NoSQL存储介绍

陈群 DBA/NoSQL

NoSQL存储

键值数据库	Memcache、Redis、Riak	Key-Value
列簇数据库	HBase、Cassandra、Bigtable	<Key, CF, Column>
文档数据库	MongoDB、CouchBase	JSON、XML
图形数据库	Neo4J、Giraph	<vertex,edge/ properties>

存储选型

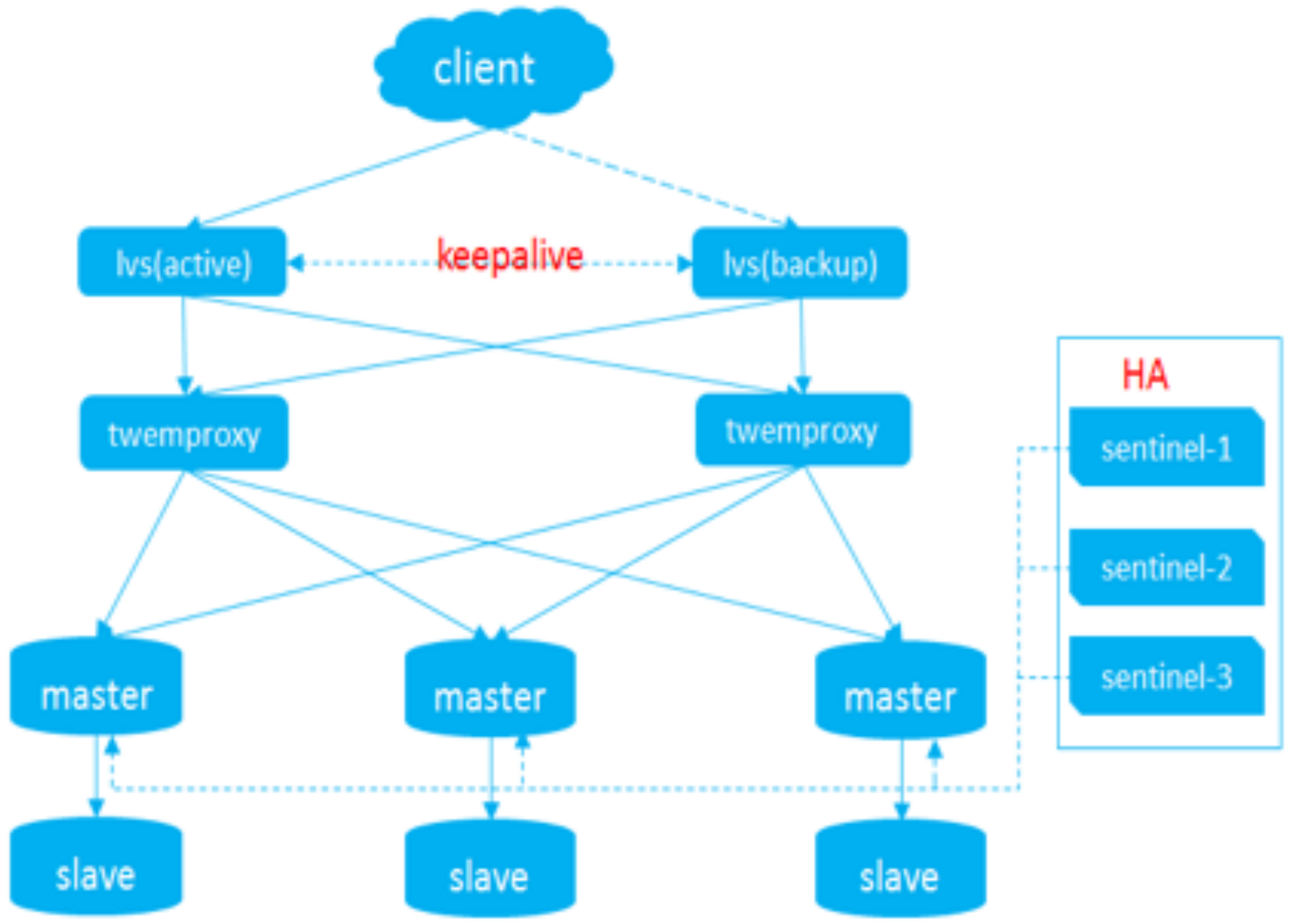
- SLA，响应时间
- Transaction，事务支持能力
- Usage，查询模式（range/like/conditions）
- HA，高可用
- Scalability，水平扩展能力
- Capacity，QPS and Volume
- TTL，数据生命周期

