MongoDB 分布式架构演进

张友东（林青）

阿里云数据库技术团队

Mongo

Mongo as in "humongous". Used to describe something extremely large or important.

MongoDB 核心优势

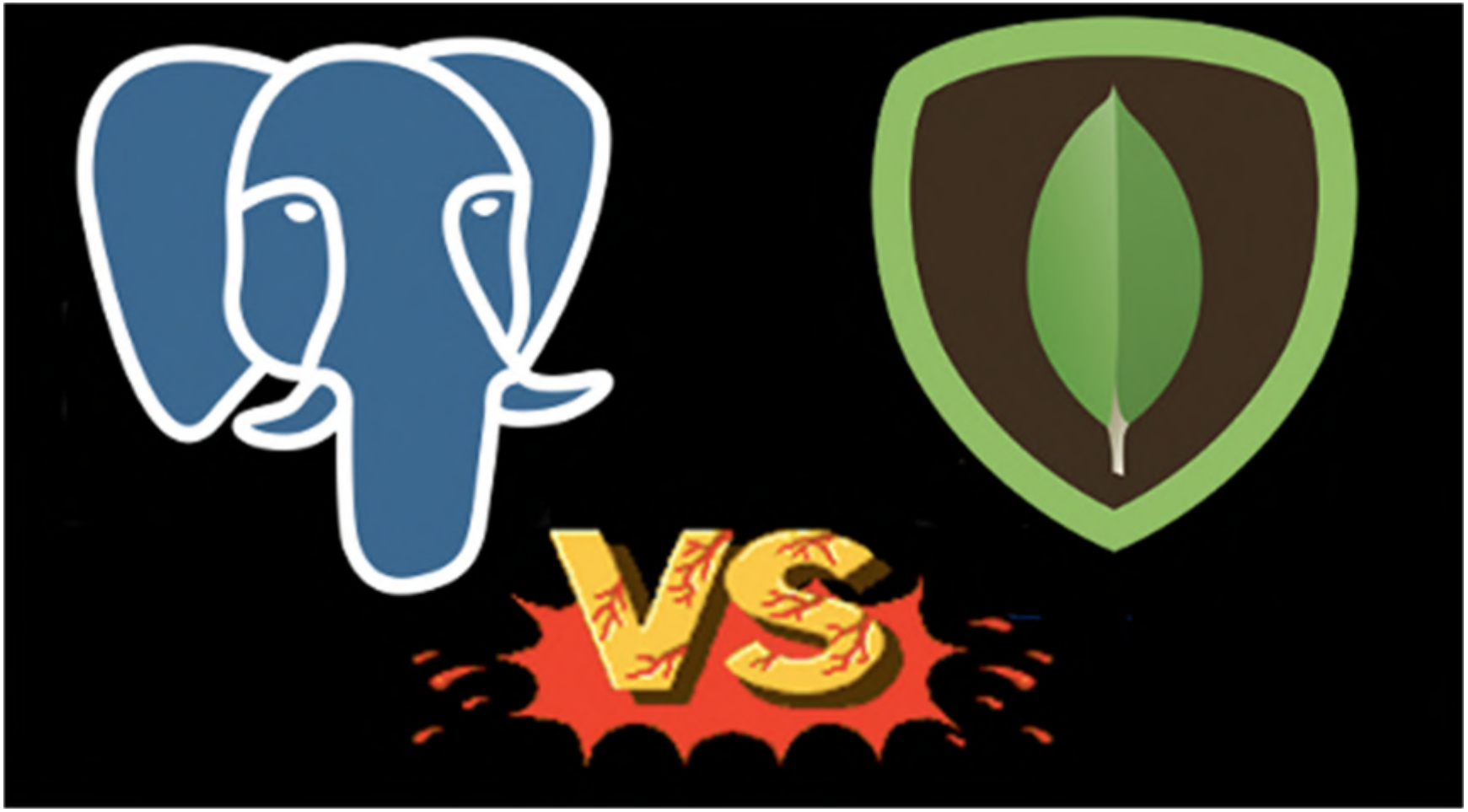
灵 活	•文档模型
高可用	•复制集
可扩展	•分片集群

文档模型

```
{
    "_id" : ObjectId("5798a011b81541133e0b137f"),
    "name" : "jack",
    "age" : 23,
    "sex" : "M",
    "hobby" : [
        "running",
        "football",
        "movie"
    ],
    "contacts" : [
        {
            "type" : "home",
            "number" : "12345678"
        },
        {
            "type" : "office",
            "number" : "87654321"
        }
    ]
}
```

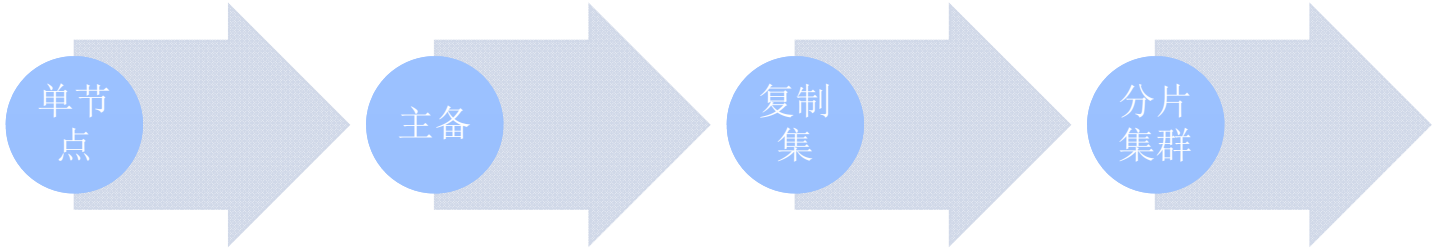
- 接近真实对象模型，对开发人员友好
- Schema free，适应灵活多变的需求，快速迭代
- 数组、内嵌文档支持，数据聚集，提升读写性能

