阿里云

MongoDB 分布式架构演进

张友东（林青）
zyd_com@126.com
阿里云数据库技术团队

PostgreSQL

316 systems in ranking, October 2016

Rank Oct 2016	Rank Sep 2016	Rank Oct 2015	DBMS	Database Model	Score Oct 2016	Score Sep 2016	Score Oct 2015
1.	1.	1.	Oracle	Relational DBMS	1417.10	-8.46	-49.85
2.	2.	2.	MySQL	Relational DBMS	1362.65	+8.62	+83.69
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1214.18	+2.62	+90.95
4.	↑ 5.	4.	MongoDB	Document store	318.80	+2.81	+25.54
5.	↓ 4.	5.	PostgreSQL	Relational DBMS	318.69	+2.34	+36.56
6.	6.	6.	DB2	Relational DBMS	180.56	-0.62	-26.25
7.	7.	↑ 8.	Cassandra	Wide column store	135.06	+4.57	+6.05
8.	8.	↓ 7.	Microsoft Access	Relational DBMS	124.68	+1.36	-17.16
9.	↑ 10.	↑ 10.	Redis	Key-value store	109.54	+1.75	+10.75
10.	↓ 9.	↓ 9.	SQLite	Relational DBMS	108.57	-0.05	+5.90

Mongo

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

mongoDB

MongoDB是什么？

MongoDB是一个 开源 OLTP 数据库。

它灵活的 文档 模型非常适合 敏捷式地 开发

高可用 和 水平扩展 的大数据应用。

MongoDB 特性

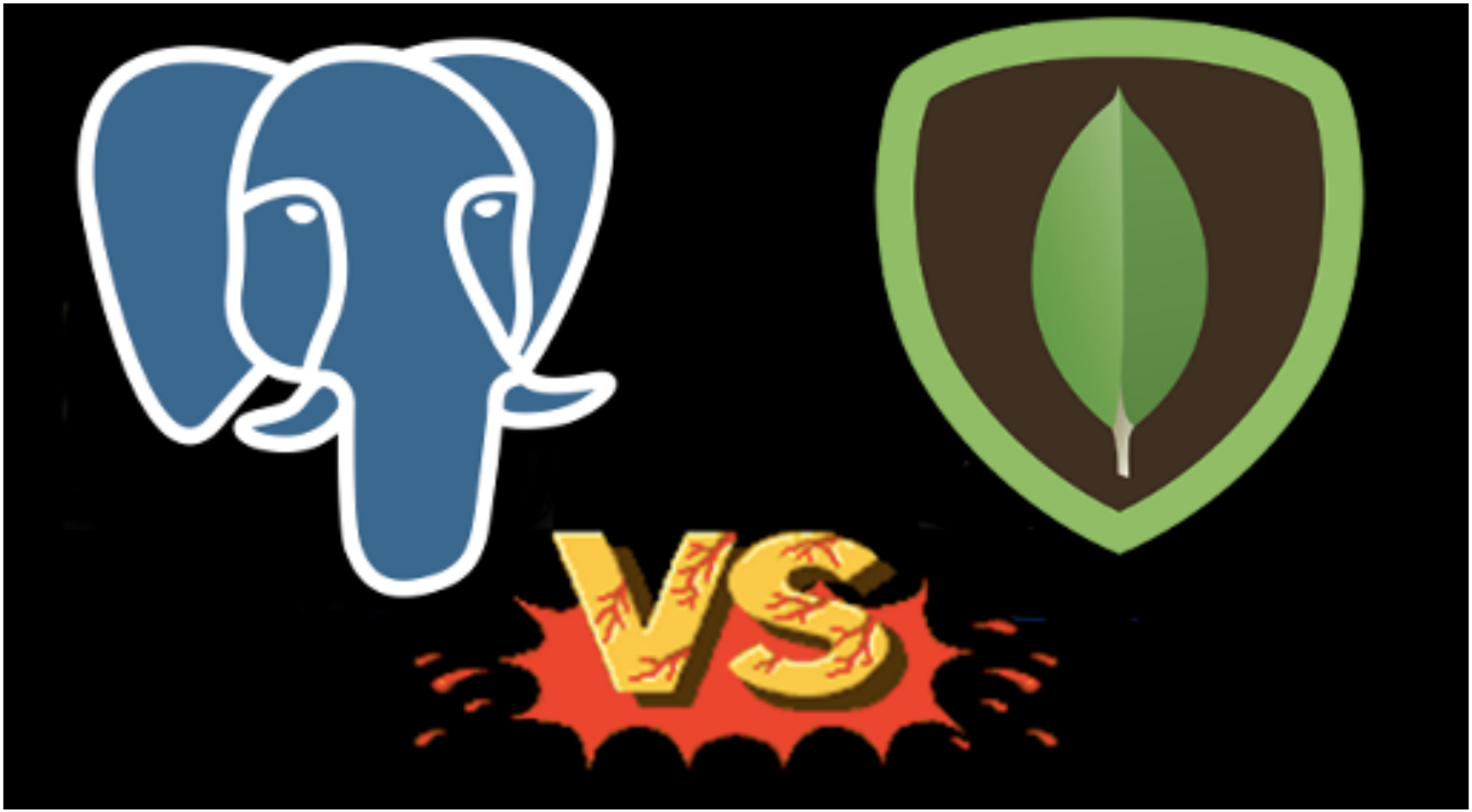
- 核心优势
  - 灵活文档模型 + 高可用复制集 + 可扩展分片集群
- 功能特点
  - 二级索引、地理位置索引、全文索引
  - aggregate、map-reduce
  - GridFS 支持文件存储
- 不足之处
  - 不支持事务、仅支持简单 left join

