Apache HBase

Sunbeam Infotech

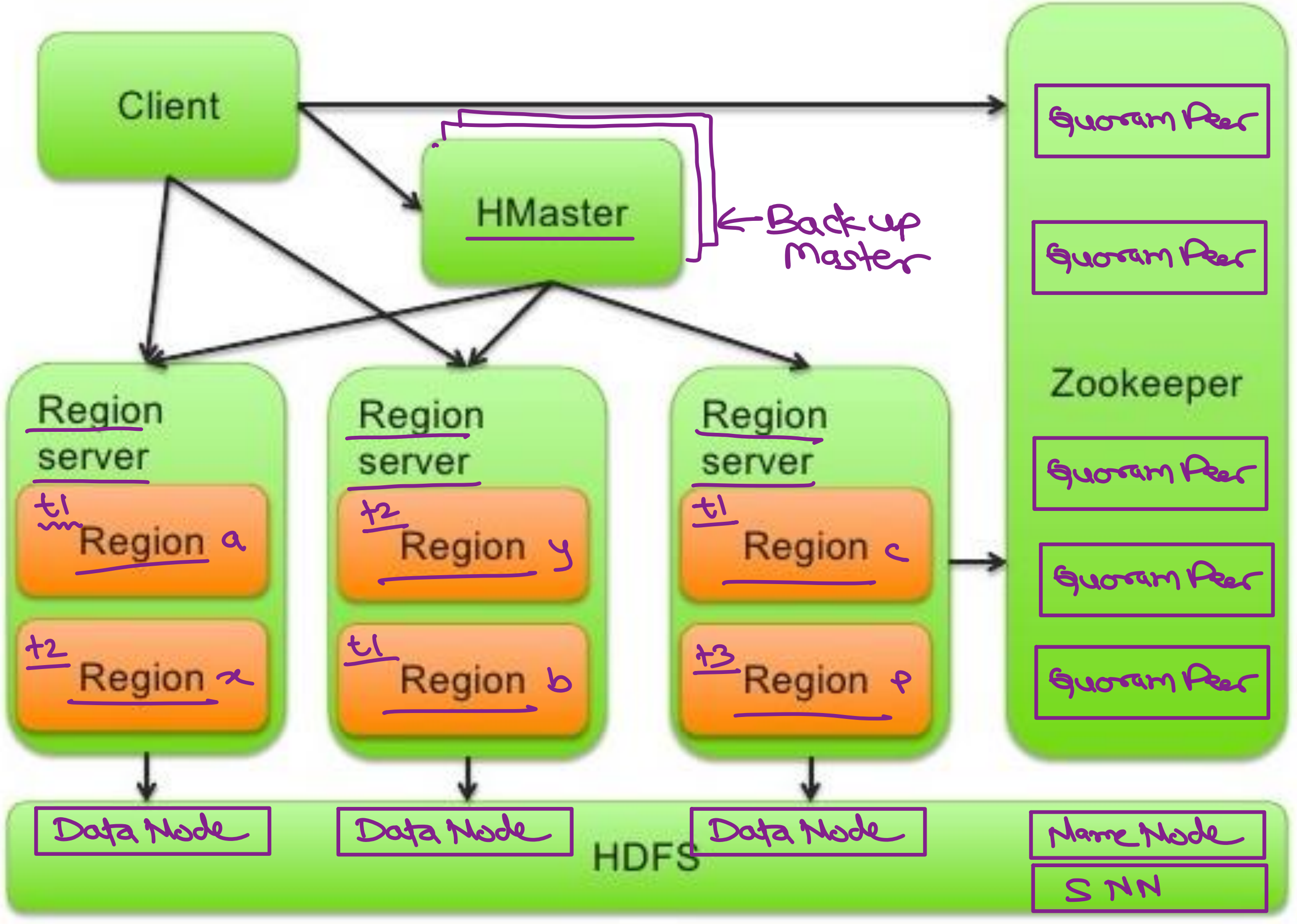
HBase Data Storage

- Different column families can have different properties and access patterns.
- Configurable column properties are cache usage, compression (none, gzip, LZO), version retention policies.
- Each column family is stored in a separate file (called HFile), which is then partitioned into **regions**.