今天不谈文档模型

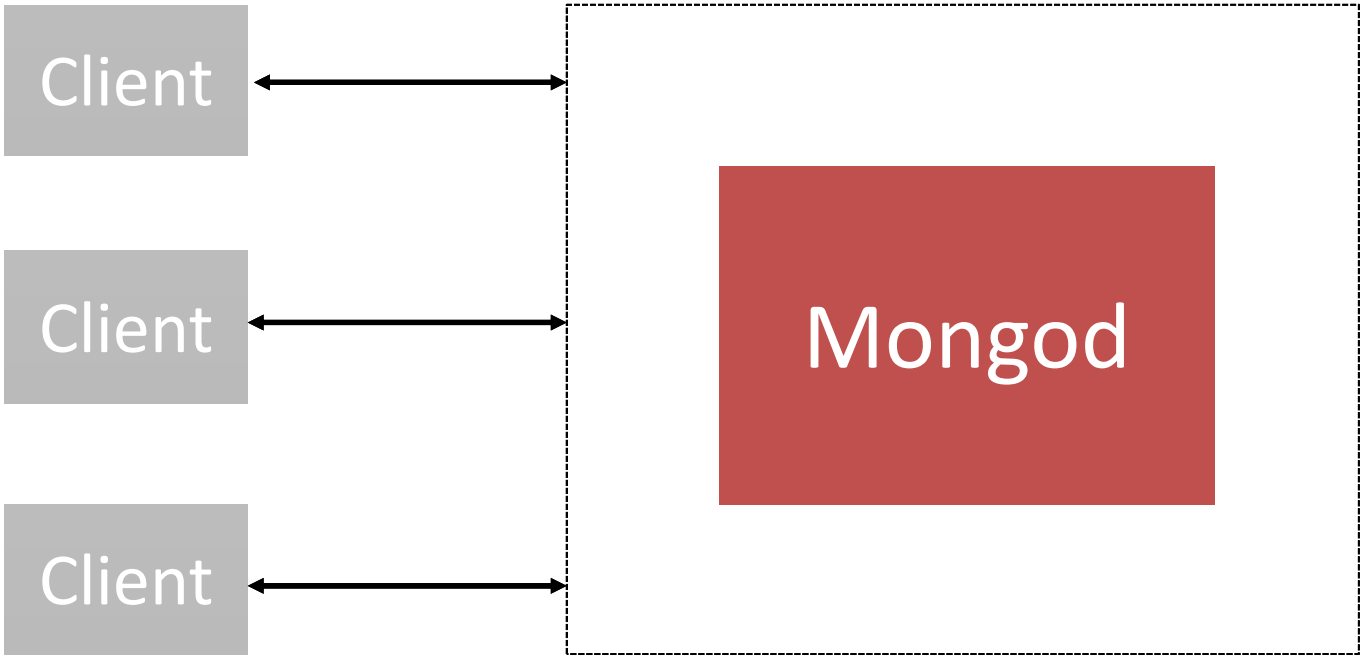


主要内容

- 如何保证数据高可靠？
- 如何保证服务高可用？
- 如何实现水平扩展？

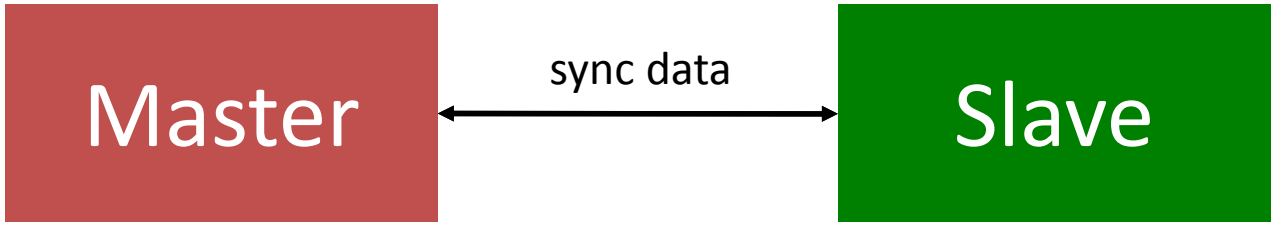


单节点



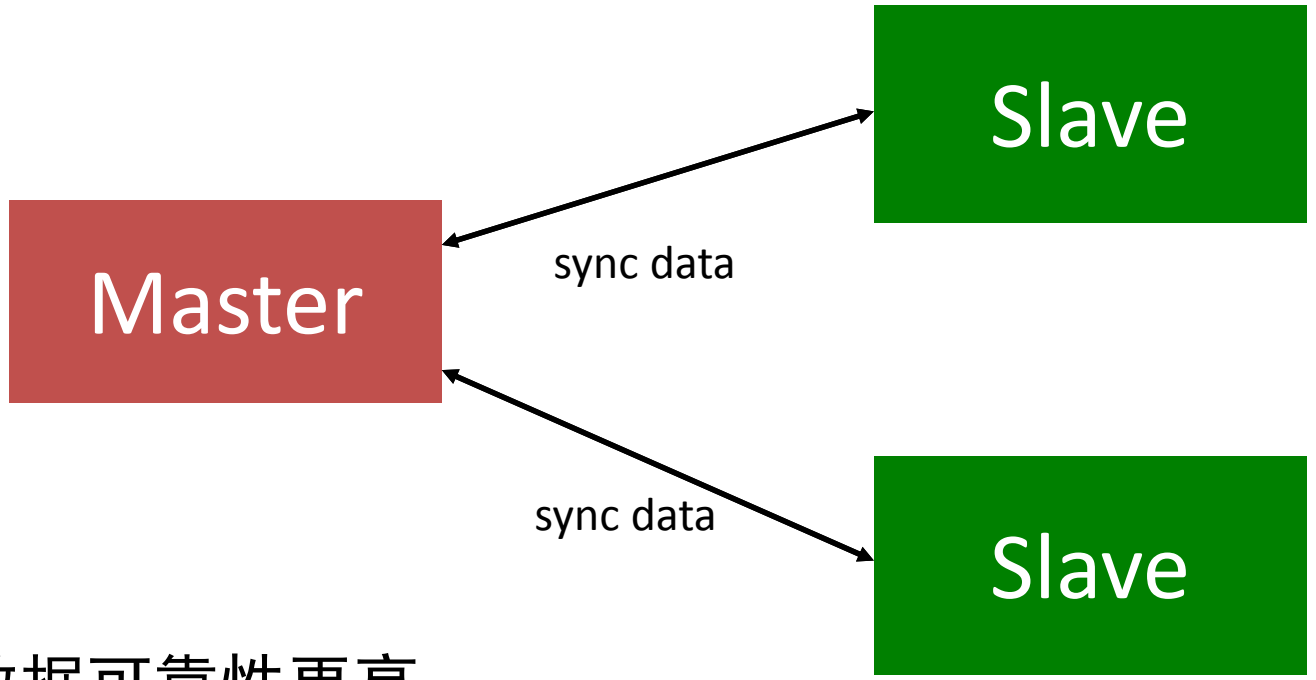
- 数据单点
- 服务单点

主备节点



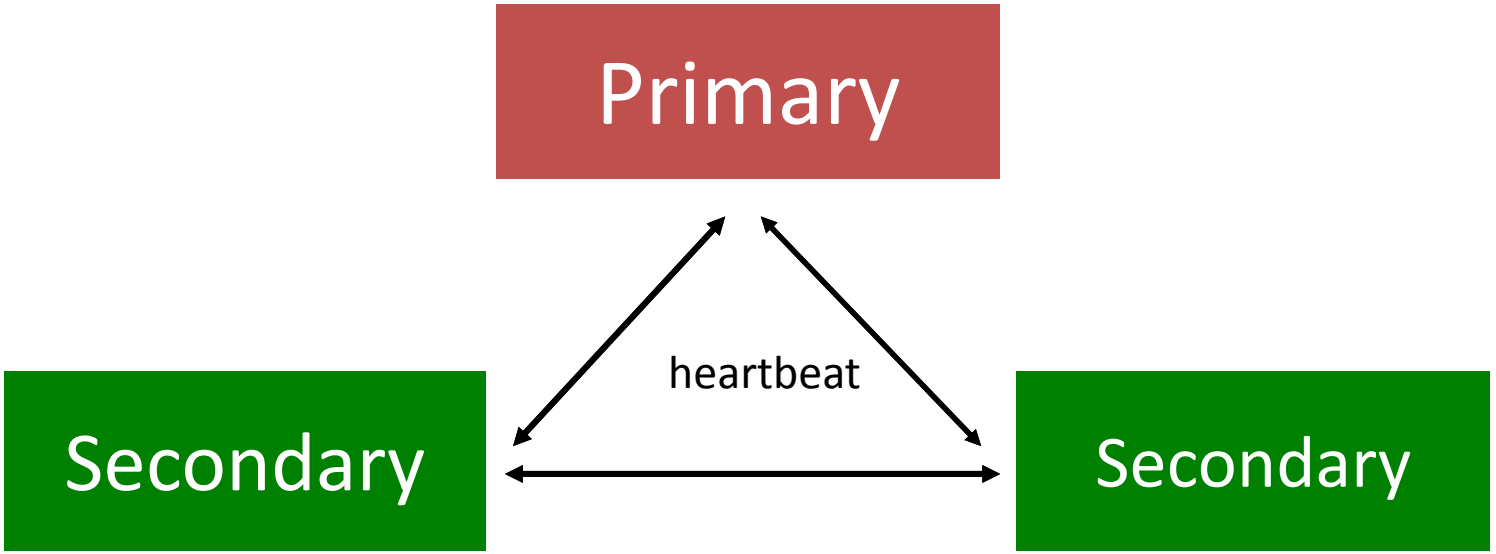