redis

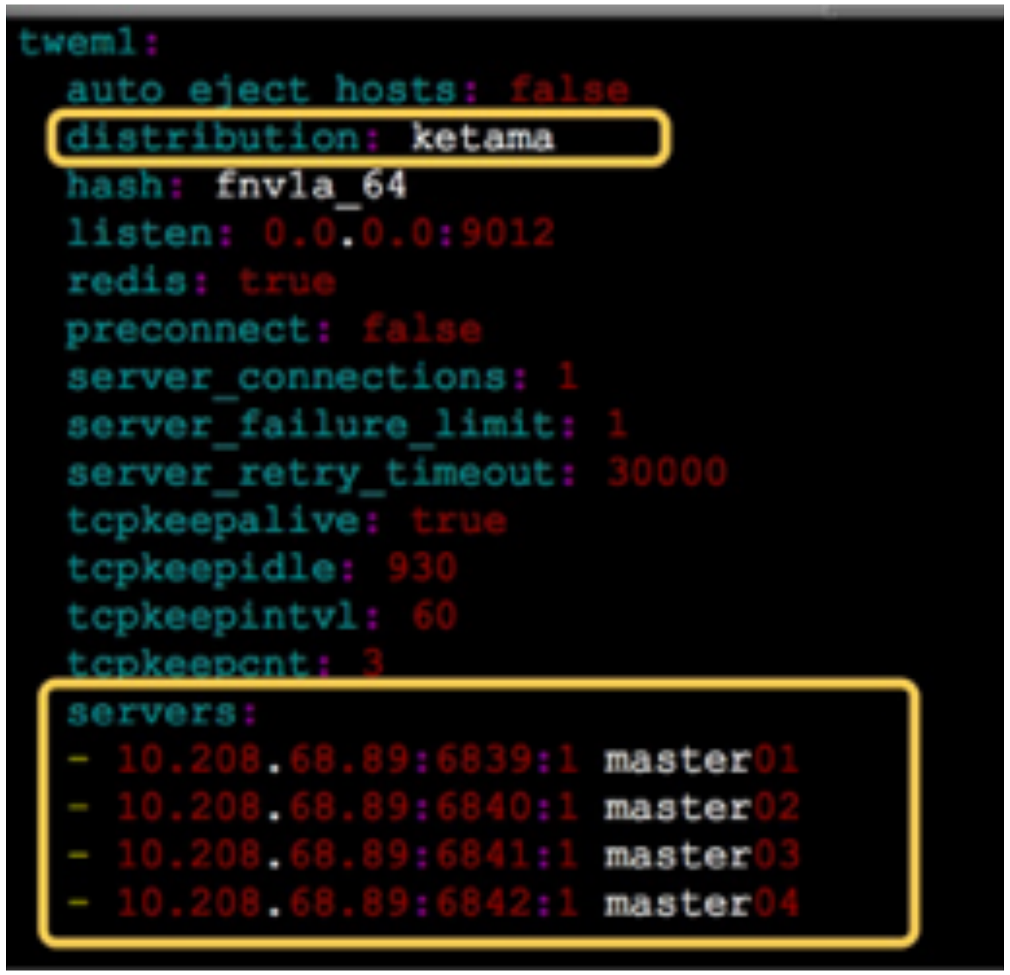
Redis集群架构

- Client Sharding
- Twemproxy
- Redis Cluster
- Codis
- Tair

Twemproxy


client
lvs(active)
keepalive
lvs(backup)
twemproxy
twemproxy
HA
sentinel-1
sentinel-2
sentinel-3
master
master
master
slave
slave
slave

Twemproxy



```
tweml:
  auto_eject_hosts: false
  distribution: ketama
  hash: fnv1a_64
  listen: 0.0.0.0:9012
  redis: true
  preconnect: false
  server_connections: 1
  server_failure_limit: 1
  server_retry_timeout: 30000
  tcpkeepalive: true
  tcpkeepidle: 930
  tcpkeepintvl: 60
  tcpkeepcnt: 3
  servers:
  - 10.208.68.89:6839:1 master01
  - 10.208.68.89:6840:1 master02
  - 10.208.68.89:6841:1 master03
  - 10.208.68.89:6842:1 master04
```