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

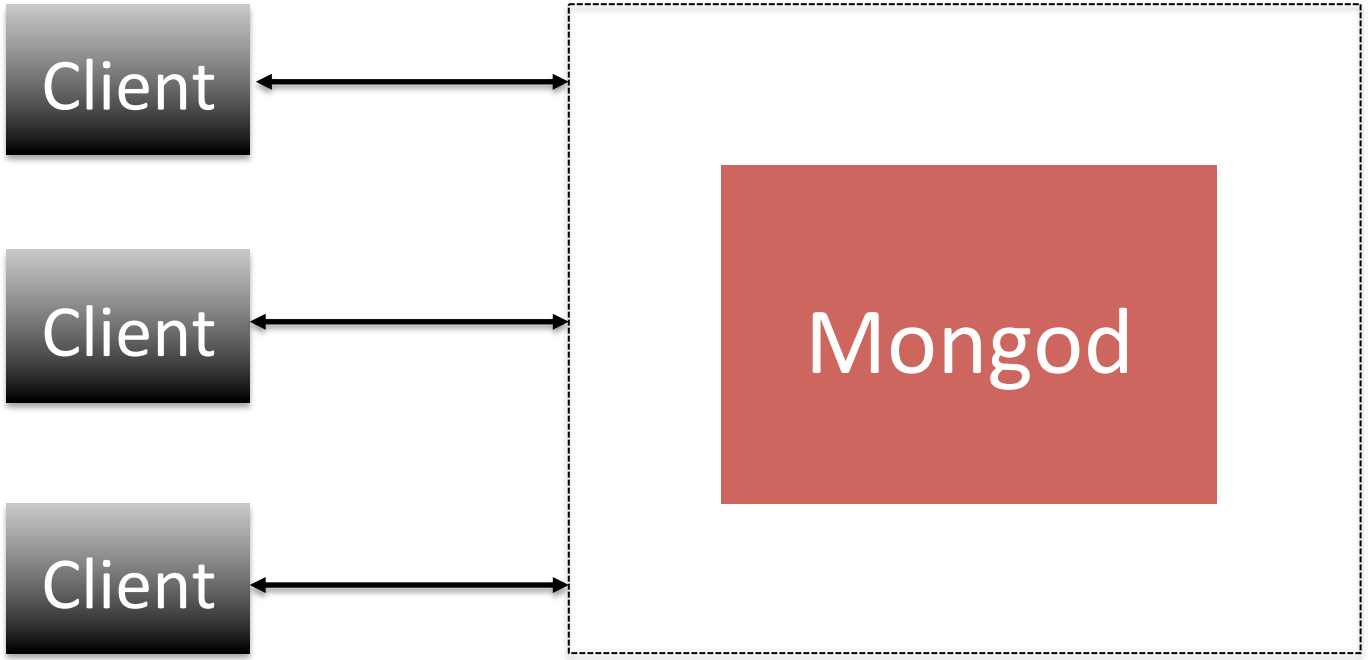
今天不谈文档模型



主要内容

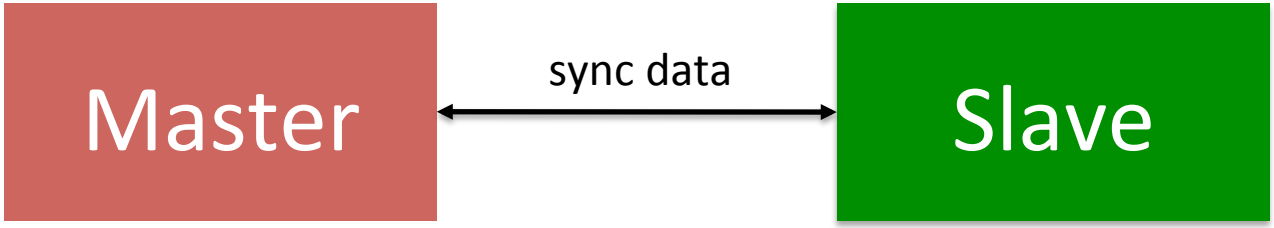
- 如何保证数据高可靠？
- 如何保证服务高可用？
- 如何实现水平扩展？

单节点



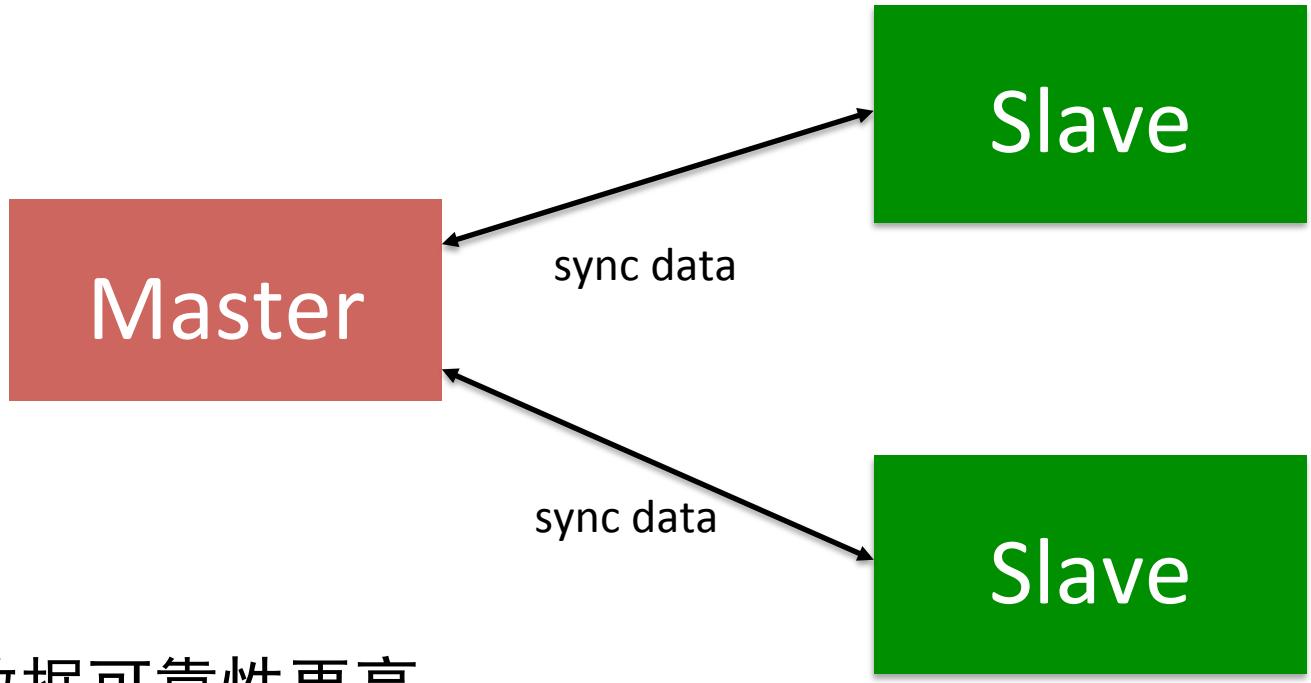
- 数据单点
- 服务单点

主备节点