table = people

row key → id

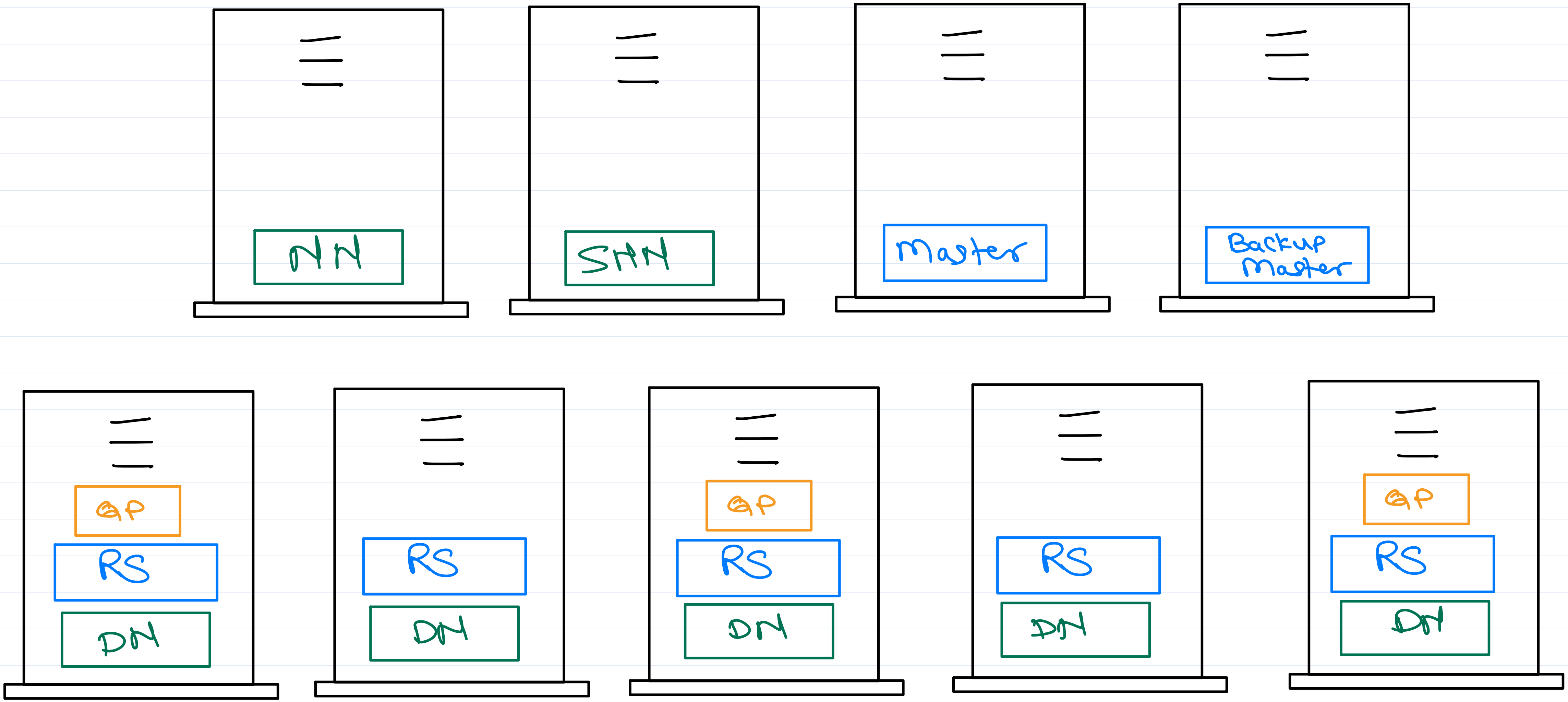
id	name - col family1			contact - col family2					
	title	fname	lname	phone	mobile	email	fax	addr	pager
1	Mr.	Nilesh	x	x	9812... ~~9527...~~	nilesh@—	x	x	
2	Mr.	x	Kulkarni	x	x	amit@—	x	Pune	
3	x	Rahul	x	x	x	x	x	x	123
7	x	james	bond	x	x	james@—.com	x	London	