优点：

1）分片逻辑对开发透明

2）可以快速从redis迁移到集群模式

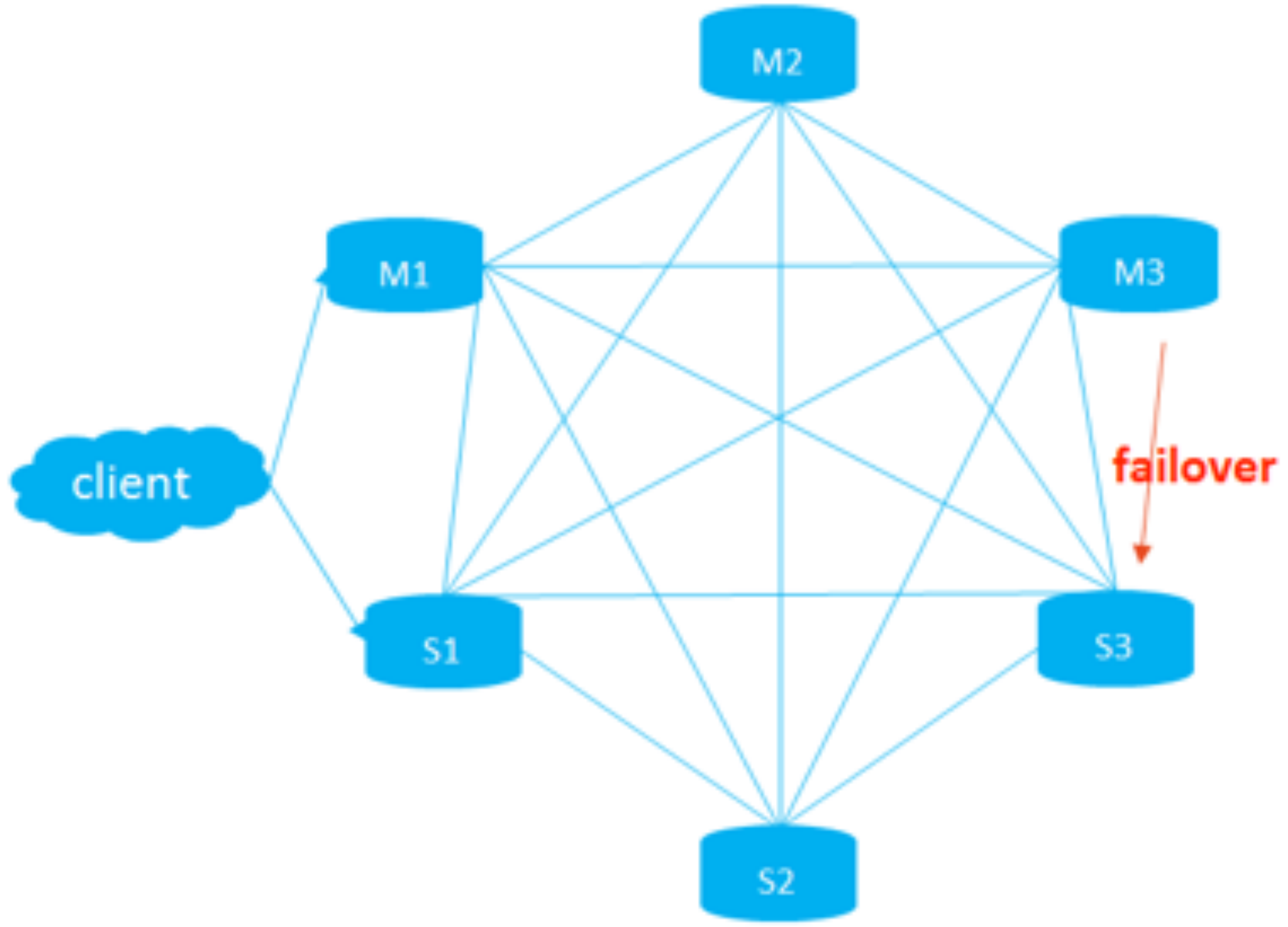
缺点：

1）架构复杂，层次多，构建成本也高

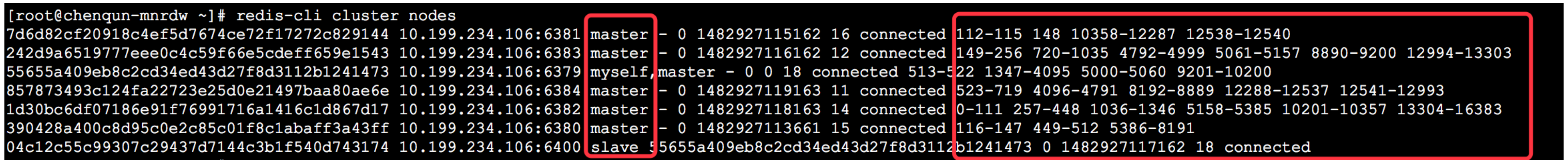
2）redis层难以在线扩容

3）需要自己开发HA模块

Redis Cluster



Redis Cluster



```
[root@chenqun-mnrdw ~]# redis-cli cluster nodes
7d6d82cf20918c4ef5d7674ce72f17272c829144 10.199.234.106:6381 master - 0 1482927115162 16 connected 112-115 148 10358-12287 12538-12540
242d9a6519777eee0c4c59f66e5cdeff659e1543 10.199.234.106:6383 master - 0 1482927116162 12 connected 149-256 720-1035 4792-4999 5061-5157 8890-9200 12994-13303
55655a409eb8c2cd34ed43d27f8d3112b1241473 10.199.234.106:6379 myself,master - 0 0 18 connected 513-522 1347-4095 5000-5060 9201-10200
857873493c124fa22723e25d0e21497baa80ae6e 10.199.234.106:6384 master - 0 1482927119163 11 connected 523-719 4096-4791 8192-8889 12288-12537 12541-12993
1d30bc6df07186e91f76991716a1416c1d867d17 10.199.234.106:6382 master - 0 1482927118163 14 connected 0-111 257-448 1036-1346 5158-5385 10201-10357 13304-16383
390428a400c8d95c0e2c85c01f8c1abaff3a43ff 10.199.234.106:6380 master - 0 1482927113661 15 connected 116-147 449-512 5386-8191
04c12c55c99307c29437d7144c3b1f540d743174 10.199.234.106:6400 slave 55655a409eb8c2cd34ed43d27f8d3112b1241473 0 1482927117162 18 connected
```

Redis Cluster

Features：

1）无中心架构

2）每个节点存储一部分数据，分布在16384个slots中

3）HASH_SLOT = CRC16(key) mod 16384

4）复制从库，standby

5）gossip协议，对异常节点检查并投票

6）auto failover和manual switch-over，可以提升slave为新的master

7）支持在线扩容，在线迁移slots和数据

缺点：

1）数据迁移速度较慢，v3.0.6开始migrate单次迁移多个key

2）仅支持redis部分命令

3）应用的稳定性依赖于smart client的实现

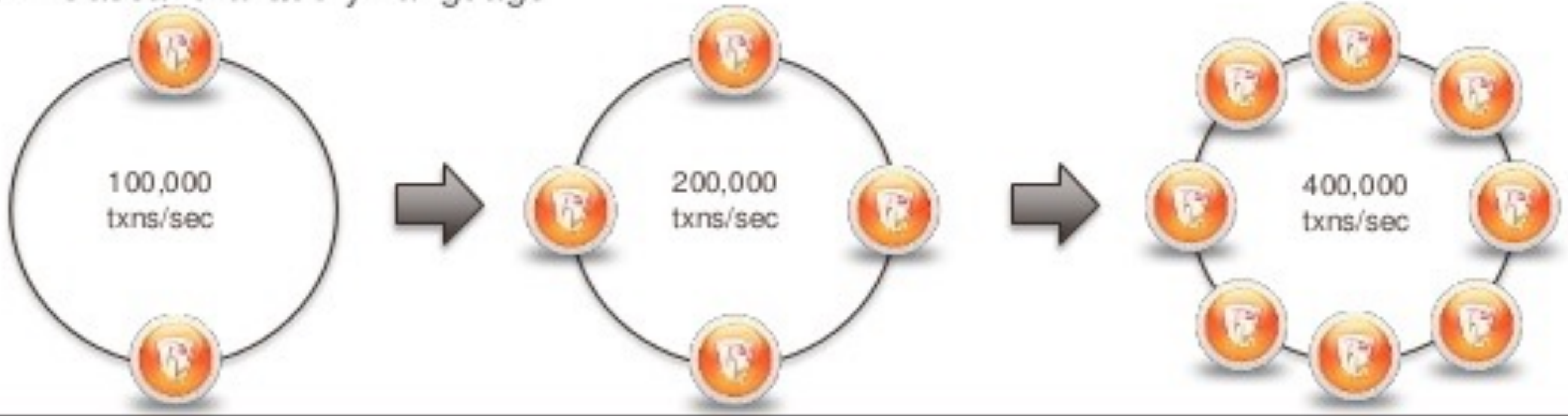
架构对比

	Client Sharding	Twemproxy	Redis Cluster
架构	• 多个独立的实例 • 主从复制 • 客户端实现分片 • 增加开发复杂度	• Twemroxy层实现数据分片逻辑 • Twemproxy仅转发请求和响应 • Twemproxy上加负载均衡，简化开发 • 缺点：架构复杂，层次多 • 优点：分片对开发透明	• 无中心架构 • Server端实现分片(crc32) • 主从复制 • 强依赖Smart Client
兼容性	• 依赖具体客户端实现	• 兼容大部分redis命令 • 支持mget/mset • 支持pipeline	• 依赖具体客户端实现
成本	• 成本低	• 机器数量多，硬件成本高 • 管理和维护成本高	• 低
扩展性	• 扩展性差 • 修改配置和代码，发布变更 • 简单扩容方式：倍增节点	• Redis层扩容能力弱	• 可以在线增加/缩容节点 • 弱点，迁移数据速度较慢
HA	• 需要自己开发HA模块 • Sentinel＋Zookeeper/DNS	• 需要自己开发HA模块 • Sentinel＋Zookeeper/DNS	• slave 热备 • 支持auto-failover/switchover