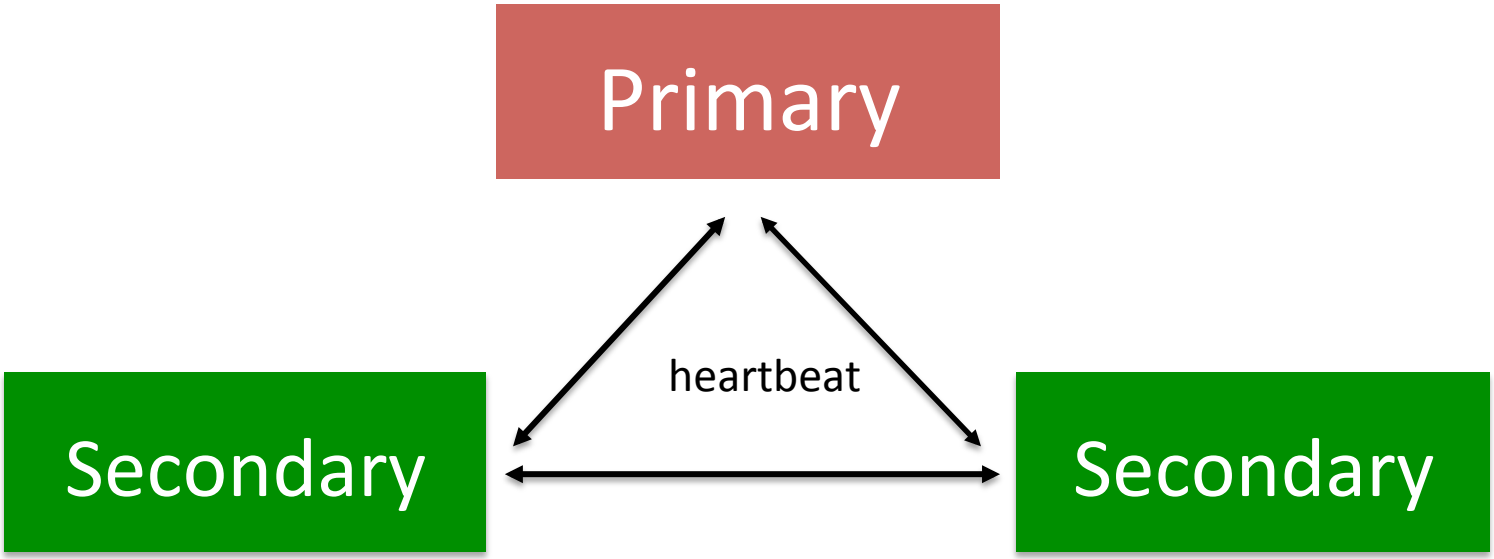
- Master 宕机无法服务写请求
- 只能容忍一个节点失效

一主多备



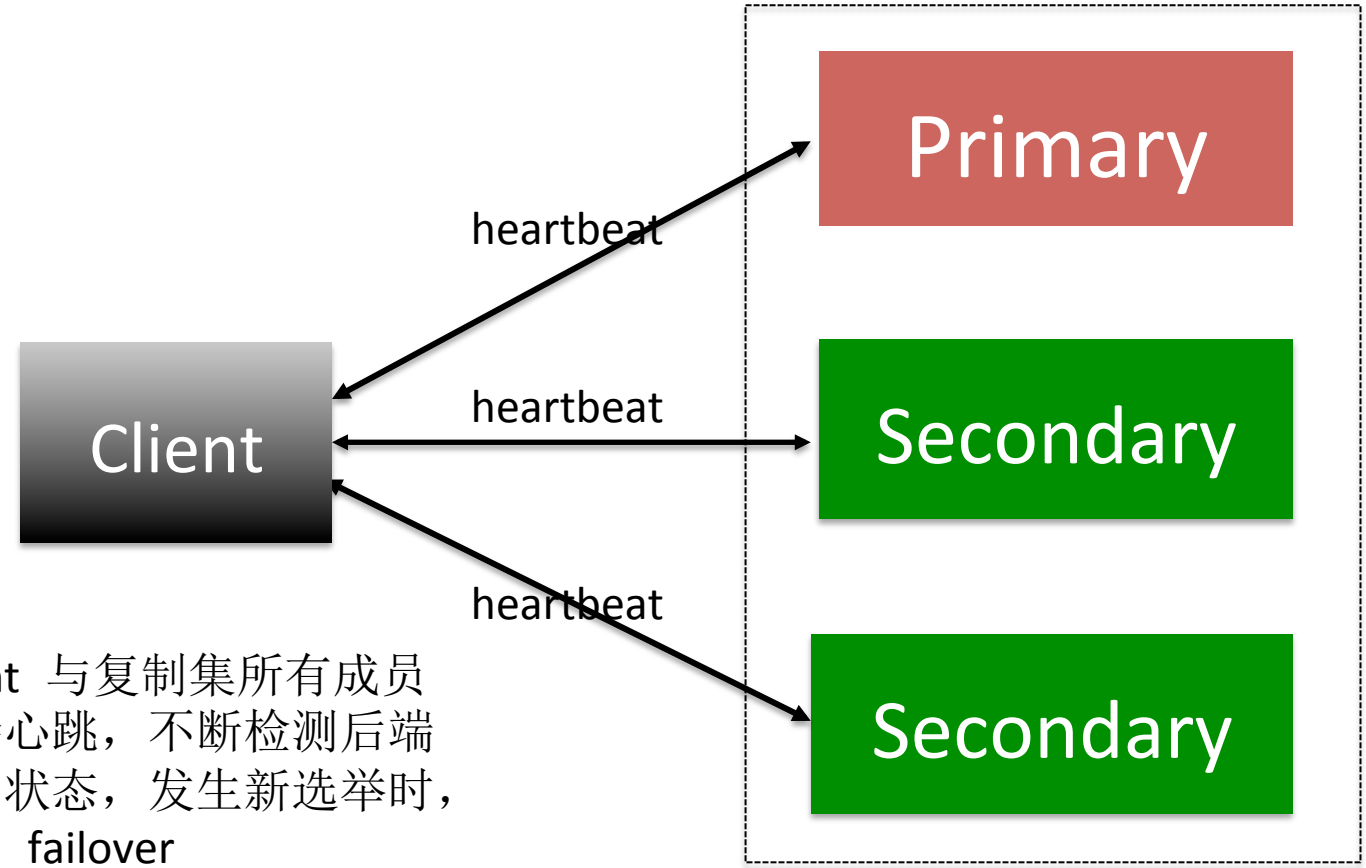
- 数据可靠性更高
- 扩展读服务能力

复制集



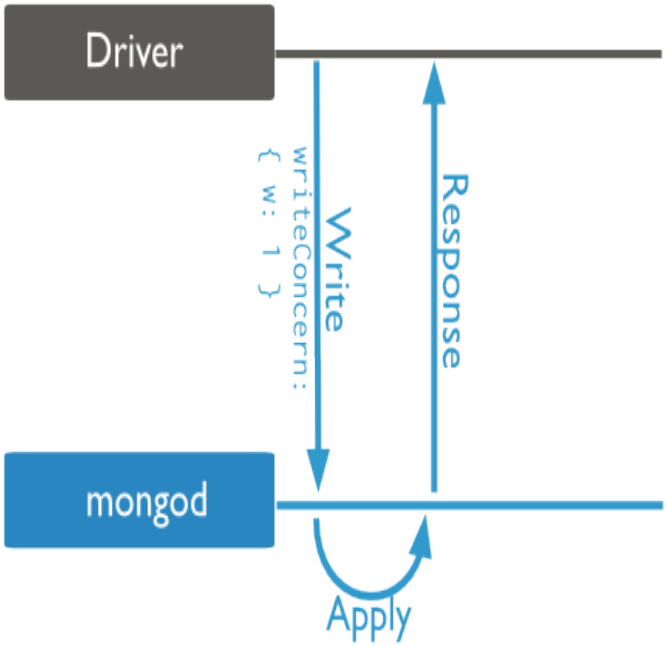
- 通过 raft 协议选举出 Primary
- 所有写请求都写到 Primary，并同步到 Secondary
- 当 Primary 故障时，自动选出新的 Primary 节点