region ——→ region server 1

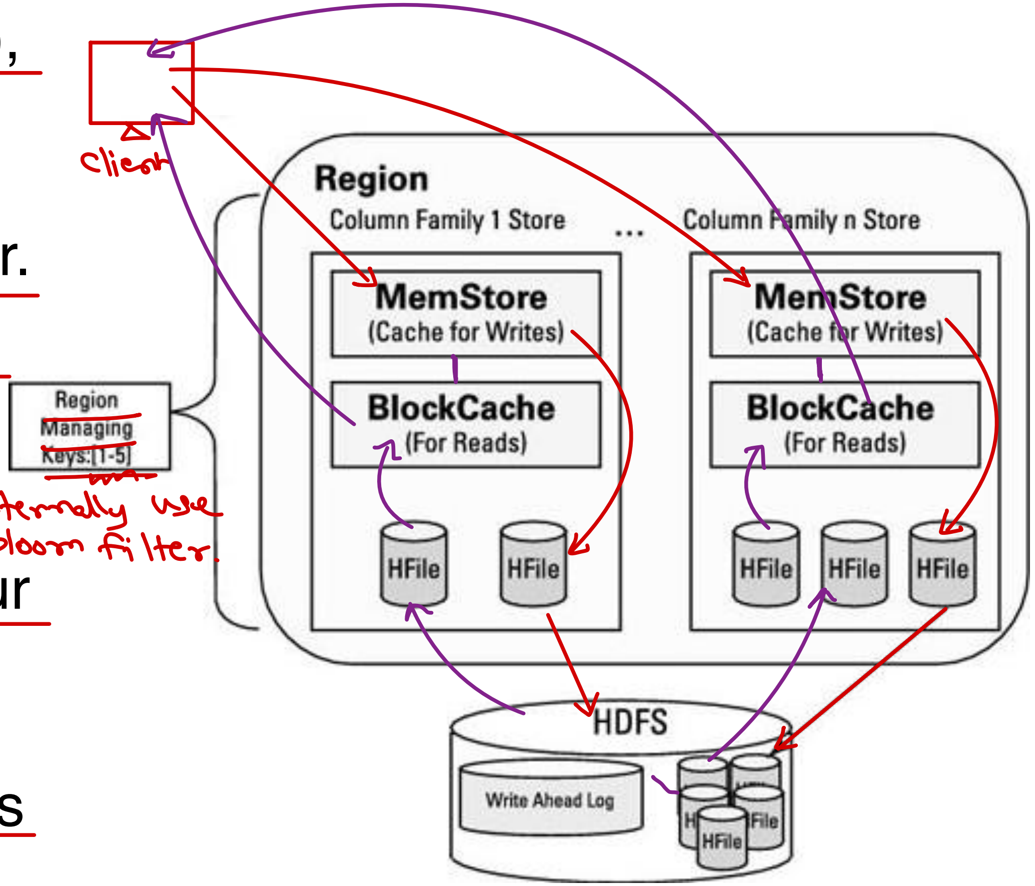
→ region ——→ region server 2

HBase Cluster



HBase RegionServers

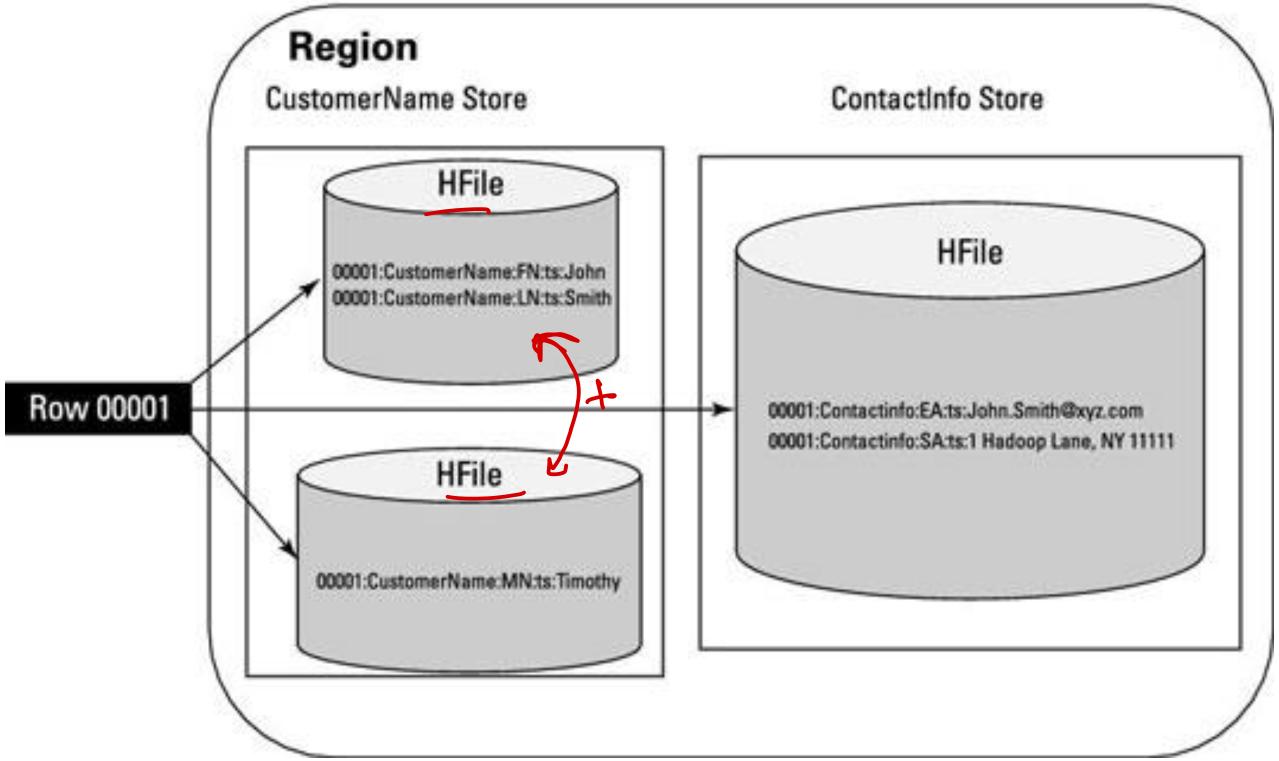
- They are software processes/daemons responsible for store and retrieve data in HBase. Running on each node in cluster. worker
- When a table grows beyond threshold (as per config), it is auto split & distributed the load to another node. This is called auto-sharding.
- Each split is separate region managed by a rgn server.
- Each column family store has a read cache called the BlockCache and a write cache called the MemStore.
- BlockCache helps with random read performance. – internally use bloom filter.
- The Write Ahead Log (WAL, for short) ensures that our Hbase writes are reliable.
- The design of HBase is to flush column family data stored in the MemStore to one HFile per flush. HFile is finally stored in HDFS blocks.



HBase Compactions

- Compaction, the process by which HBase cleans up itself.
- Minor compactions combine (a configurable number of smaller HFiles into one larger HFile. Due to combining data together, disk access speed increases.
- Major compaction combine all HFiles into one large HFile. It also removes extra versions & deleted cells as per config.