- Master 宕机无法服务写请求
- 只能容忍一个节点失效

一主多备



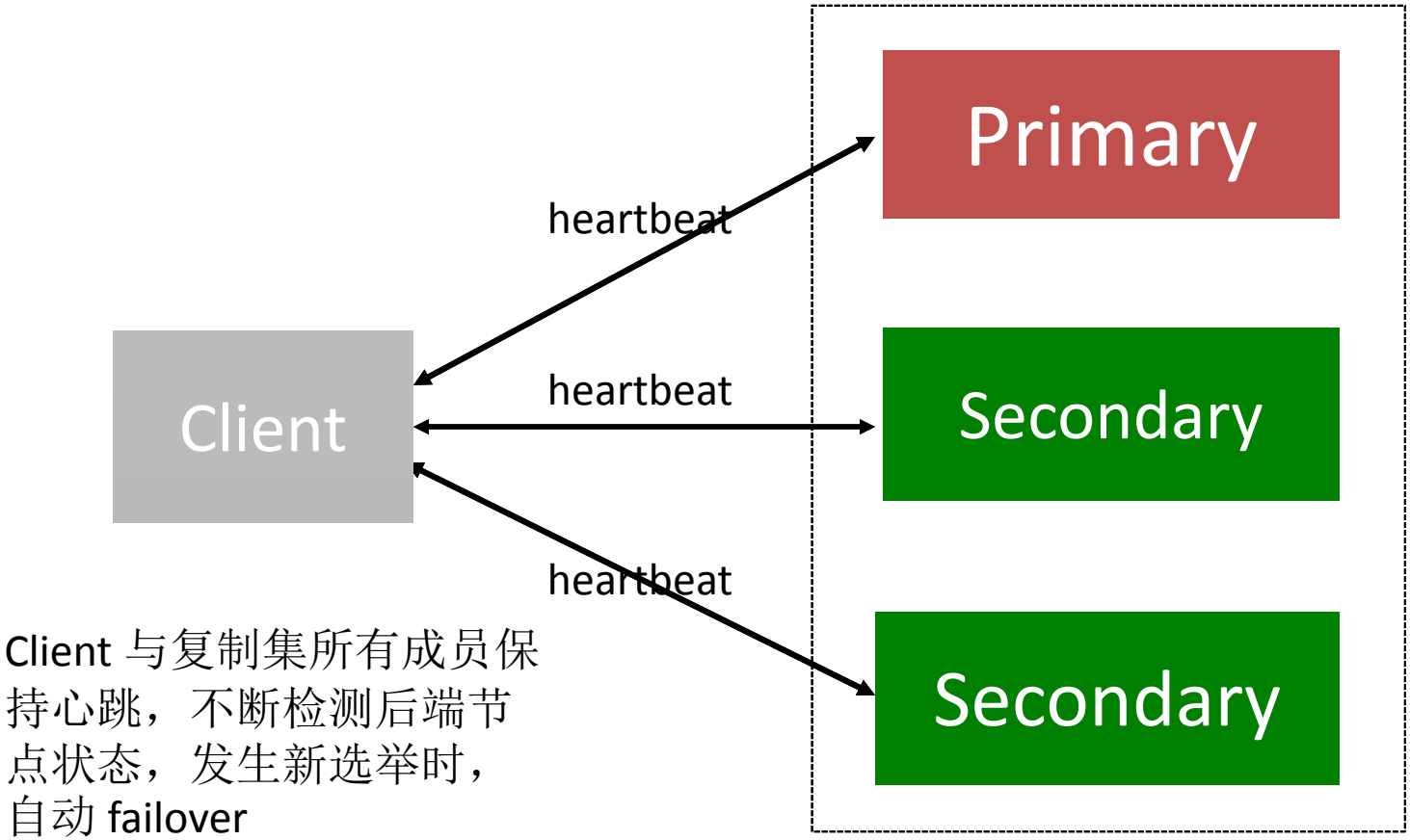
- 数据可靠性更高
- 扩展读服务能力

复制集



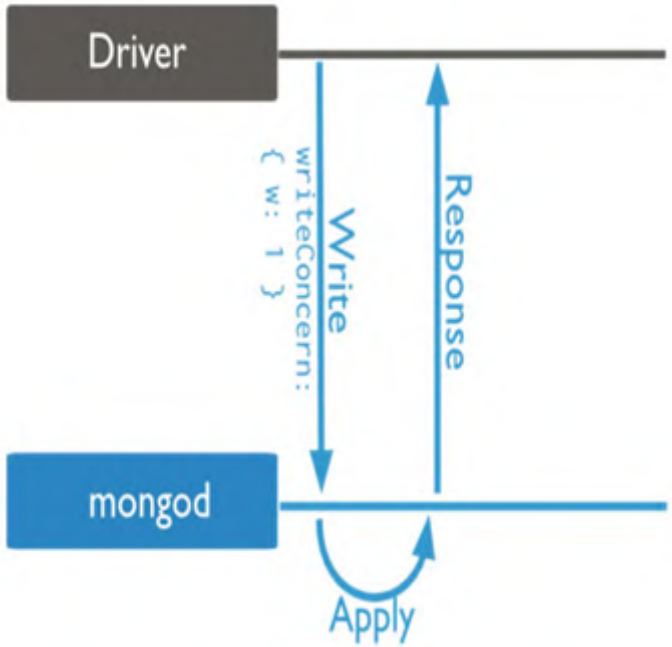
- 通过 raft 协议选举出 Primary
- 所有写请求都写到 Primary，并同步到 Secondary
- 当 Primary 故障时，自动选出新的 Primary 节点