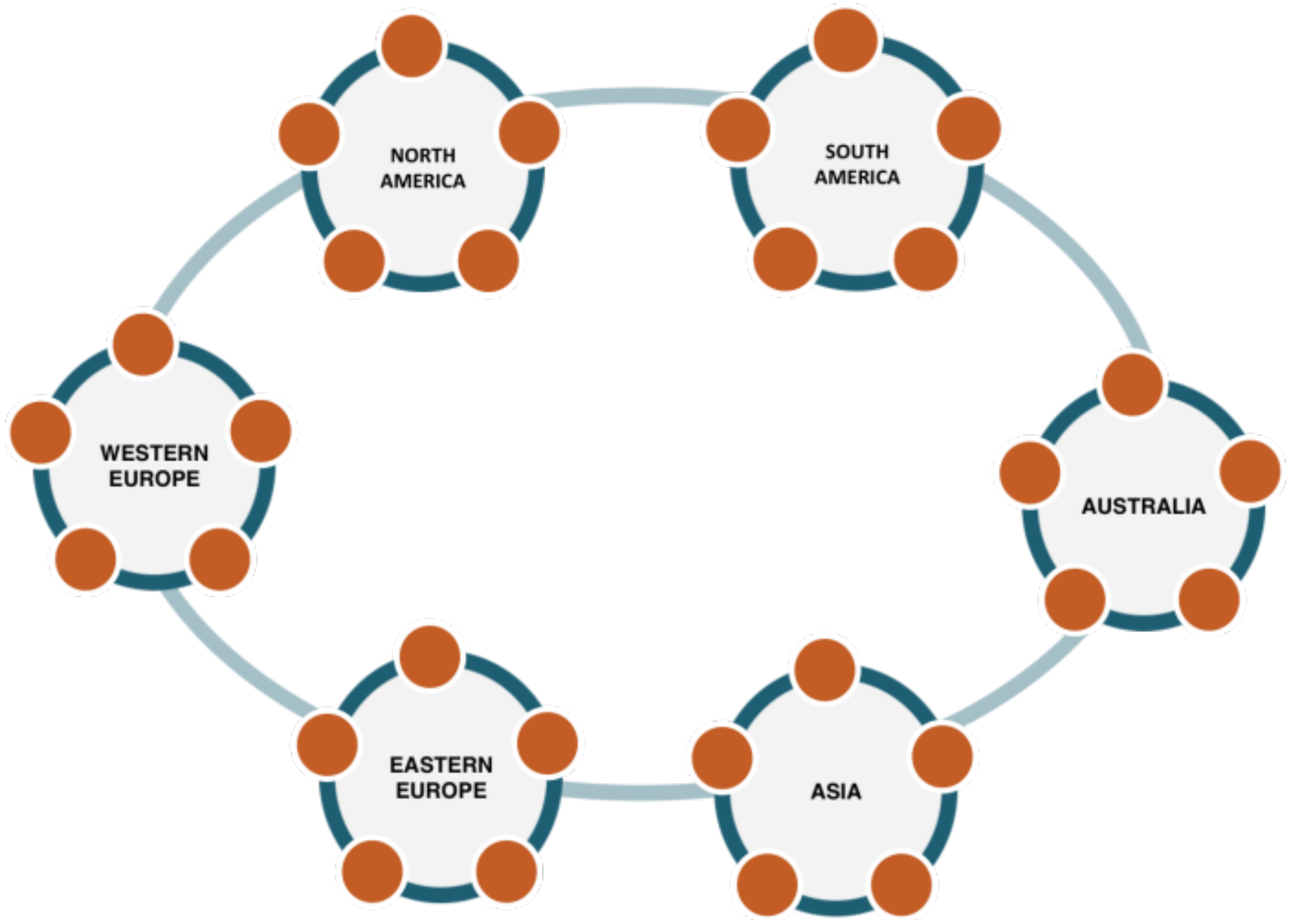
cassandra

What is Apache Cassandra

- Masterless Architecture with read/write anywhere design
- Continuous Availability with no single point of failure
- Multi-Data Center and Zone support
- Flexible data model for unstructured, semi-structured and structured data
- Linear scalable performance with online expansion (scale-out and scale-up)
- Security with integrated authentication
- Operationally simple
- CQL - Cassandra Query Language

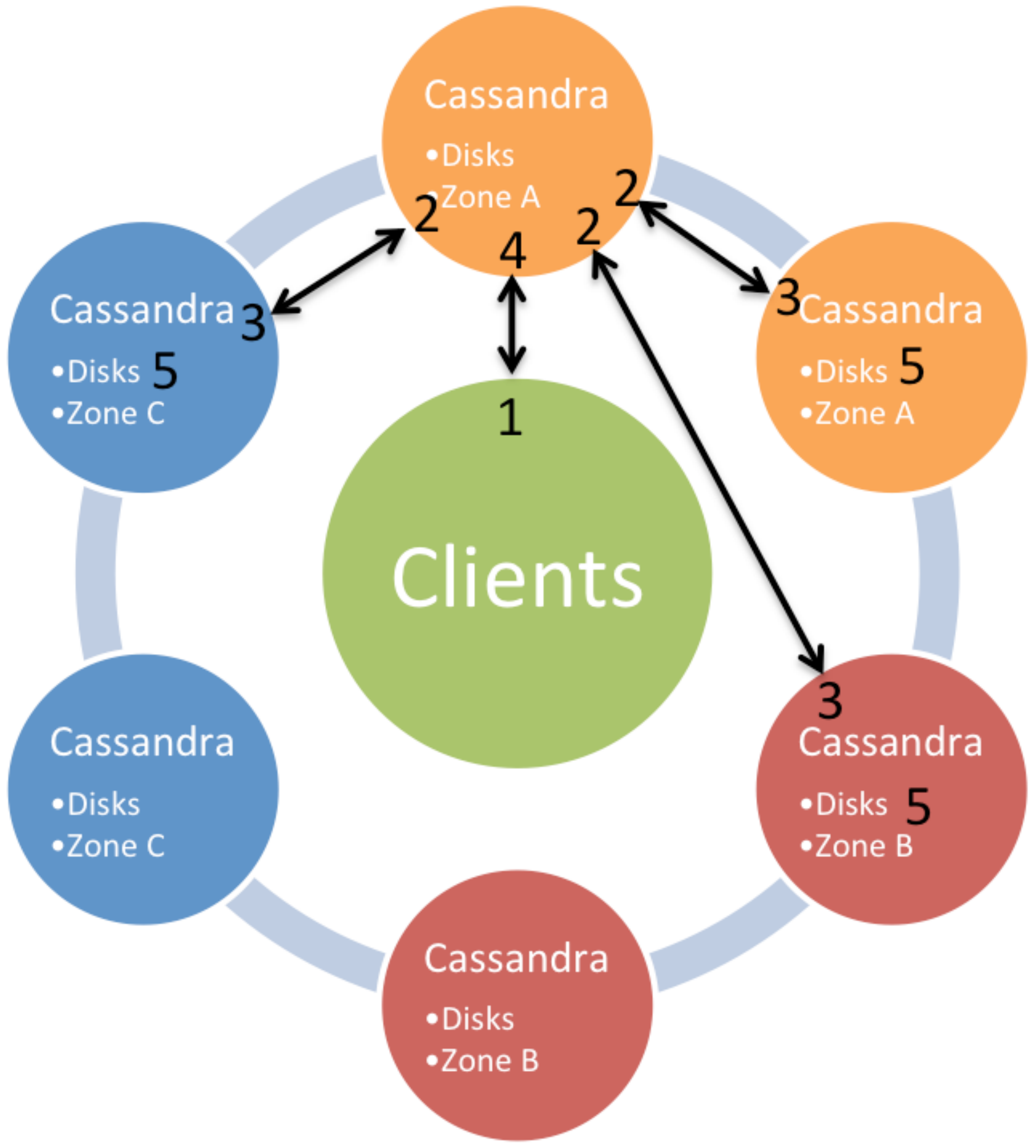


Multi Datacenter



Cassandra Write Data Flows

Single Region, Multiple Availability Zone



1. Client Writes to any Cassandra Node
2. Coordinator Node replicates to nodes and Zones
3. Nodes return ack to coordinator
4. Coordinator returns ack to client
5. Data written to internal commit log disk

If a node goes offline, hinted handoff completes the write when the node comes back up.