HFiles and Minor Compaction


Region
CustomerName Store
ContactInfo Store
HFile
00001:CustomerName:FN:ts:John
00001:CustomerName:LN:ts:Smith
Row 00001
+
HFile
00001:CustomerName:MN:ts:Timothy
HFile
00001:Contactinfo:EA:ts:John.Smith@xyz.com
00001:Contactinfo:SA:ts:1 Hadoop Lane, NY 11111

HBase Master

- Monitor region servers in cluster.
- Handle metadata operations.
- Assign regions (after split) & balance load.
- Manage region server failover.
- Manage and clean catalog tables.
- Clear the WAL (Write Ahead Logs).
- Usually a back copy of master server is maintained to handle failover of master.

HBase 2.x multiple backup master are allowed (max: 9)

HBase ZooKeeper

- ZooKeeper is a distributed cluster that provides reliable coordination & synchronization services for clustered applications. Peers
- This team consists of ZooKeeper Leader and ZooKeeper Follower(s).
- HBase comes with an instance of ZK that coordinates operations of master server, region server(s) and client(s).

maintains the metadata

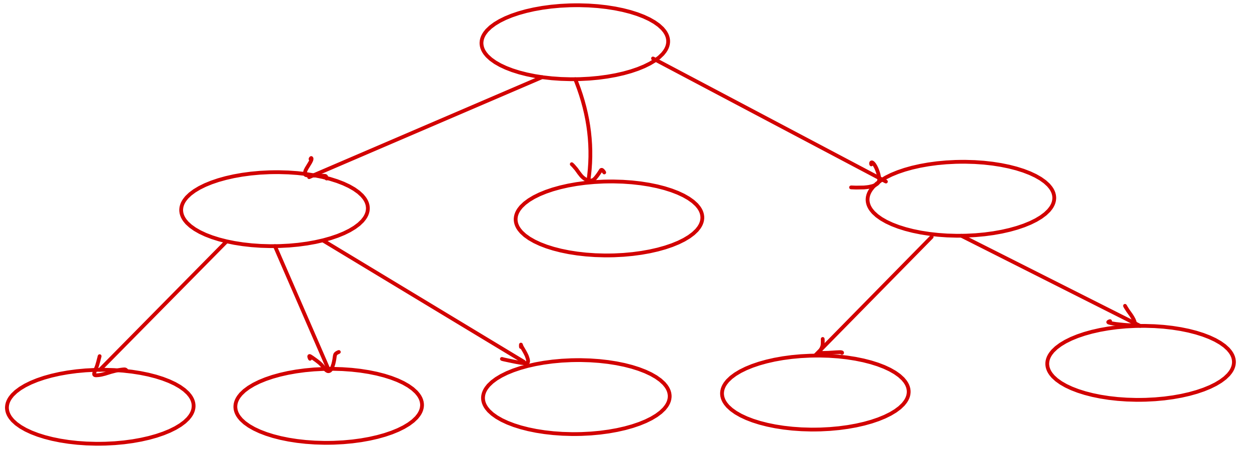
master

region servers ?

tables ?

