look in HFiles



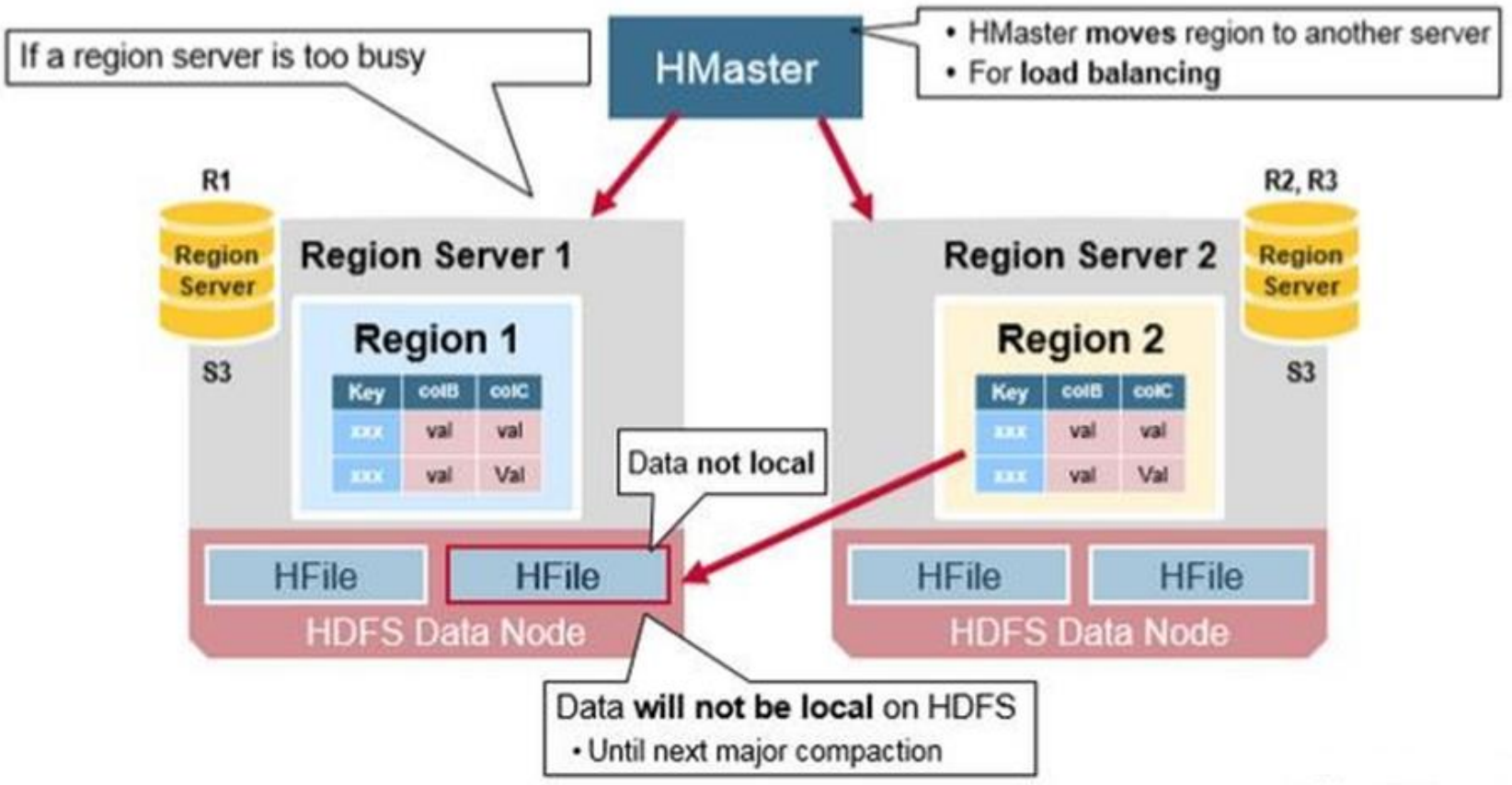
Processing: Compaction



Processing: Compaction

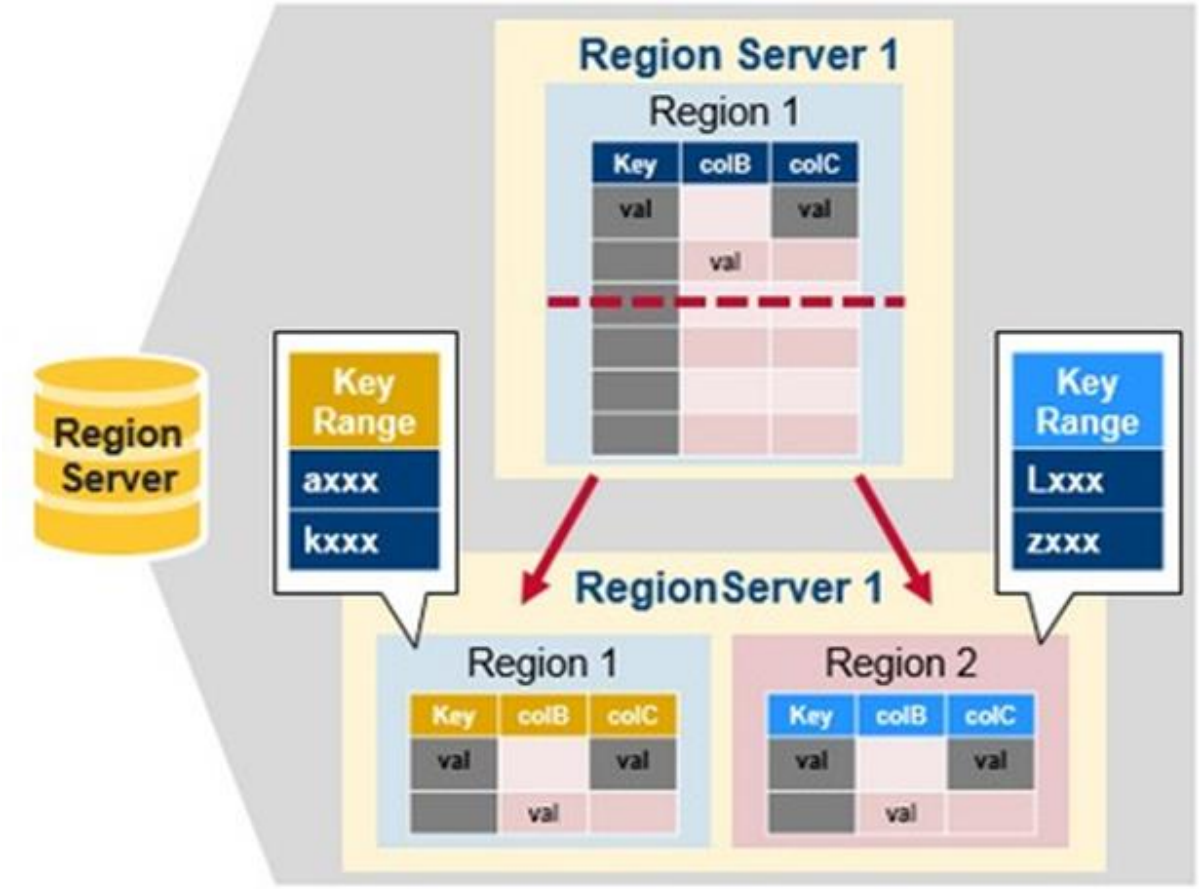


Load balancing



Load balancing: Region Split

- Hbase.hregion.max.filesize



Reference

- 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