高可用

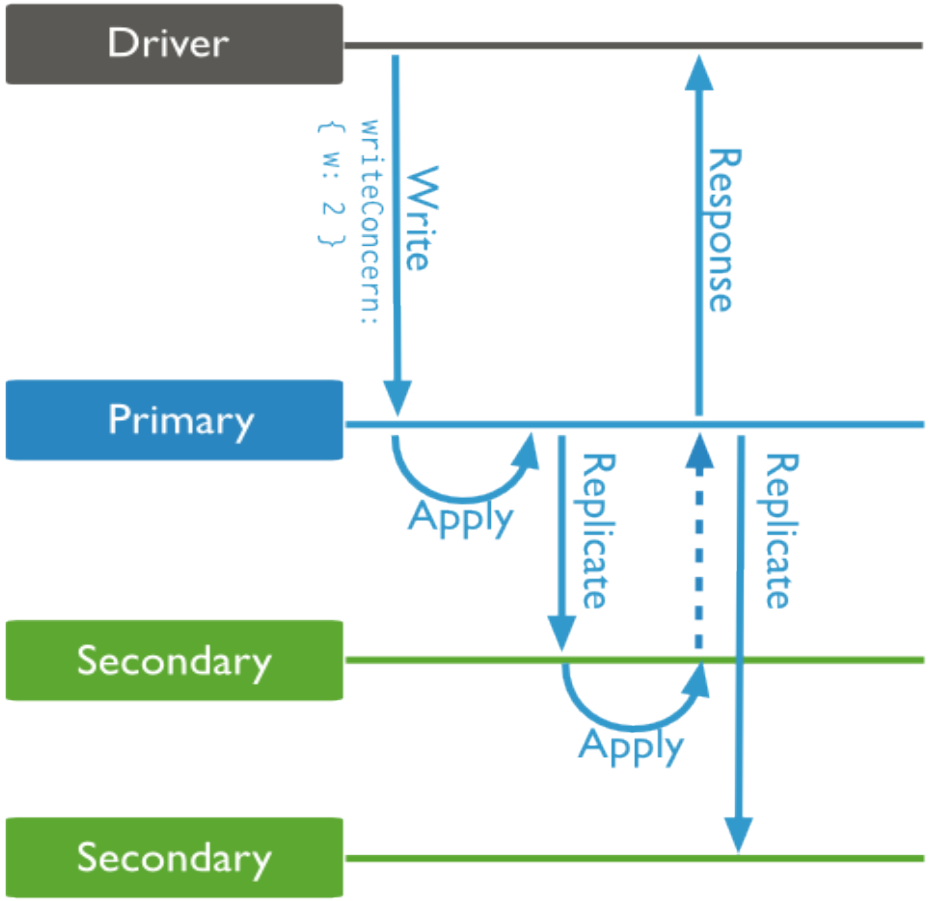


Client 与复制集所有成员保持心跳，不断检测后端节点状态，发生新选举时，自动 failover

写策略



WriteConcern: {w: 1}



WriteConcern: {w: 2}

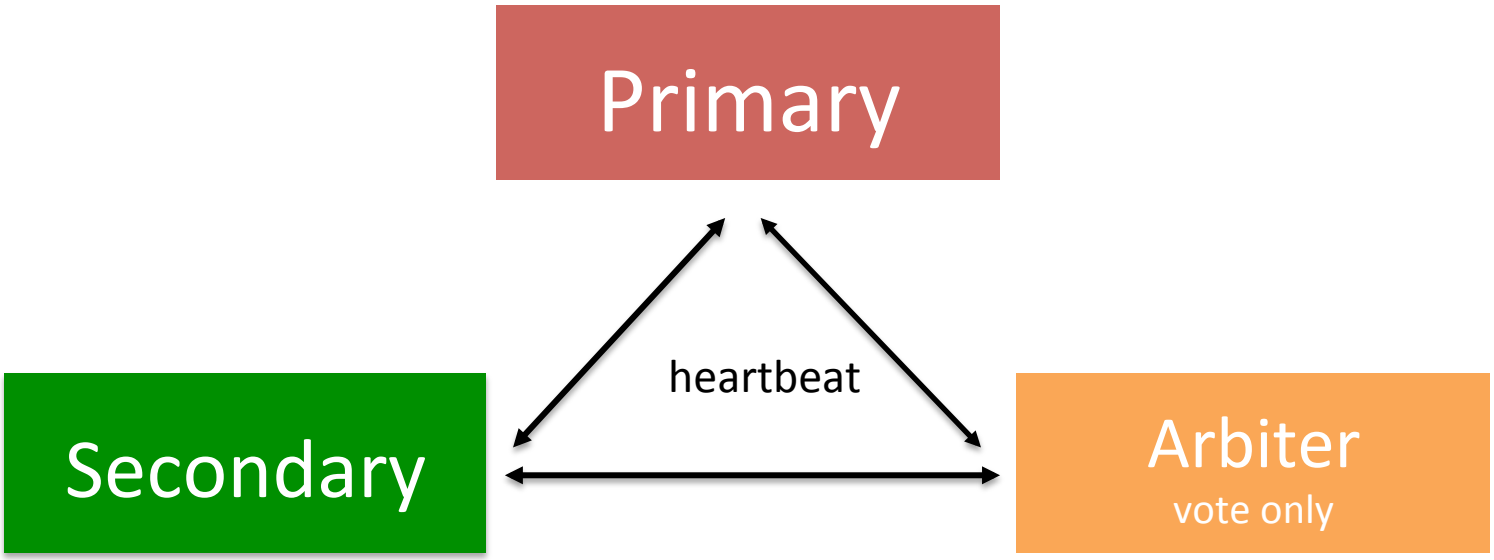
选举规则

https://raft.github.io/