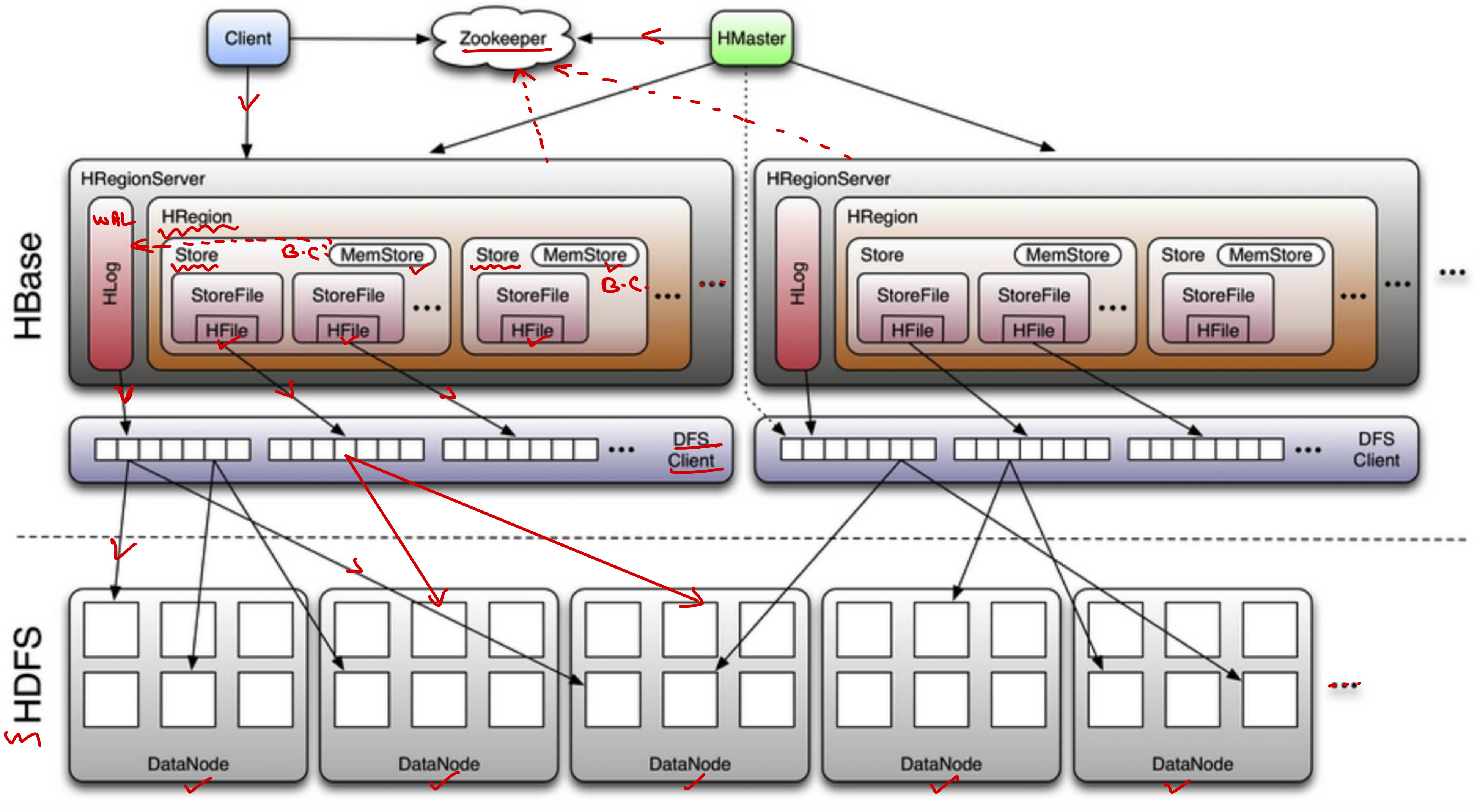
regions ?

ZK keeps all info in hierachial manner.



HBase on HDFS

ACID tx ← RDBMS → Codd's rules
BASE tx ← NoSQL → CAP theorm



HBase: CAP and ACID properties

- C : Consistency ✓
  - HBase is consistent as same data is visible from any node (because of HDFS). ✓
- P : Partition Tolerance ✓
  - HBase it tolerant i.e. if any node goes down the data can be still accessible (because of Replica).
- A : Available ?
  - HBase doesn't guarantee 100% uptime i.e. each request may not get response (success/failure).

- HBase is not ACID compliant like RDBMS as it doesn't support Isolation.
- A : Atomic - HBase guarantees update single record atomically
- C : Consistent - HBase is in consistent stage - No PK/FK relations.
- D : Durable - HBase changes are durable in HDFS.
- I : Isolation - HBase doesn't guarantee changes sequentially.