Requests can choose to wait for one node, a quorum, or all nodes to ack the write

SSTable disk writes and compactions occur asynchronously

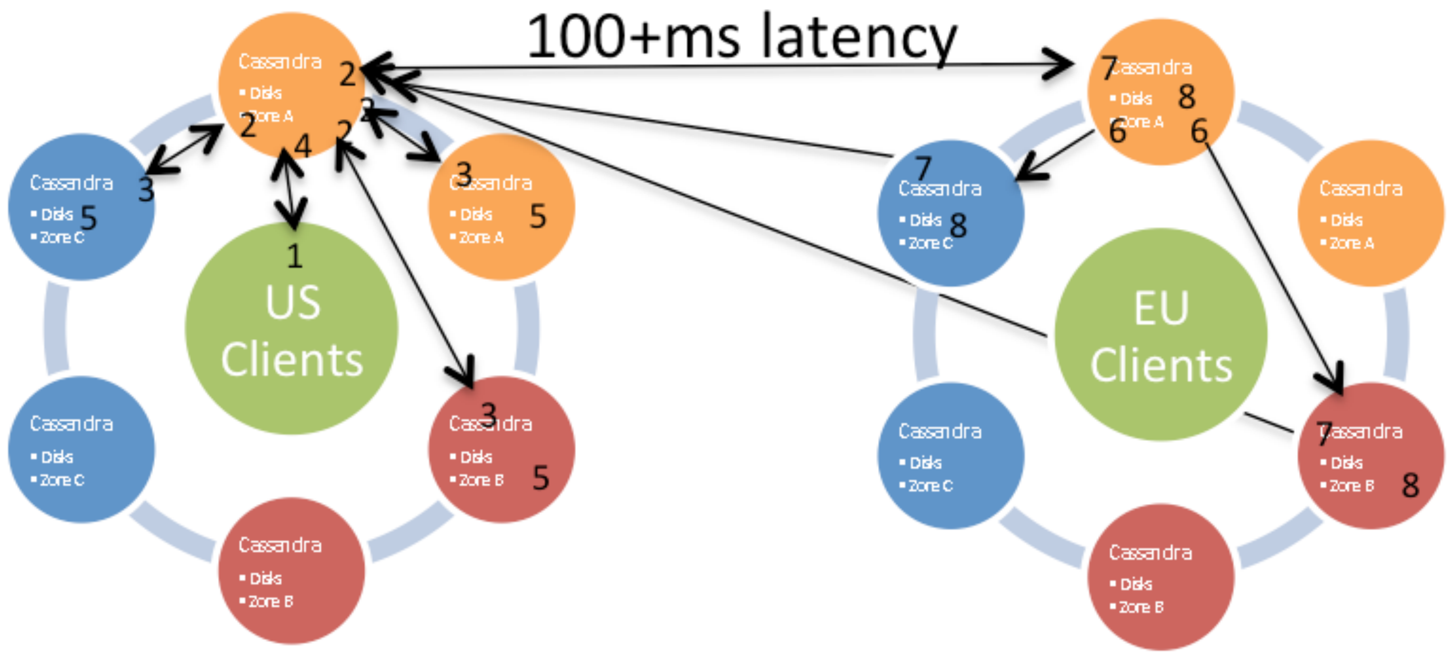
NETFLIX

Data Flows for Multi-Region Writes

Consistency Level = Local Quorum

1. Client Writes to any Cassandra Node
2. Coordinator node replicates to other nodes Zones and regions
3. Local write acks returned to coordinator
4. Client gets ack when 2 of 3 local nodes are committed
5. Data written to internal commit log disks
6. When data arrives, remote node replicates data
7. Ack direct to source region coordinator
8. Remote copies written to commit log disks

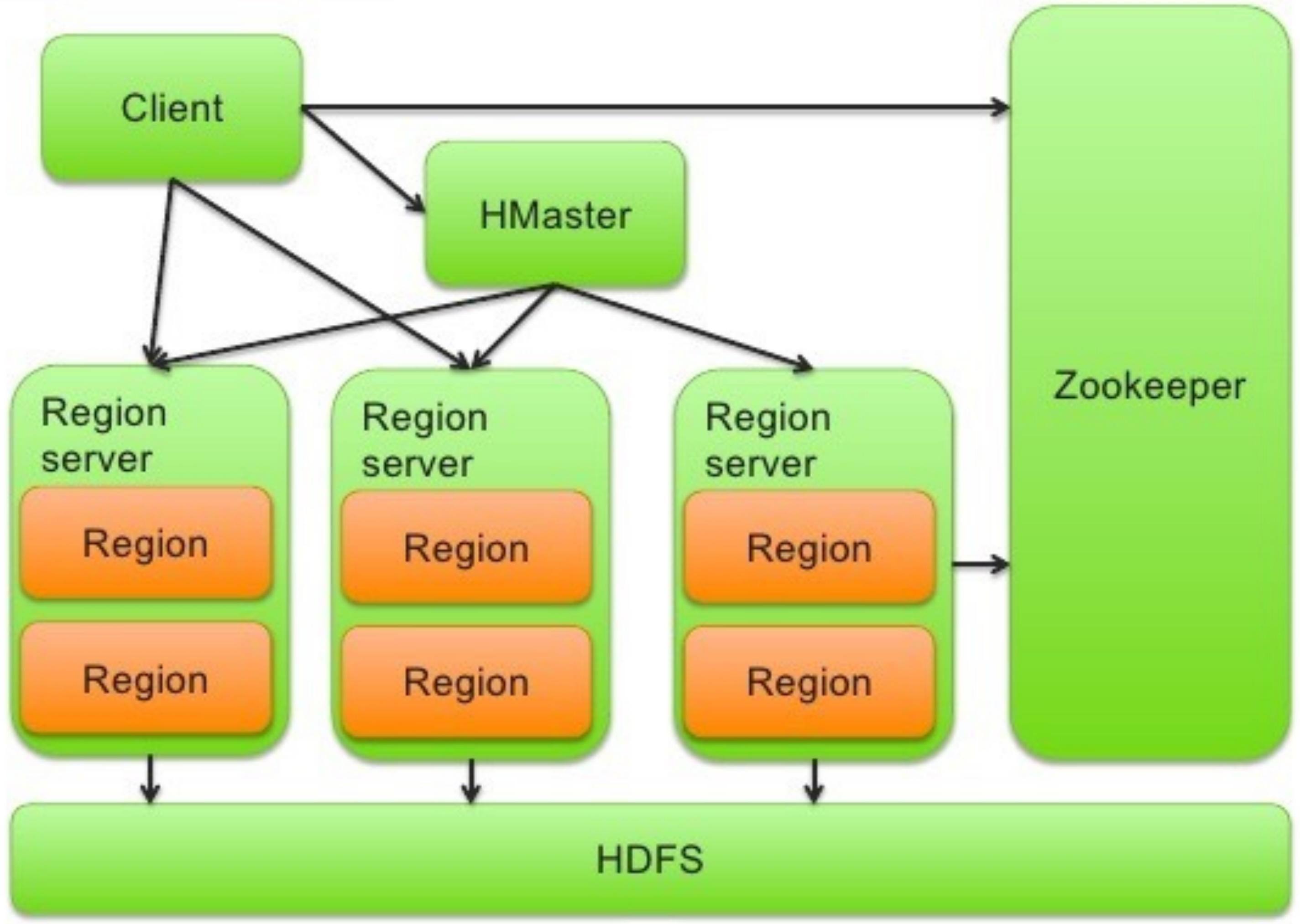
If a node or region goes offline, hinted handoff completes the write when the node comes back up. Nightly global compare and repair jobs ensure everything stays consistent.