高可用

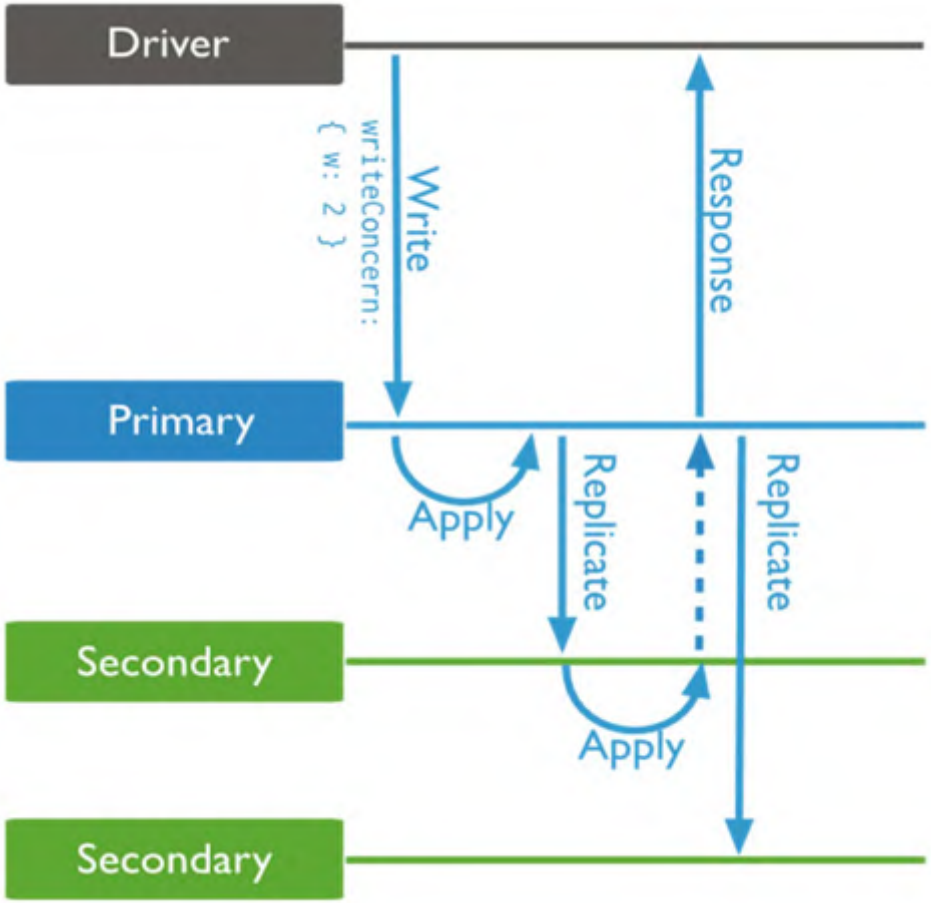


Client 与复制集所有成员保持心跳，不断检测后端节点状态，发生新选举时，自动 failover

写策略



WriteConcern: {w: 1}



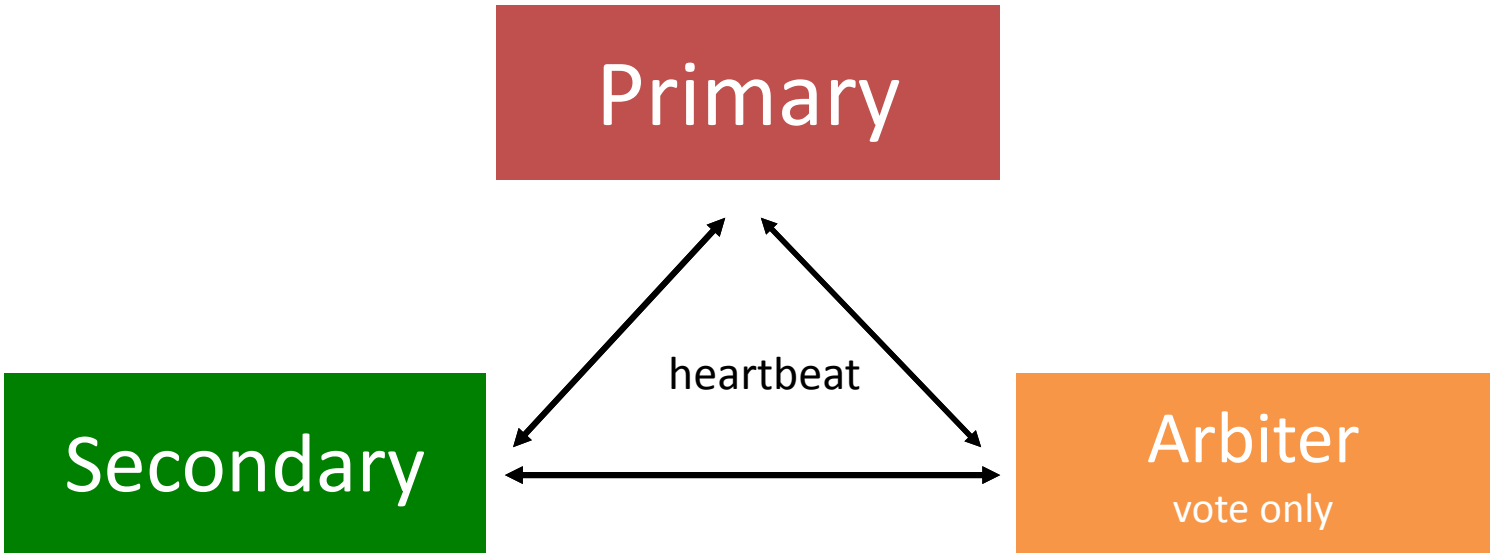
WriteConcern: {w: 2}

选举规则

https://raft.github.io/

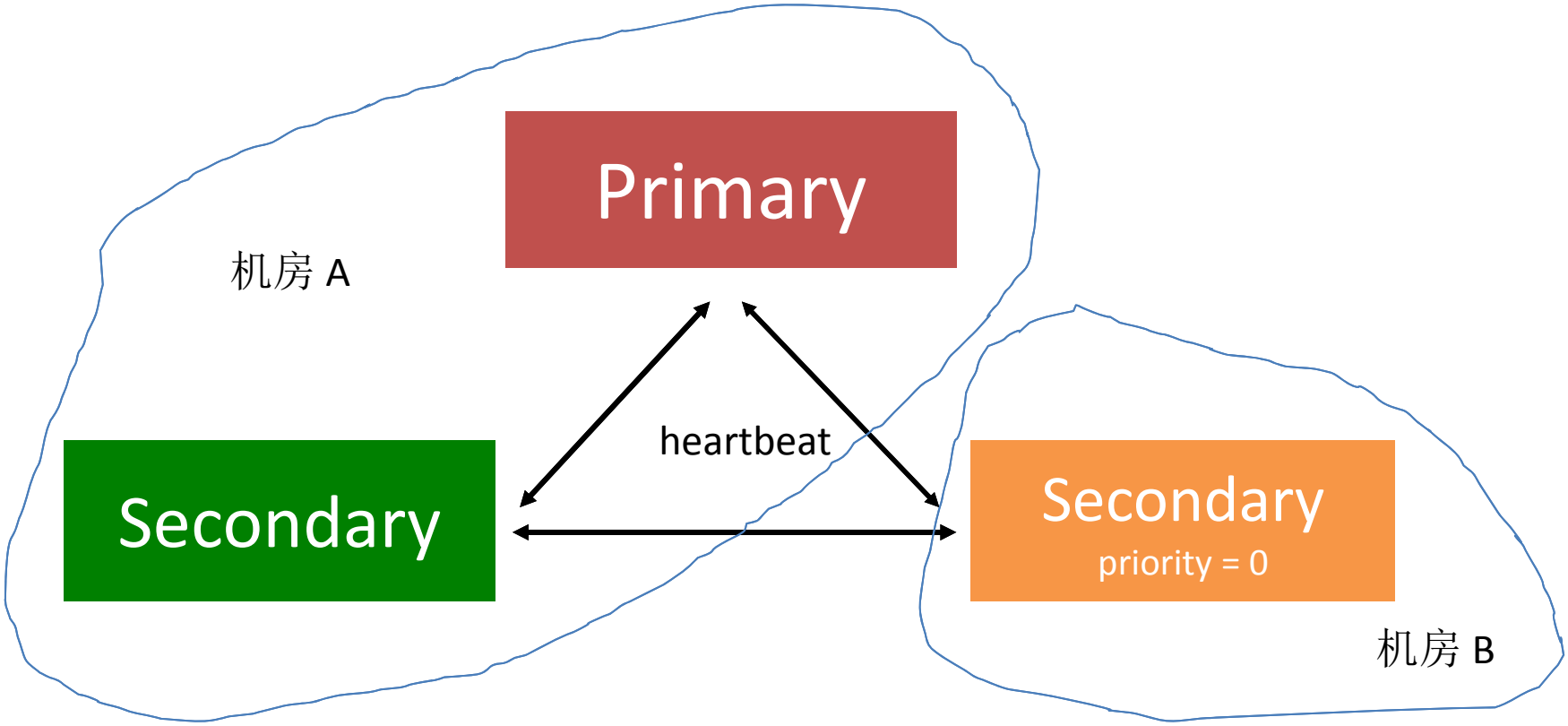
节点数	大多数
1	1
2	2