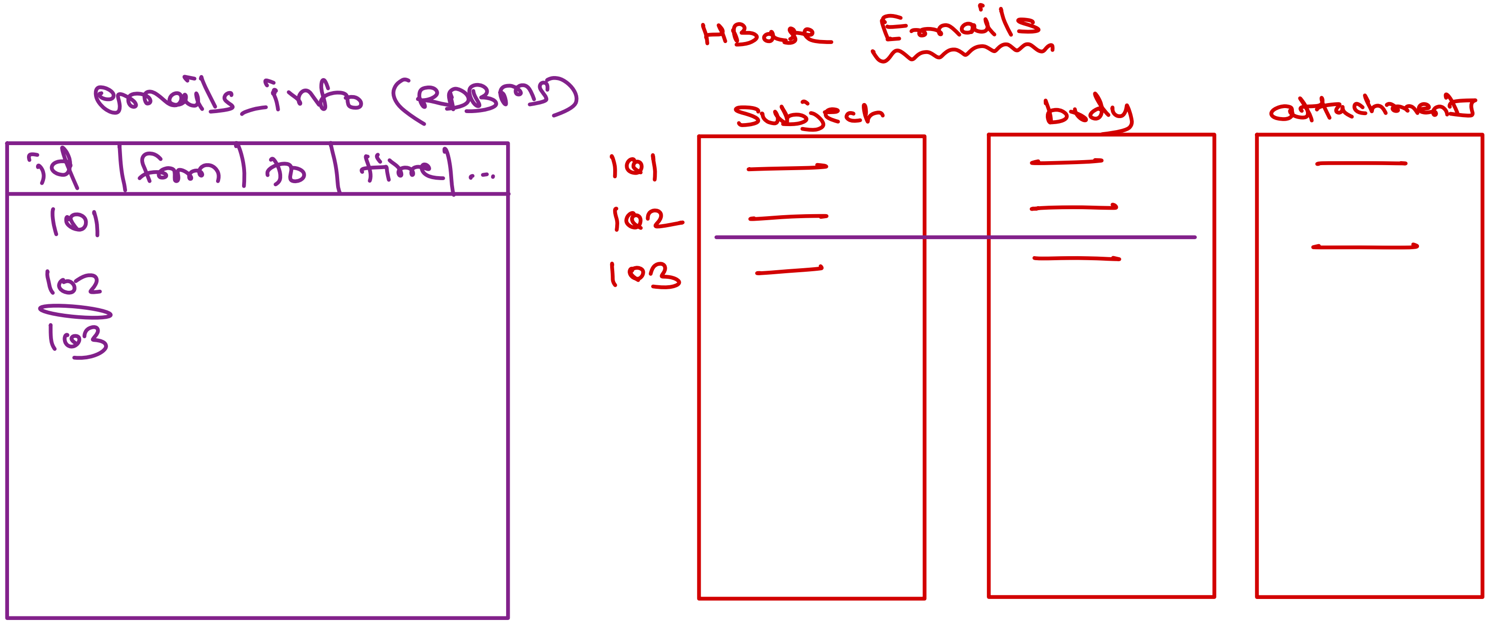
HBase Access

- HBase shell - ruby shell with set of commands
- Web interface - browser on port ~~60010~~ 16010
- Java API - native APIs for HBase
- REST (HTTP) - REST API on port 8080
  - hbase rest start -p <port>
- Thrift - enable access from other language
- Hive/Pig for Analytics or MR

- HBase Shell Commands:
  - https://learnhbase.wordpress.com/2013/03/02/hbase-shell-commands/ ✓

HBase Applications

- HBase is preferred for random reads, random writes or both.
- Do not use HBase if random access is not required.
- HBase can perform thousands of operations per seconds on TBs of data.
- HBase performs well is access pattern is well known & simple.
- Used by: Facebook, Mozilla, Twitter, OpenLogic, Meetup, …



SUNBEAM

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>