节点数	大多数
1	1
2	2
3	2
4	3
5	3
6	4
7	4

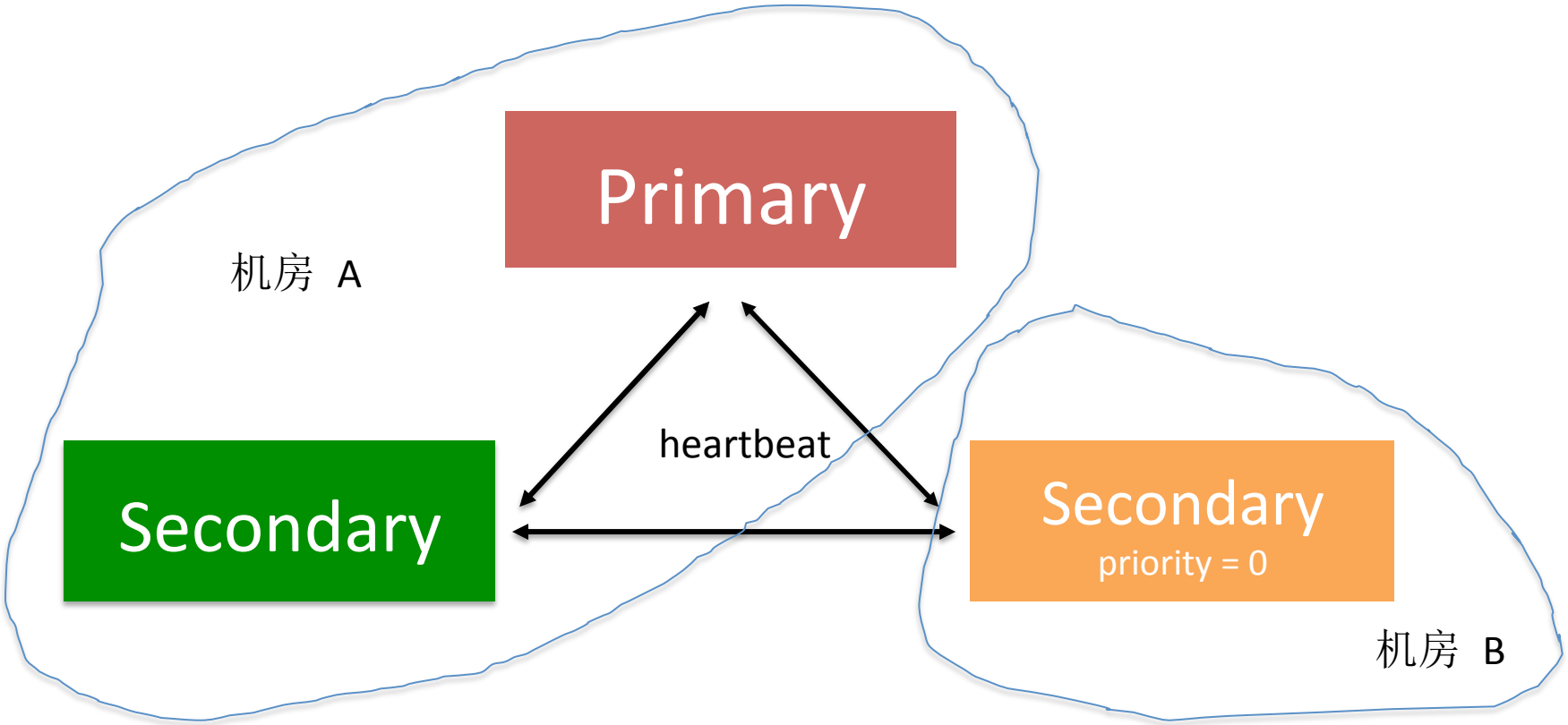
- 建议复制集部署『奇数』个节点

仲裁节点



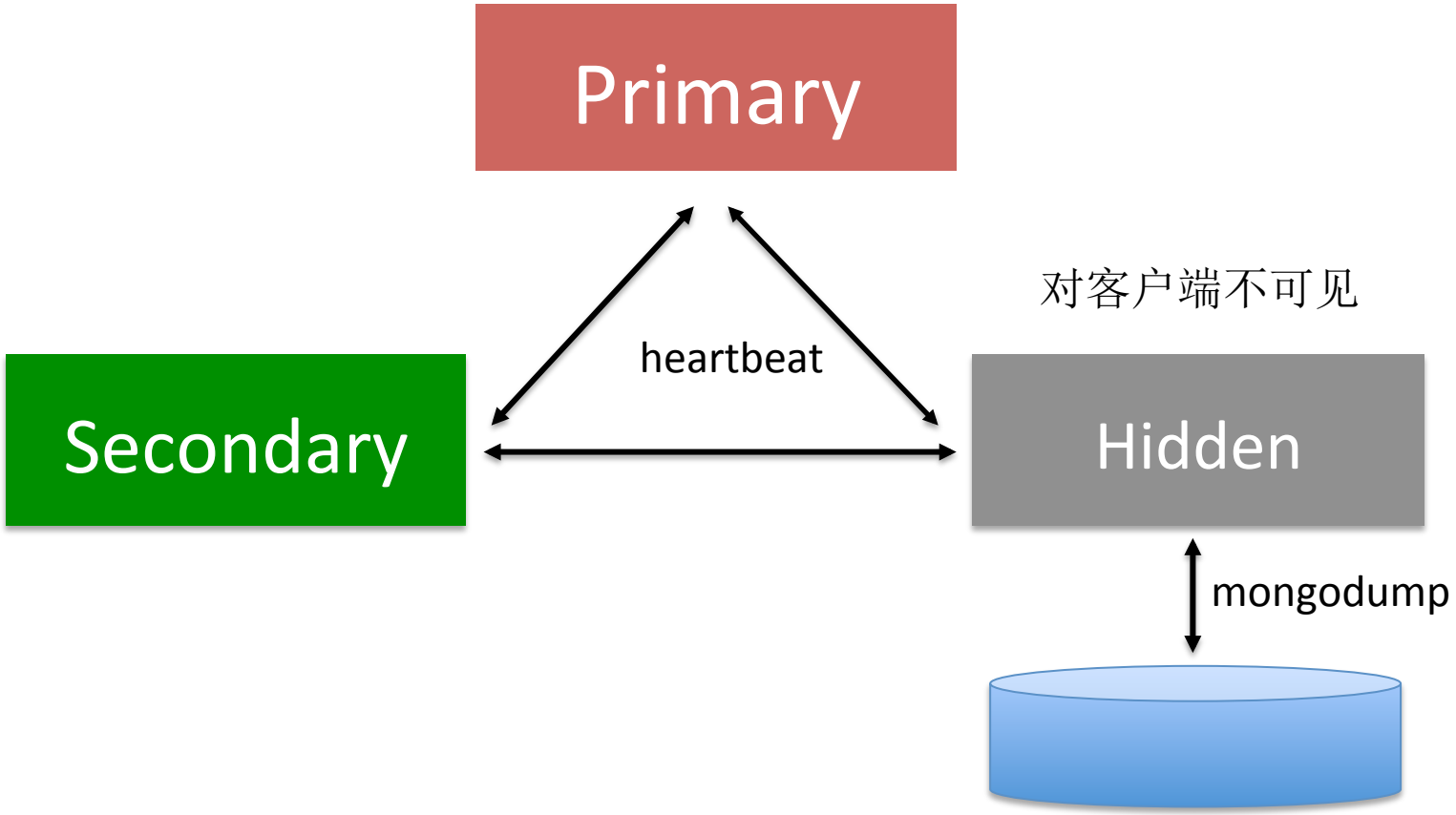
Arbiter 节点只参与投票，不存储数据

选举优先级

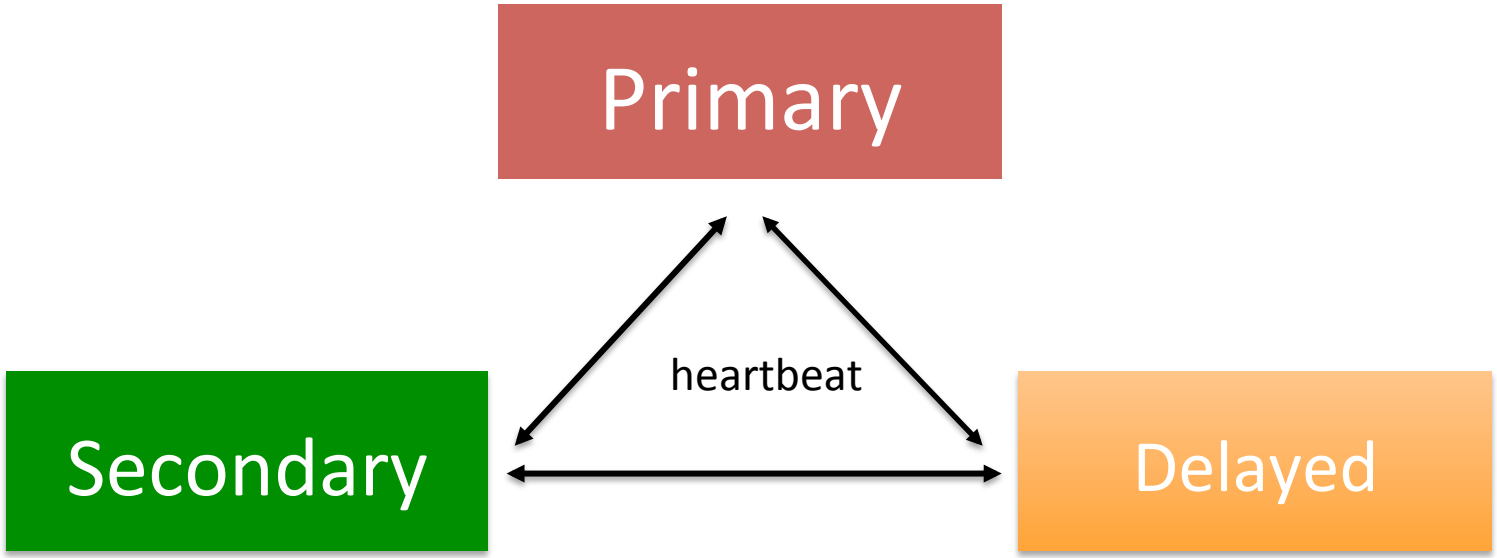