3	2
4	3
5	3
6	4
7	4

仲裁节点

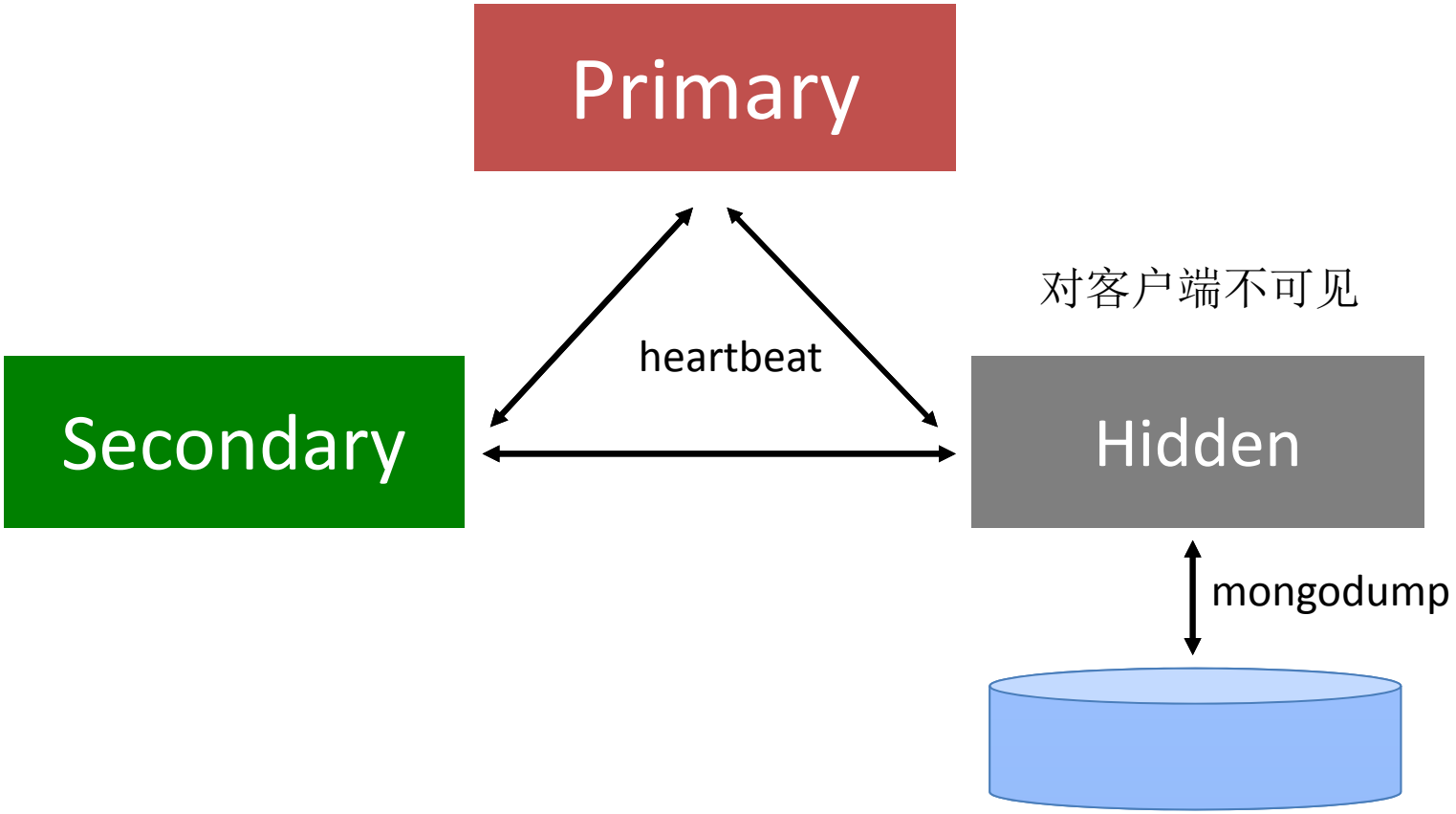


Arbiter 节点只参与投票，不存储数据

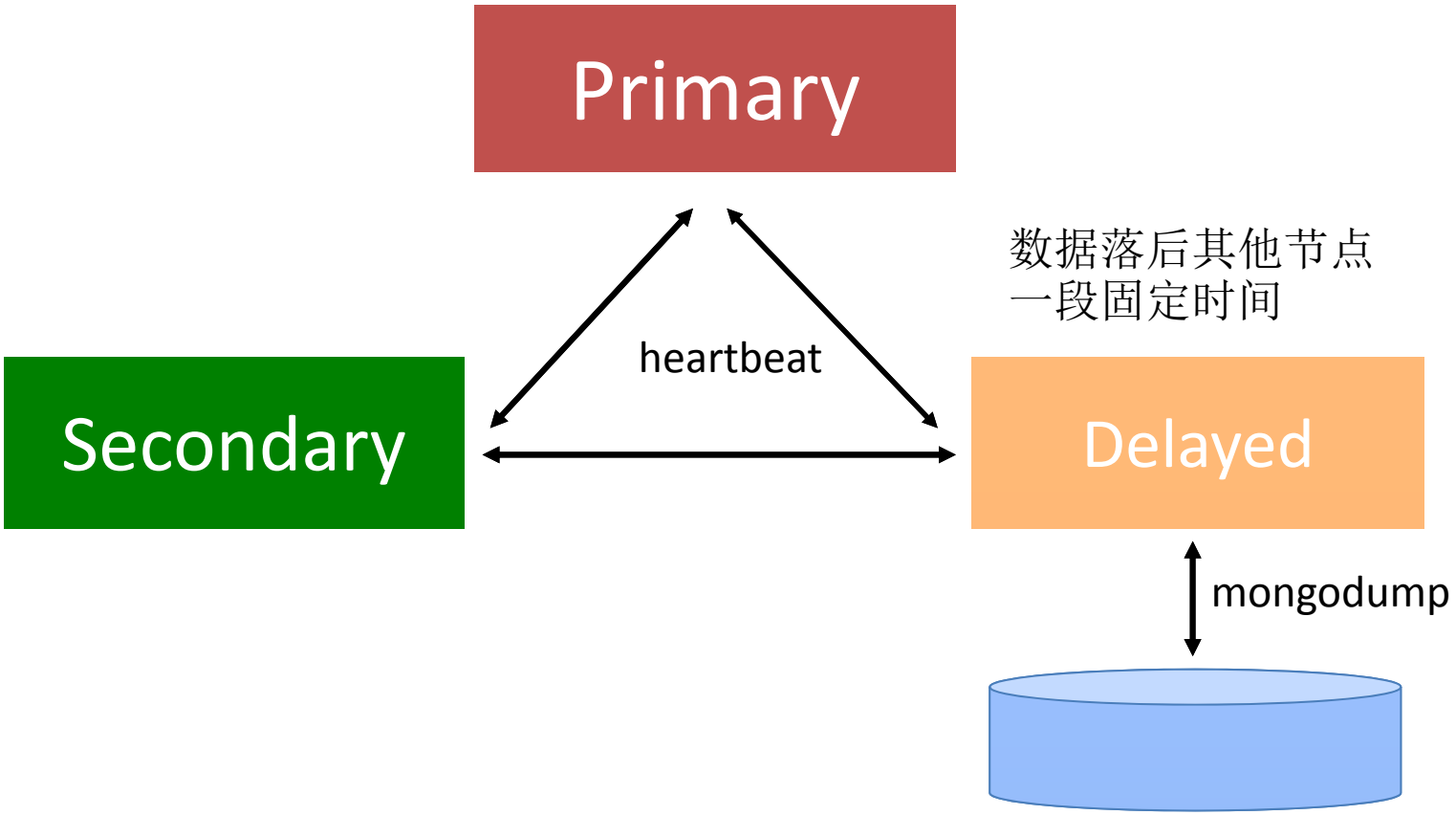
选举优先级



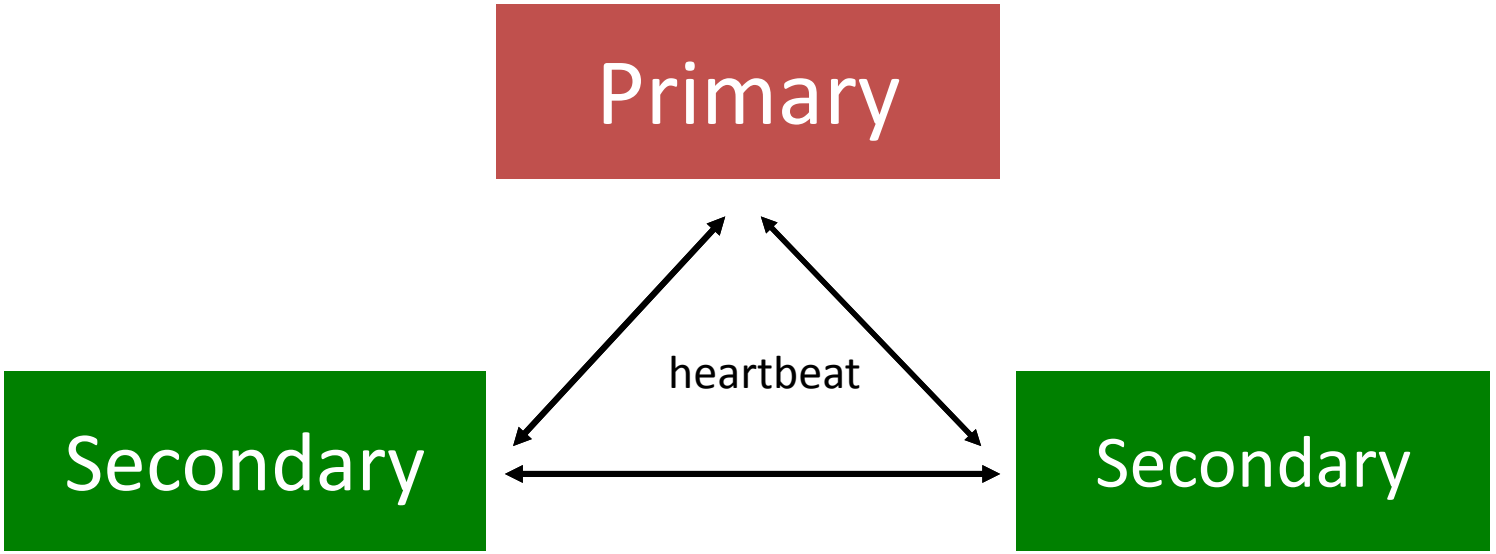
隐藏节点



延迟节点

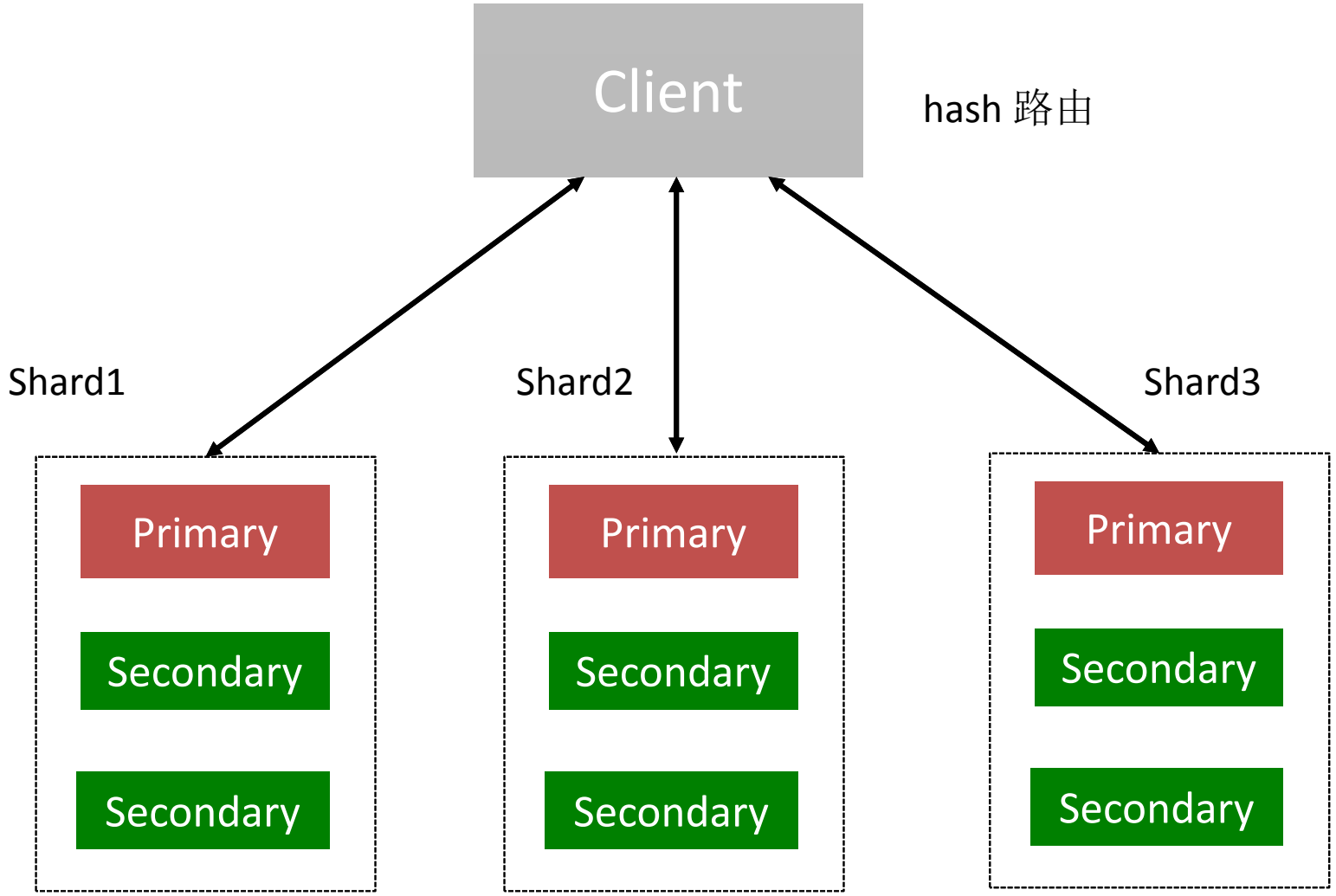

Primary
数据落后其他节点
一段固定时间
heartbeat
Secondary