NETFLIX

APACHE
HBASE

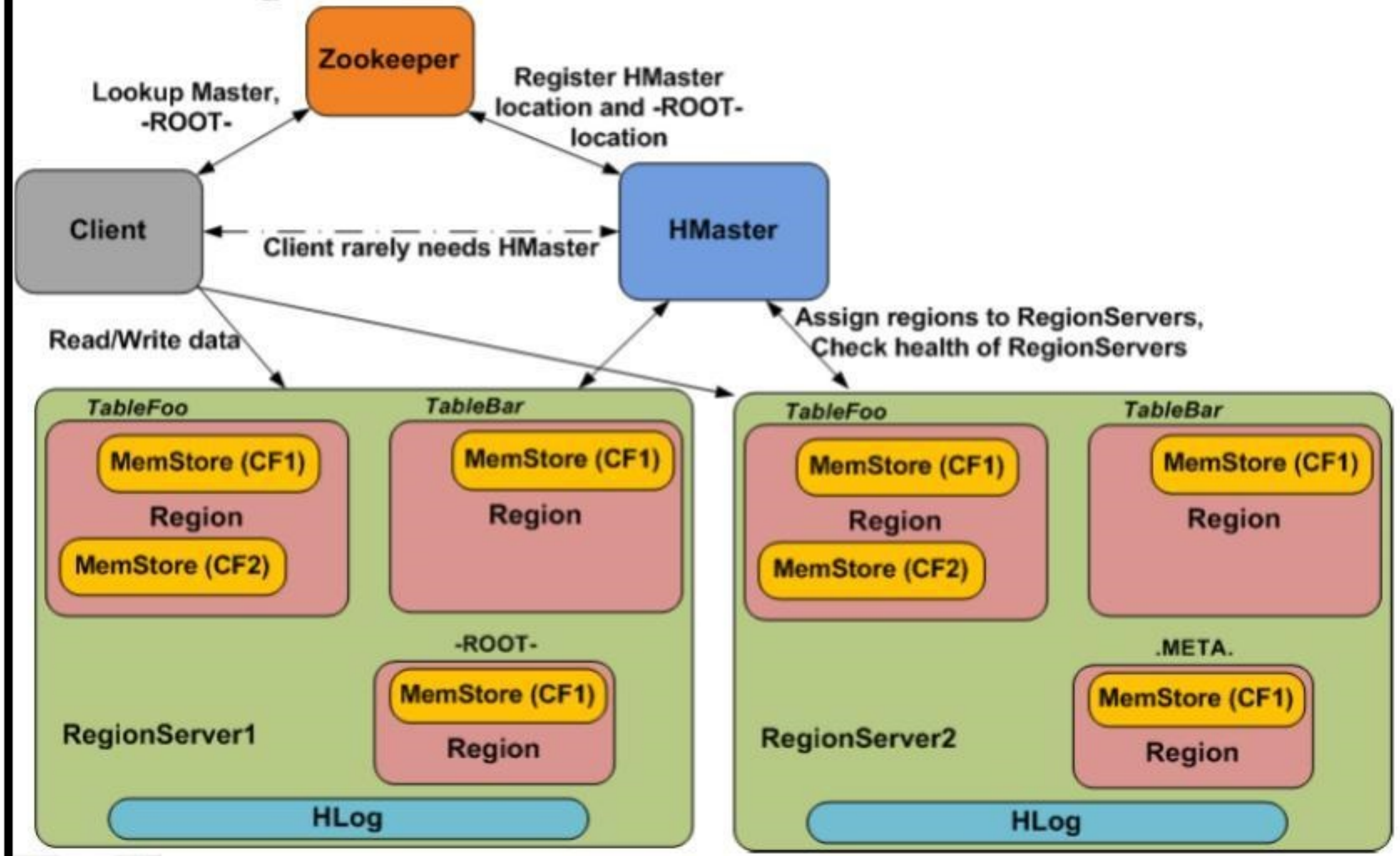
Apache HBase Architecture



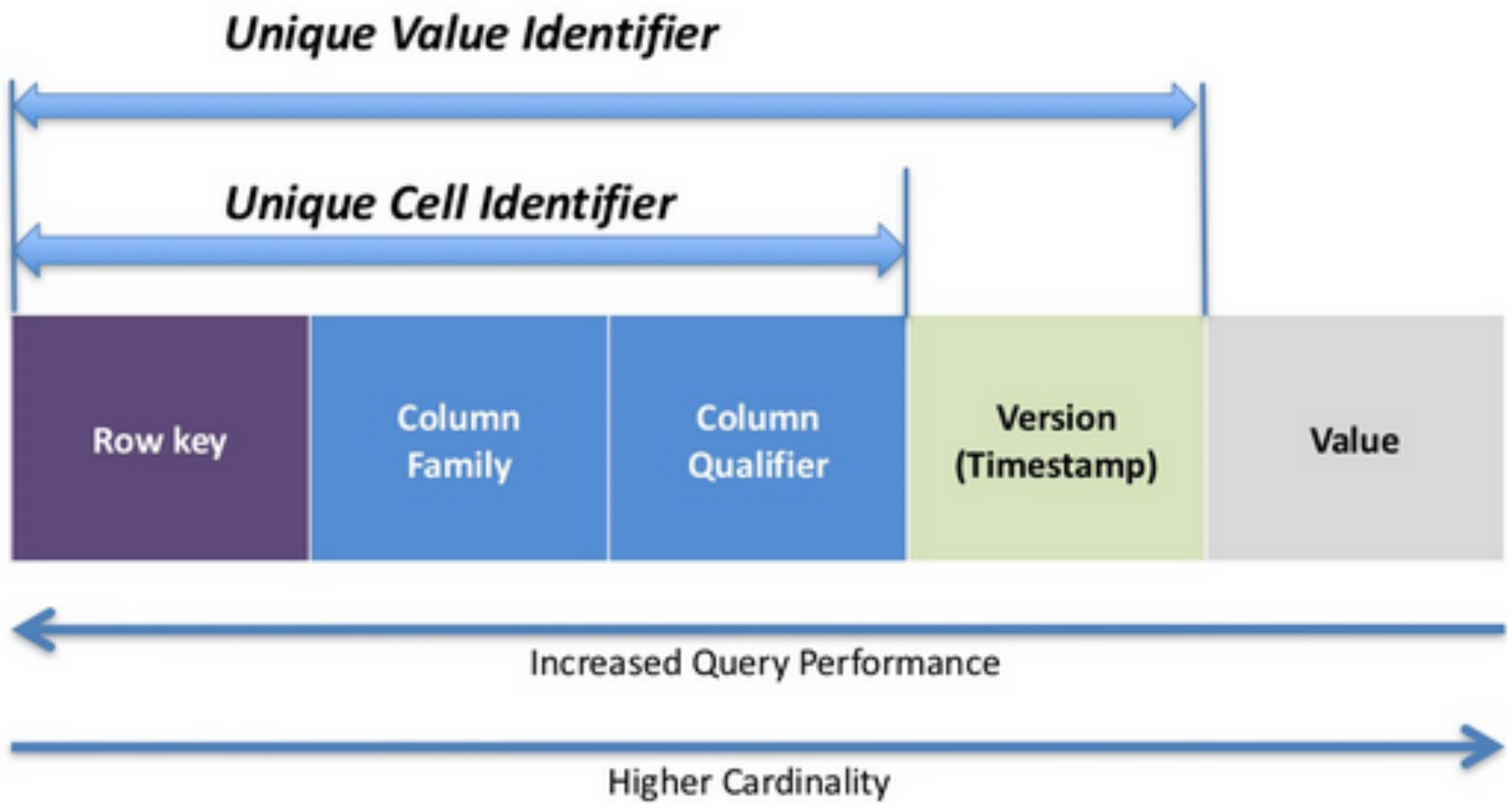
HBase Features

- A distributed, scalable, big data store.
- Random, realtime read/write access to your Big Data.
- Linear and modular scalability.
- Strictly consistent reads and writes.
- Automatic and configurable sharding of tables
- Automatic failover support between RegionServers.
- Easy to use Java API for client access.