机房 B 的节点优先级为0，不会被选为主

隐藏节点

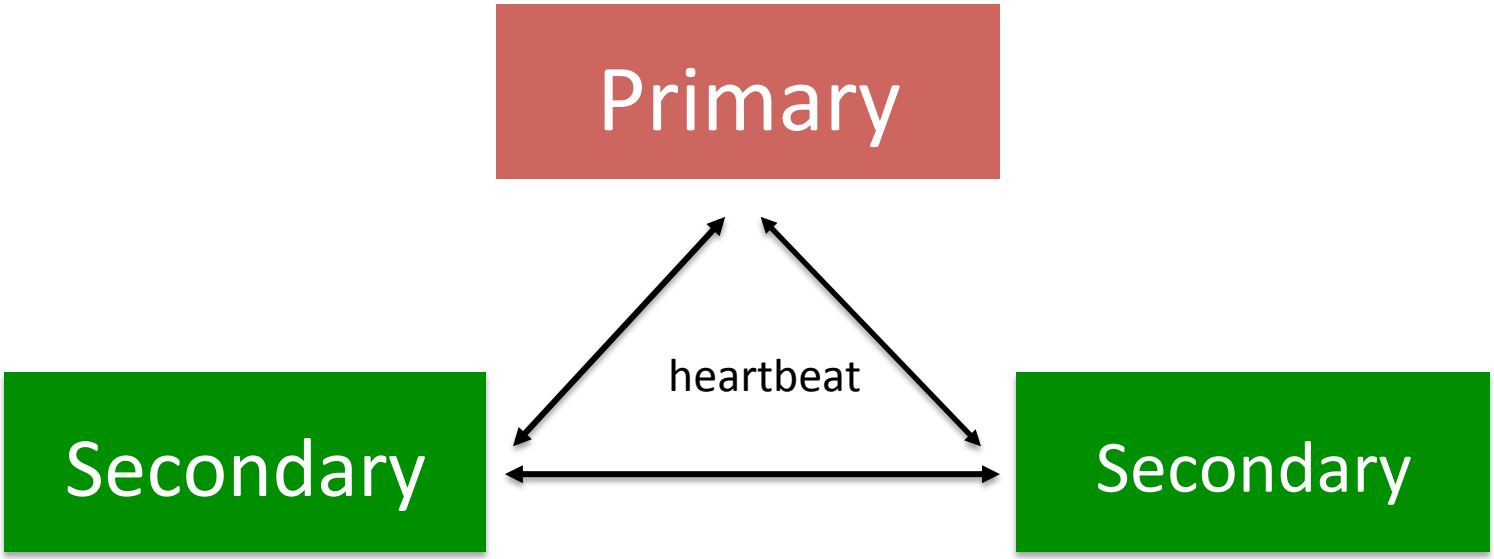


延迟节点



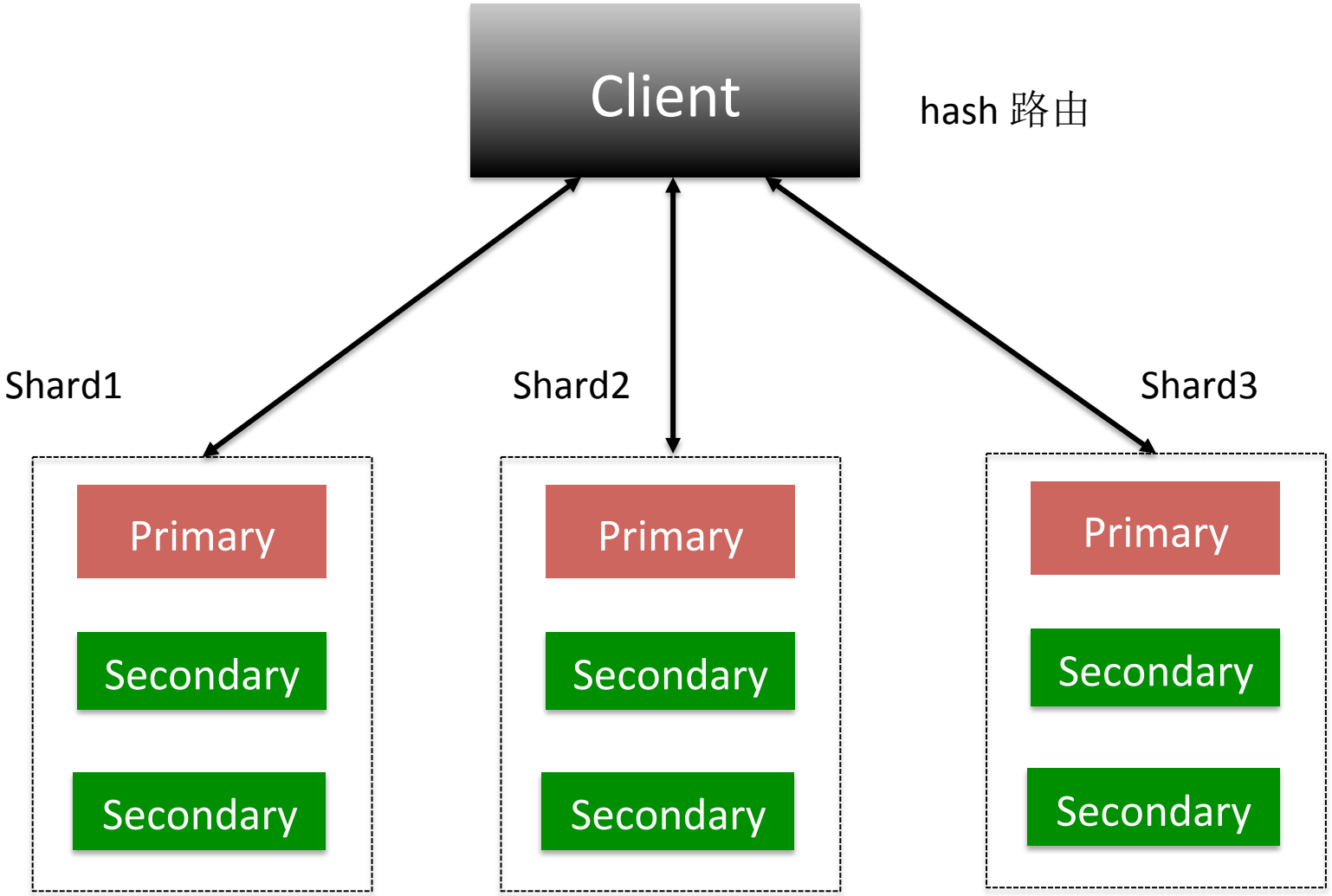
数据落后其他节点一段固定时间，可用于数据回滚恢复

复制集的问题

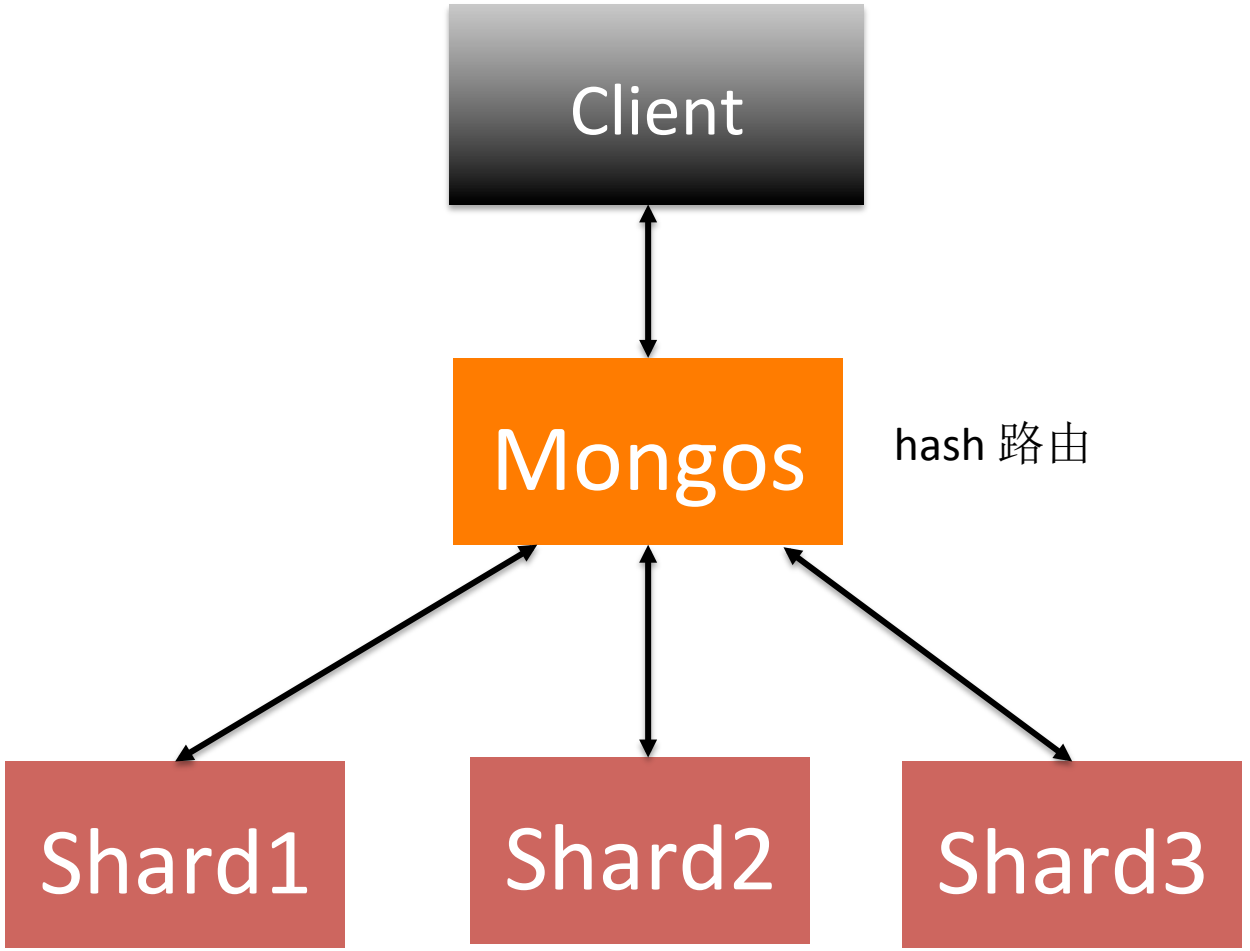