Delayed
mongodump

复制集的问题

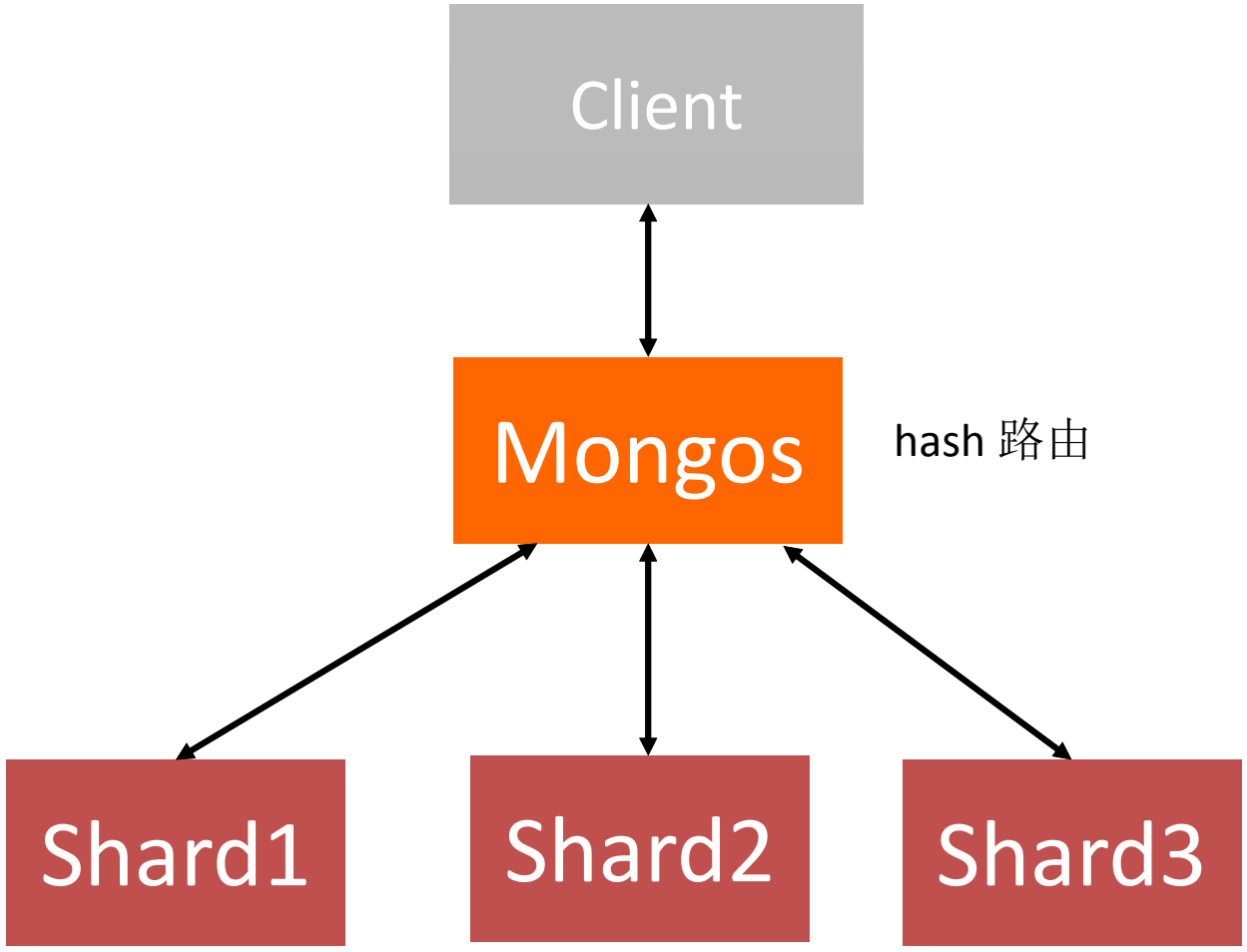


- 存储容量受限于单个 Primary
- 写服务能力受限于单个 Primary

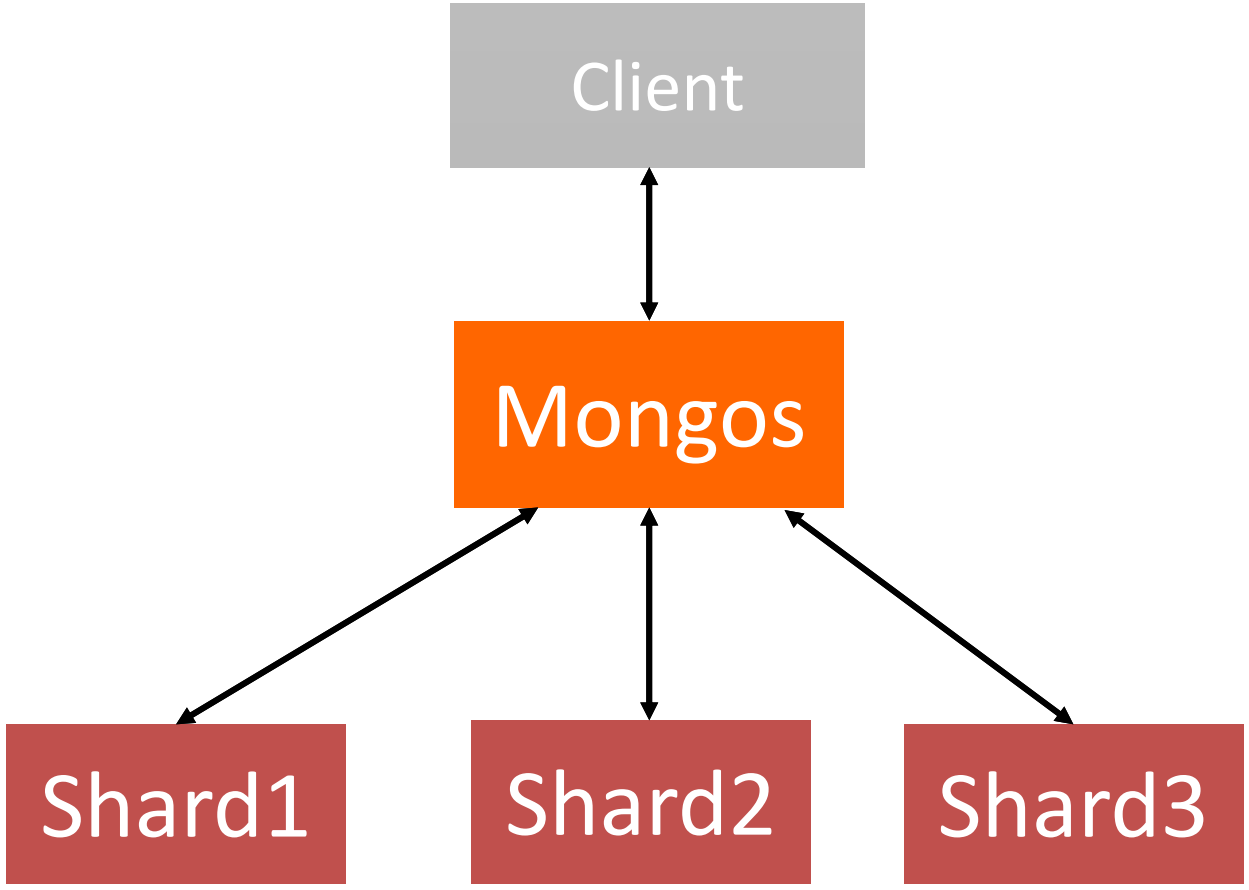
客户端分片


Client
hash 路由
Shard1
Shard2
Shard3
Primary
Secondary
Secondary
Primary
Secondary
Secondary
Primary
Secondary
Secondary