- Block cache and Bloom Filters for real-time queries.
- Query predicate push down via server side Filters
- Thrift gateway and a REST-ful Web service that supports XML, Protobuf, and binary data encoding options

HBase high-level architecture



Row Schema

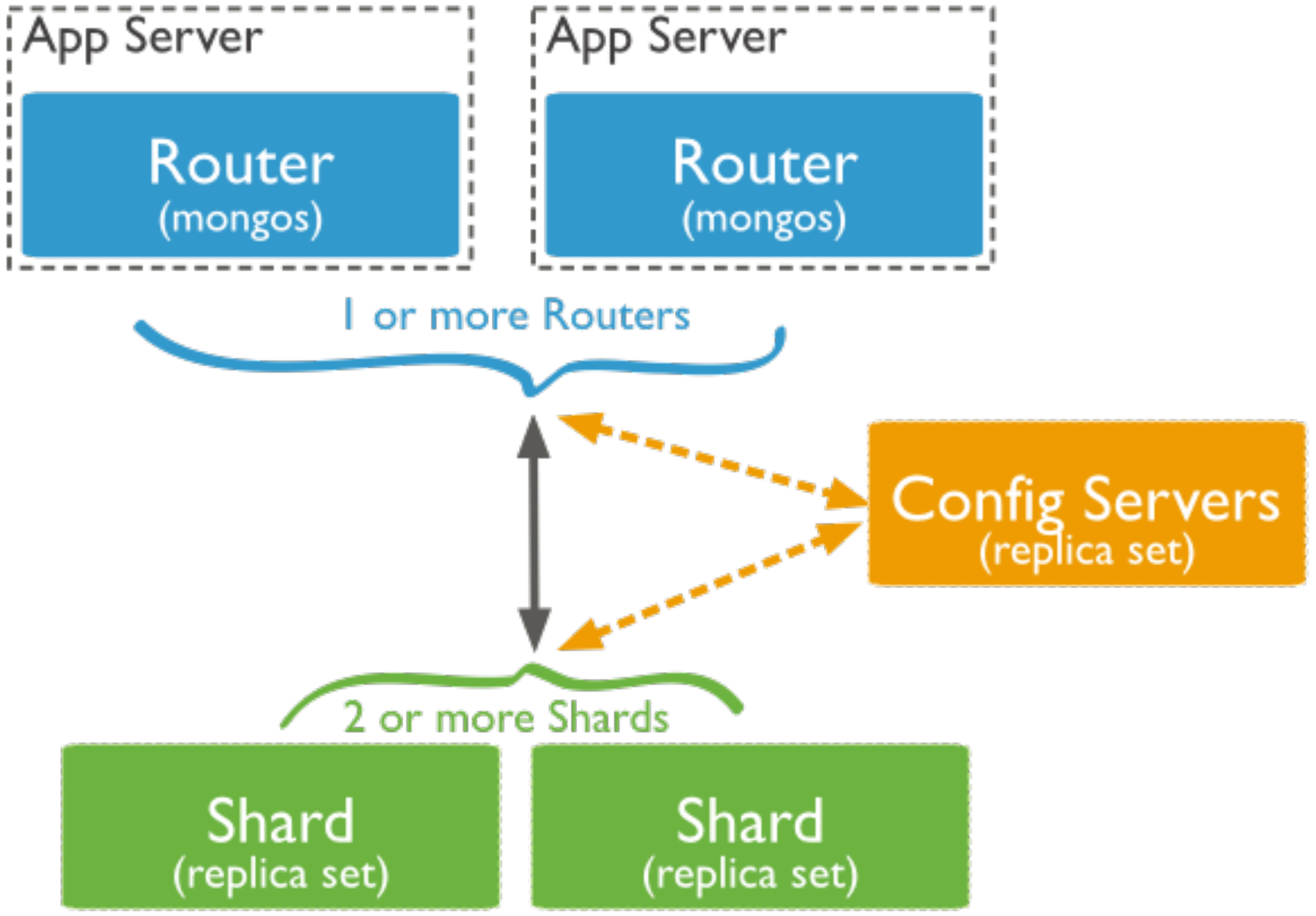

Unique Value Identifier
Unique Cell Identifier
Row key
Column Family
Column Qualifier
Version (Timestamp)
Value
Increased Query Performance
Higher Cardinality