- 存储容量受限于单个 Primary
- 写服务能力受限于单个 Primary

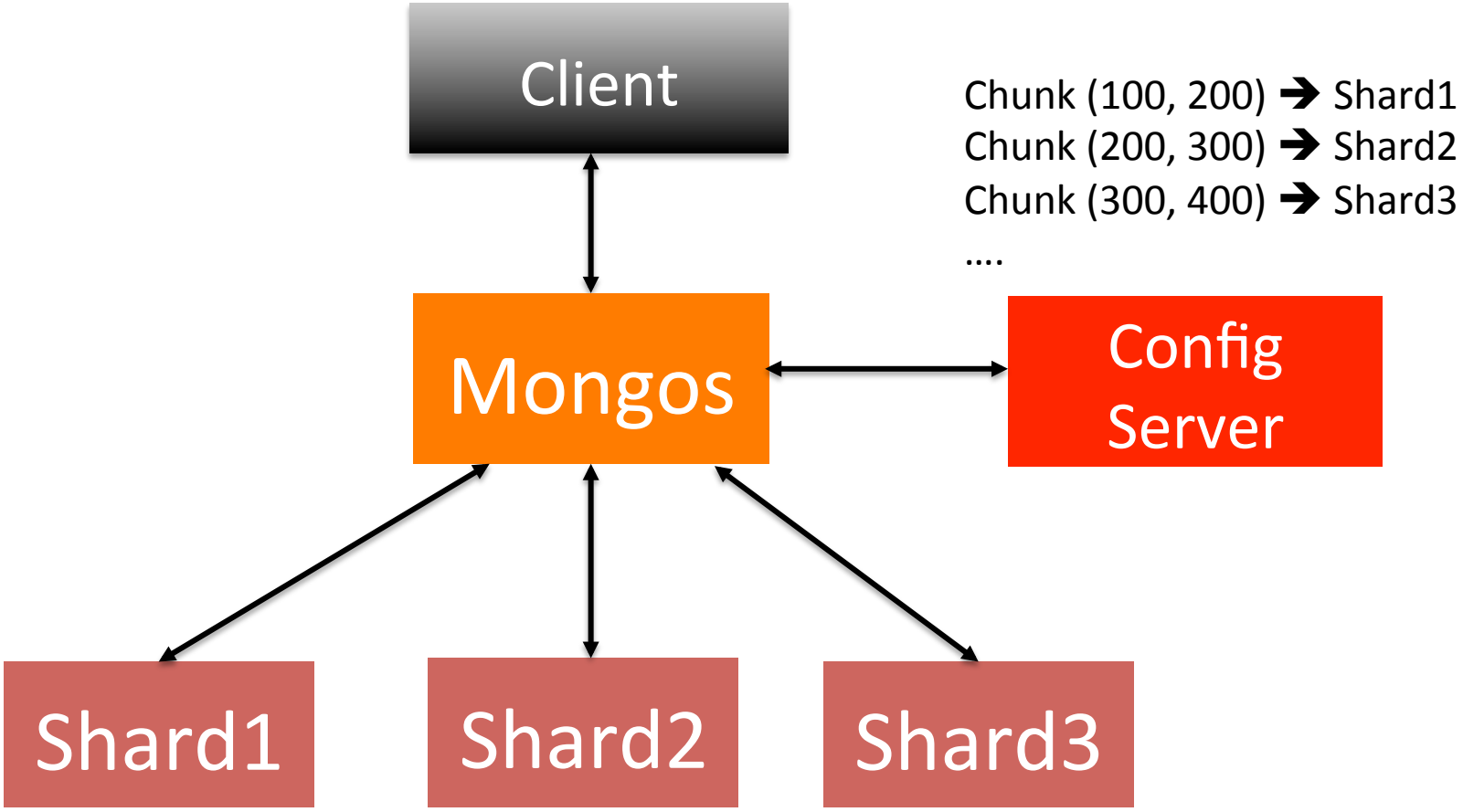
客户端分片


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

Proxy分片



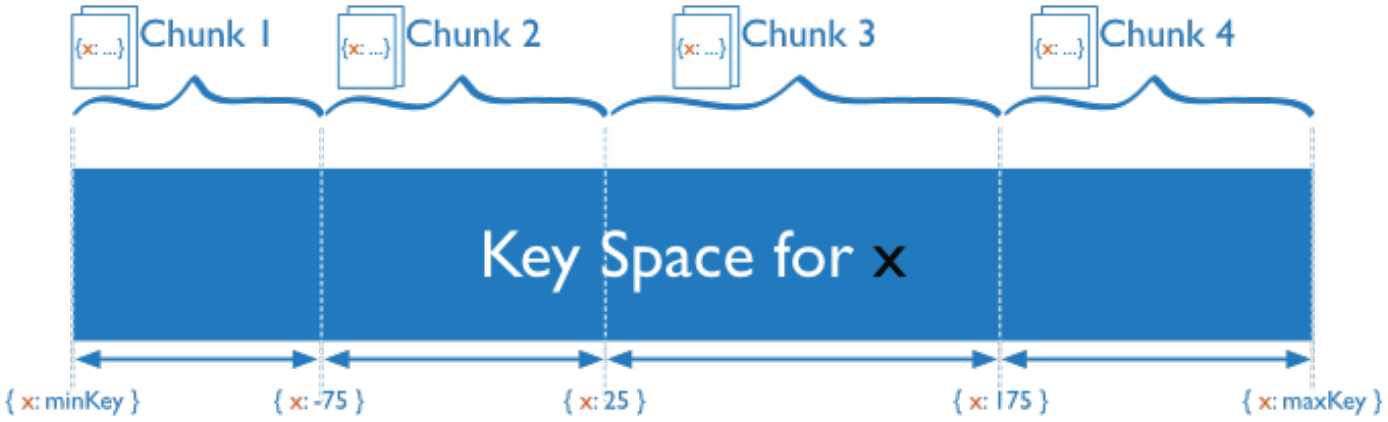
可扩展分片