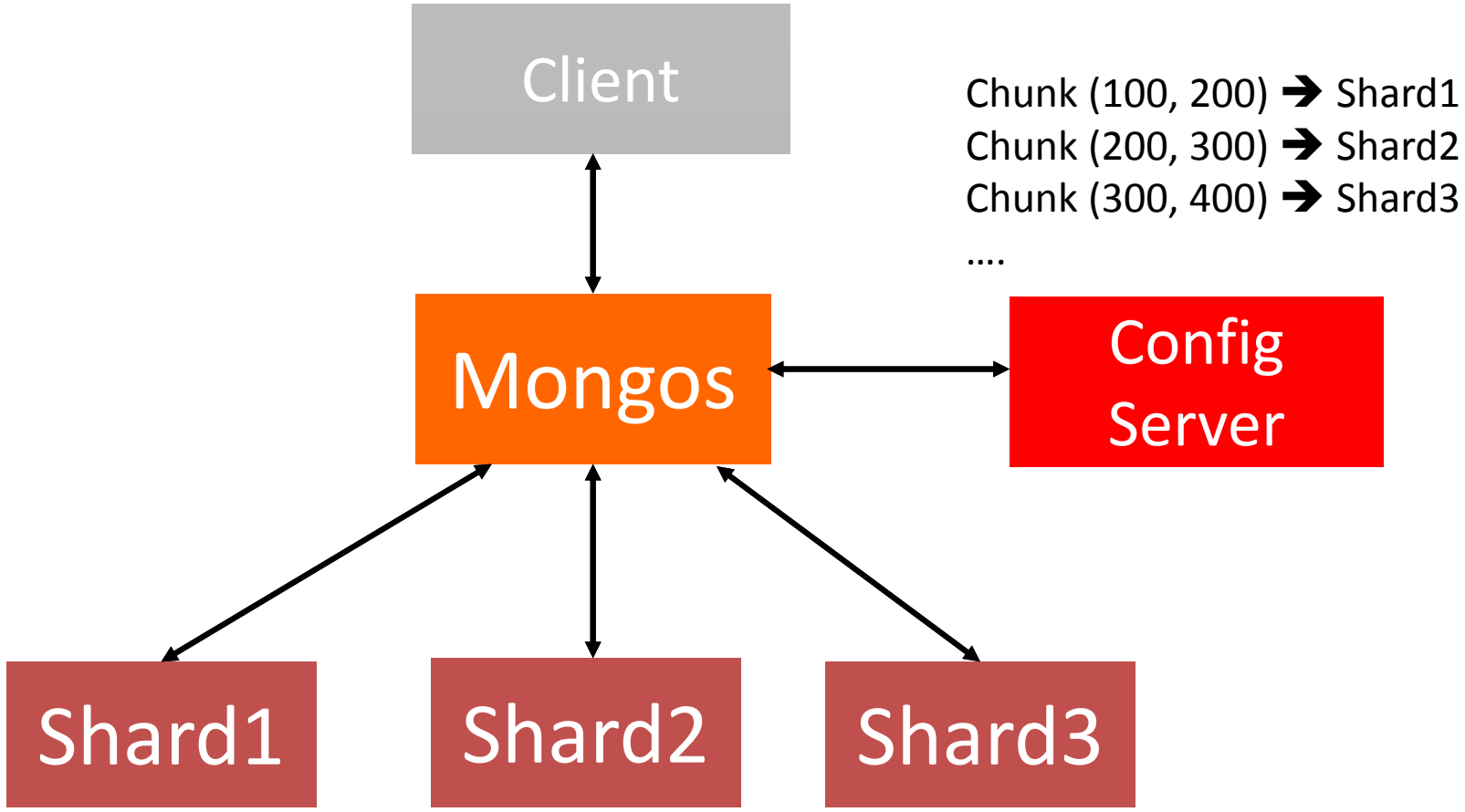
Proxy分片



Proxy分片



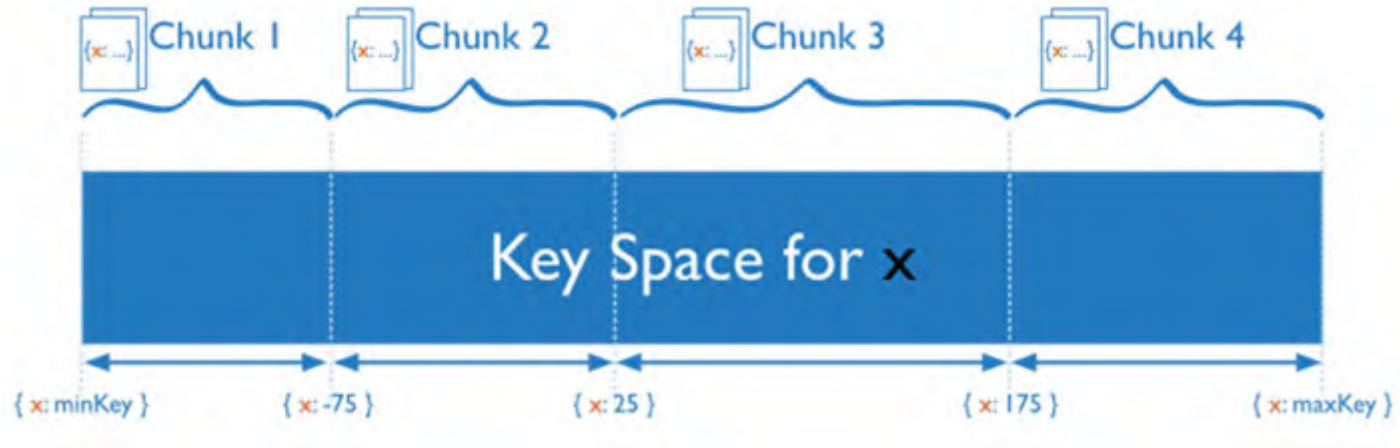
可扩展分片


Client
Chunk (100, 200) ➔ Shard1
Chunk (200, 300) ➔ Shard2
Chunk (300, 400) ➔ Shard3
....
Mongos
Config Server
Shard1
Shard2
Shard3

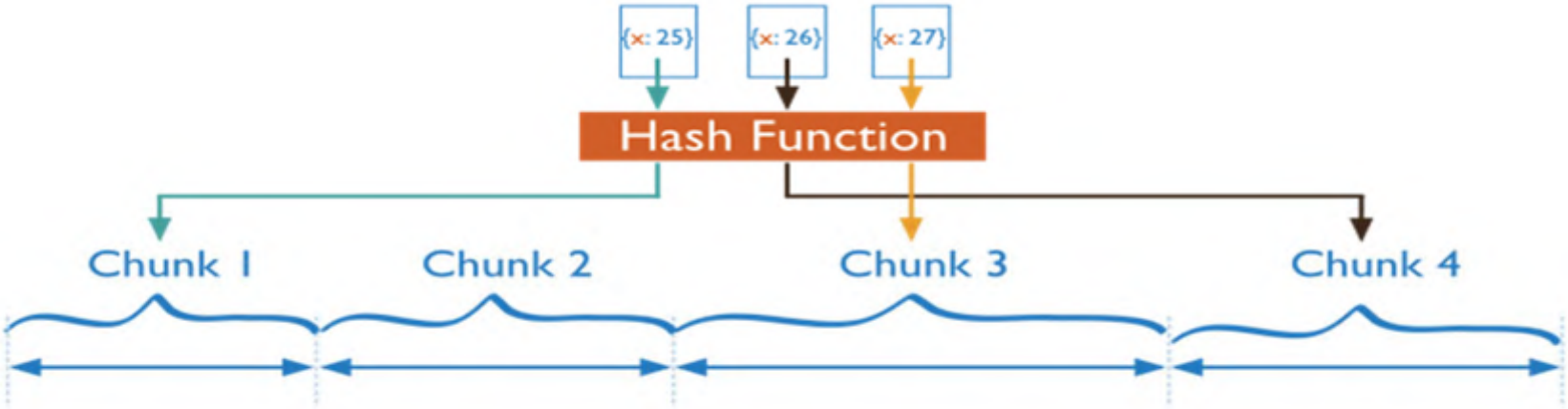
分片方式-范围

范围	所在分片
Chunk1 [minKey, -75)	Shard2
Chunk2 [-75, 25)	Shard1
Chunk3 [25, 175)	Shard3
Chunk4 [175, MaxKey]	Shard1