mongoDB

MongoDB Features

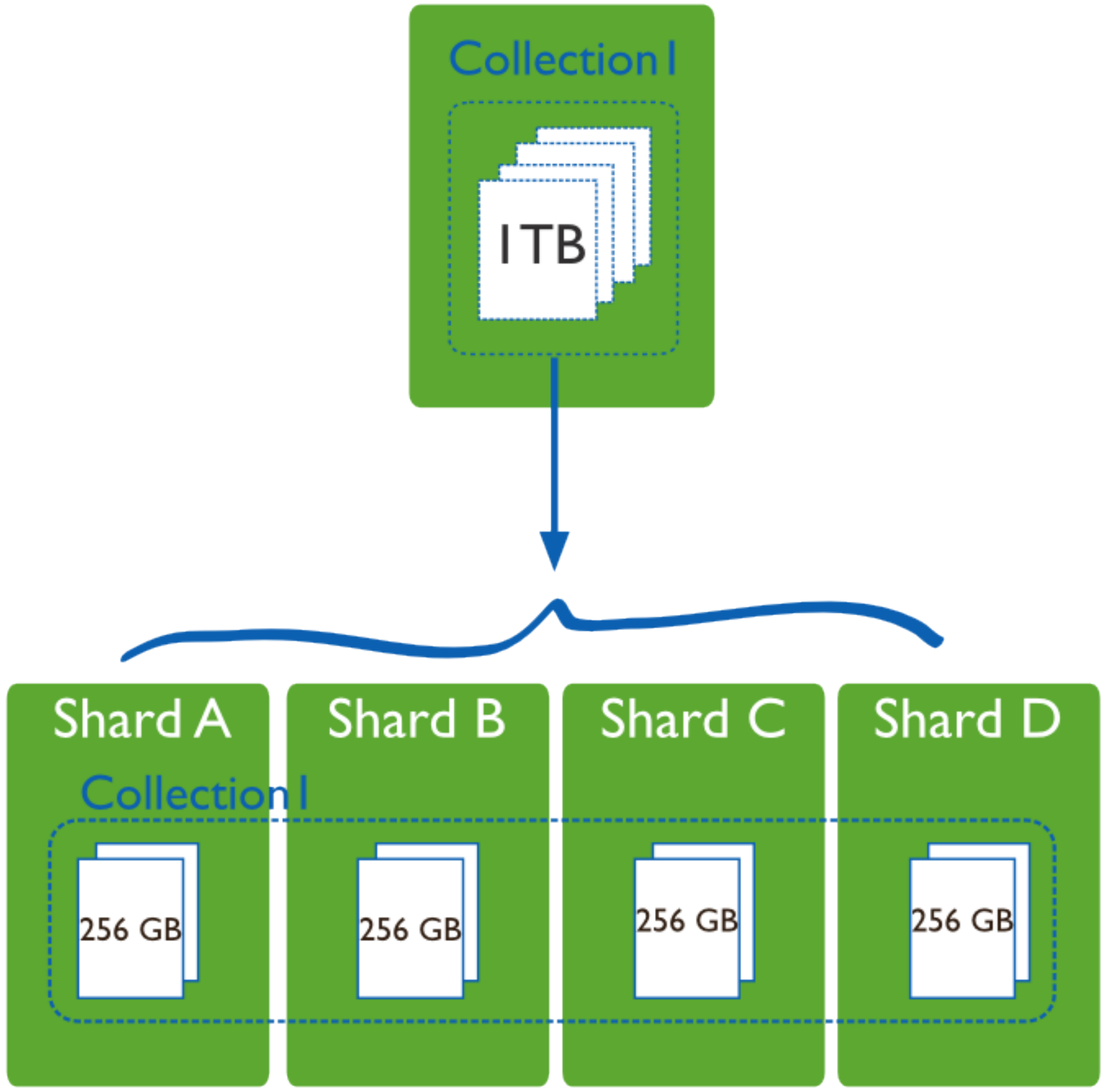
- document database
- binary JSON and flexible schema
- Index and ad-hoc query
- replication
- load balancing and HA
- scaling-out

MongoDB

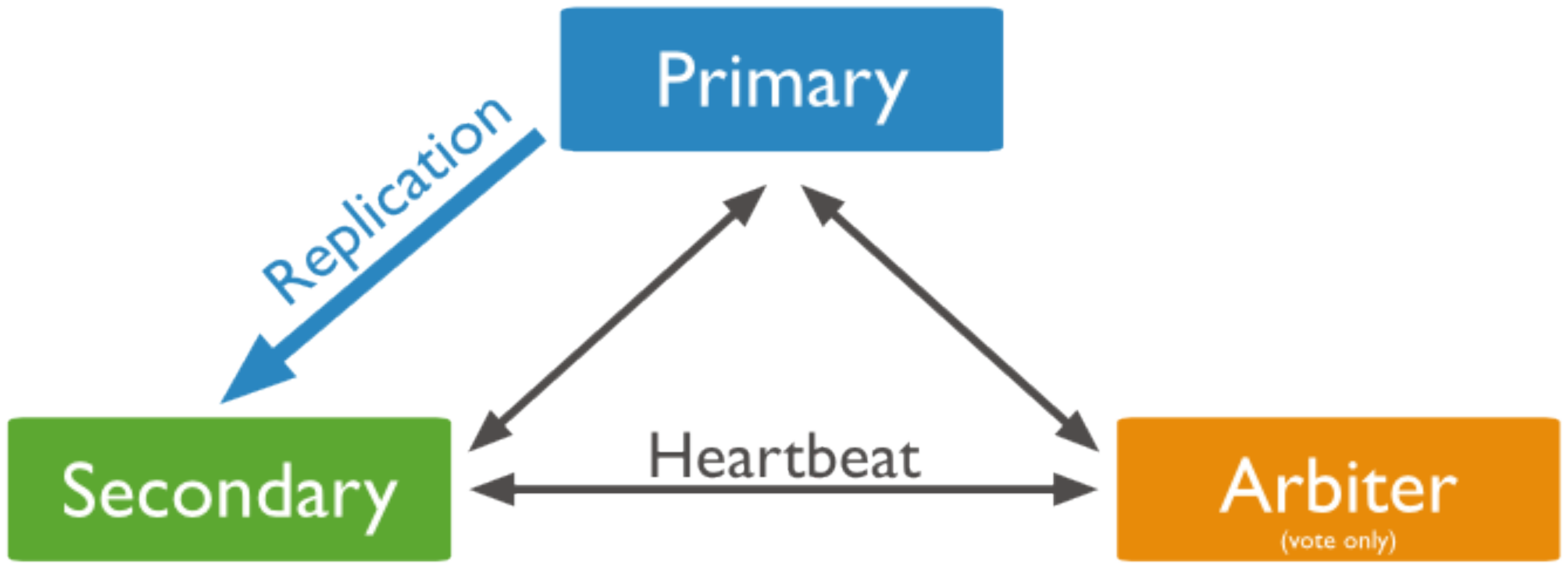


- replica set (master/slave)
- shard (replica set within a cluster)
- config server (topology)
- mongos (router)
- shard key

Sharding


Collection I
1TB
Shard A
Shard B
Shard C
Shard D
Collection I
256 GB
256 GB
256 GB
256 GB

Arbiter


Primary
Replication
Heartbeat
Secondary
Arbiter
(vote only)