- 根据某个字段的值，顺序划分为多个范围，每个范围对应一个 Shard，能很好的支持范围查询

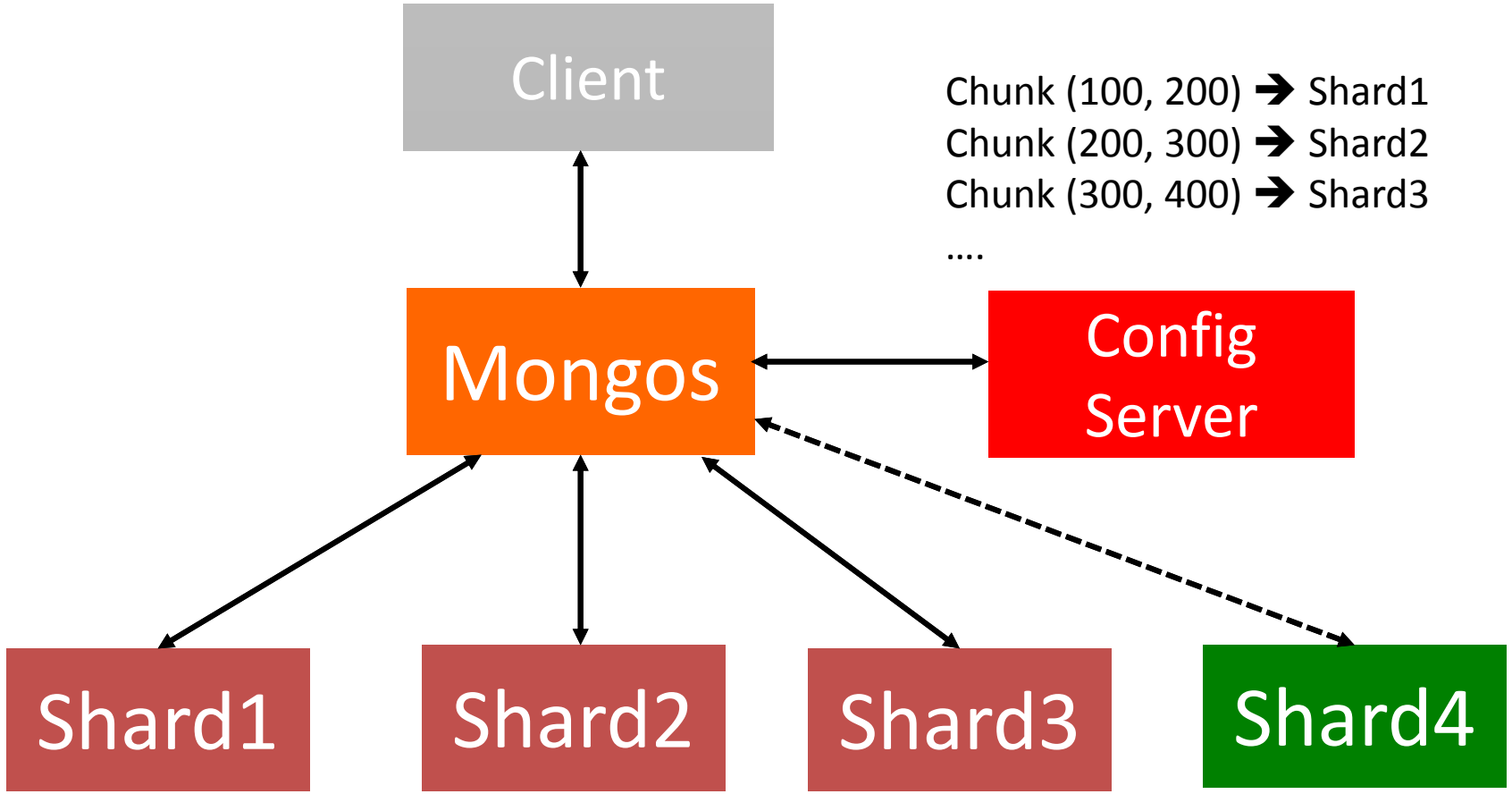


分片方式-hash

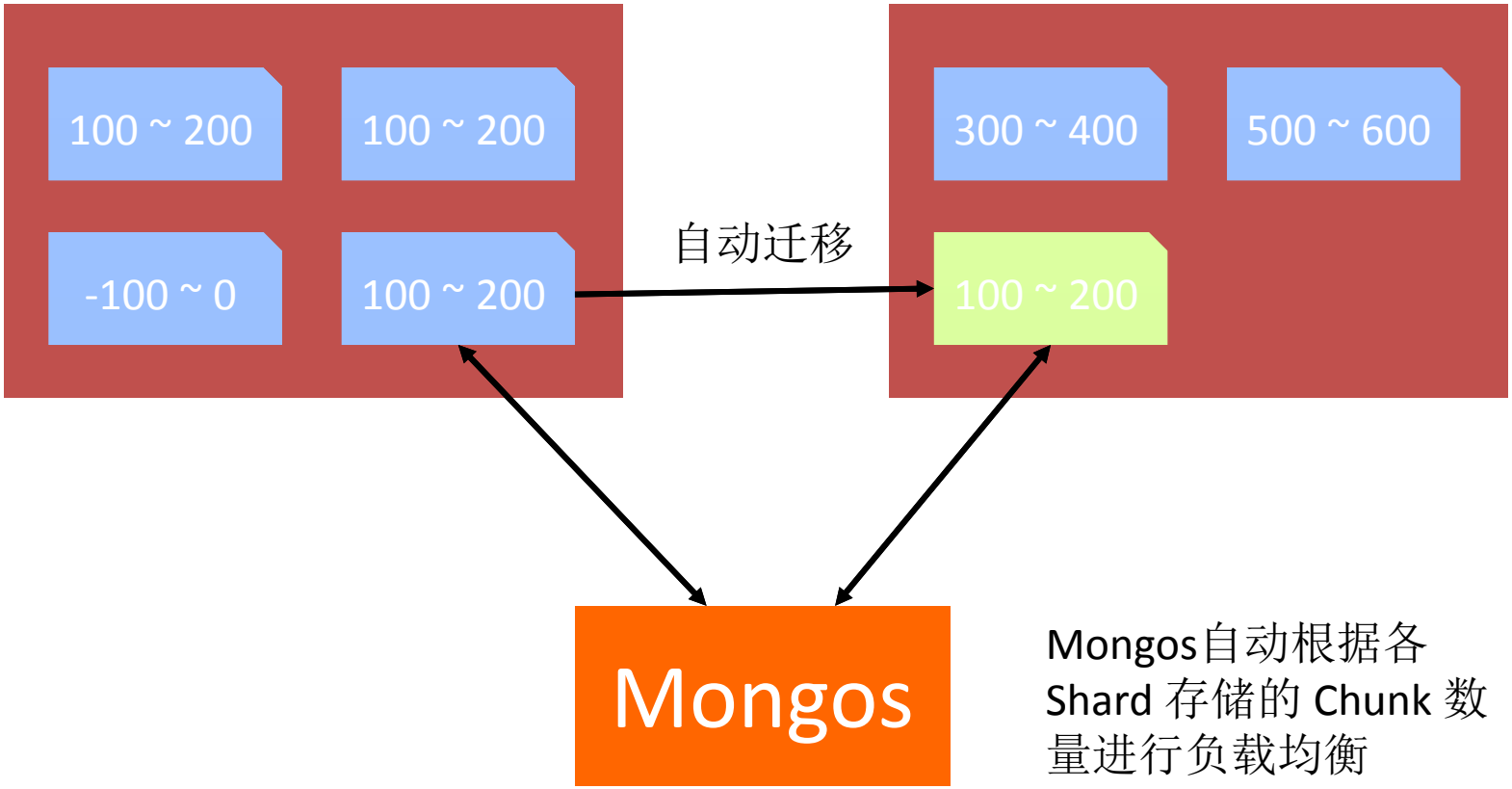


- 根据某个字段的hash值，顺序划分为多个范围，每个范围对应一个 Shard，能将数据均匀的分布到各个 Shard

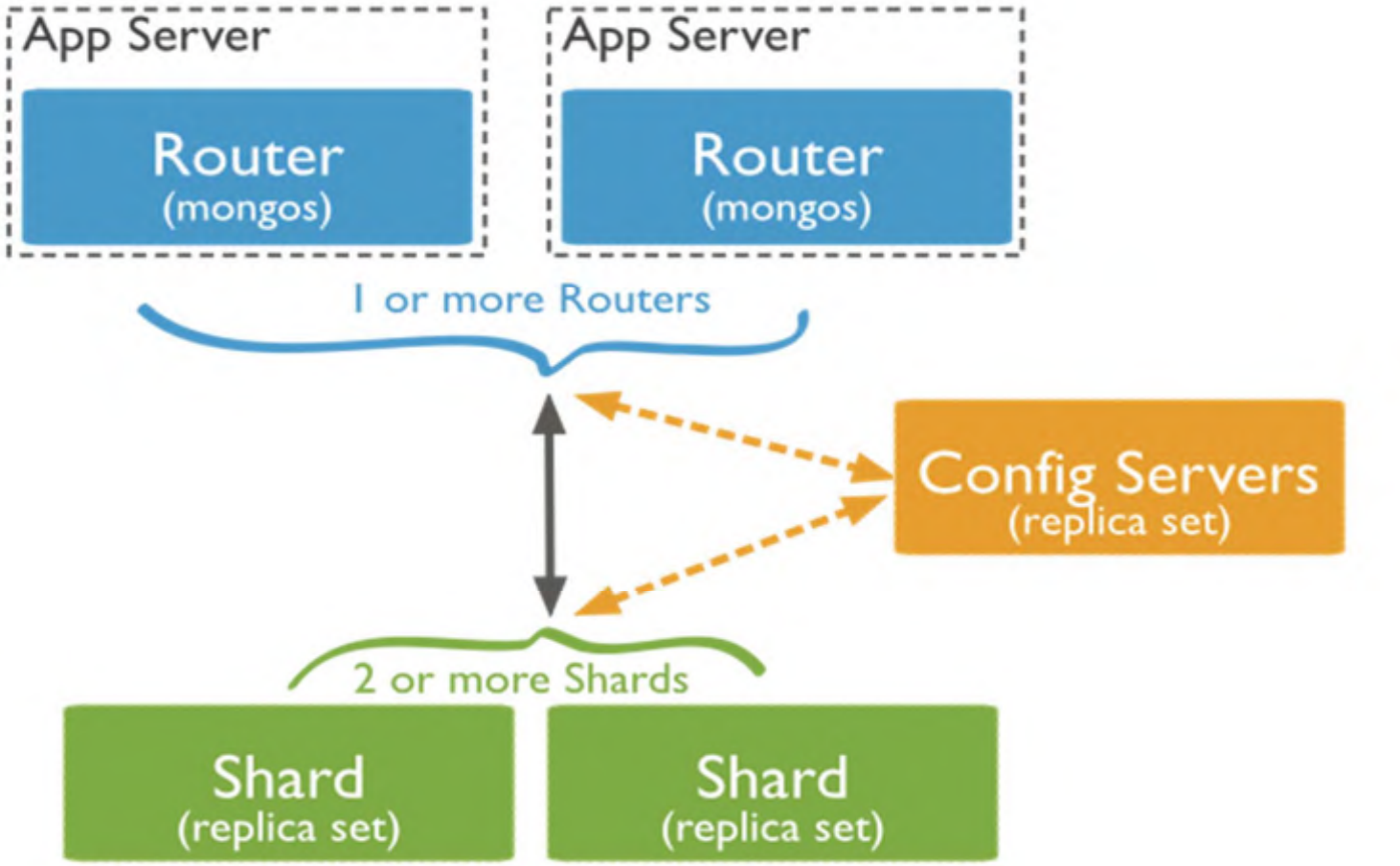
增加、删除 Shard


Client
Chunk (100, 200) ➔ Shard1
Chunk (200, 300) ➔ Shard2
Chunk (300, 400) ➔ Shard3
....
Mongos
Config Server
Shard1
Shard2
Shard3
Shard4

自动负载均衡



Mongos自动根据各Shard 存储的 Chunk 数量进行负载均衡


App Server
Router
(mongos)
App Server
Router
(mongos)
1 or more Routers
Config Servers
(replica set)
2 or more Shards
Shard
(replica set)
Shard
(replica set)

广告时间

- MongoDB中文社区 mongoing.com
- 阿里云 MongoDB 数据库目前已支持3节点复制集，分片集群即将上线。

Thanks!

Q & A