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

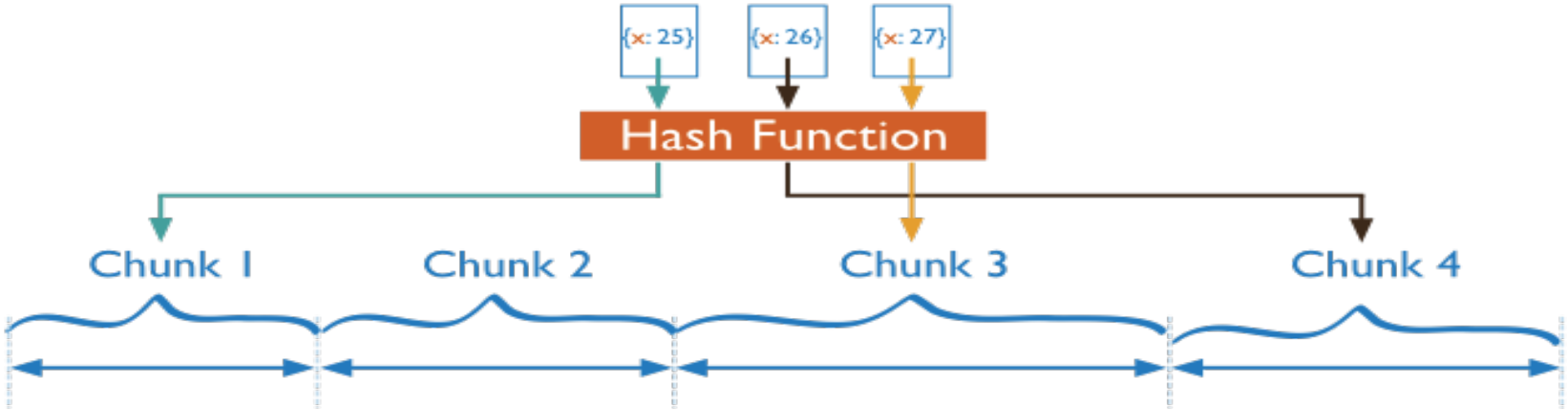
分片方式-范围

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

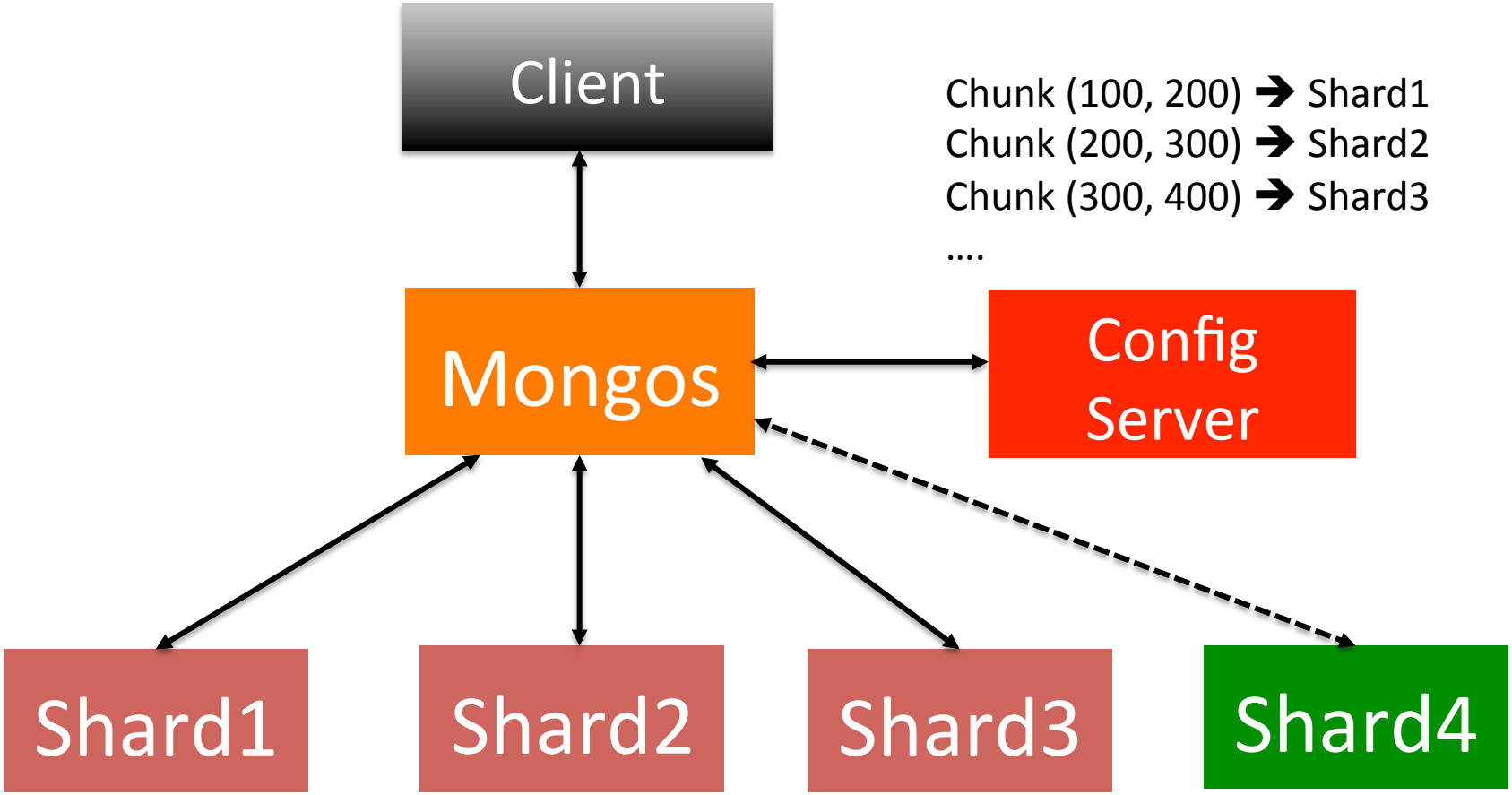


分片方式-hash

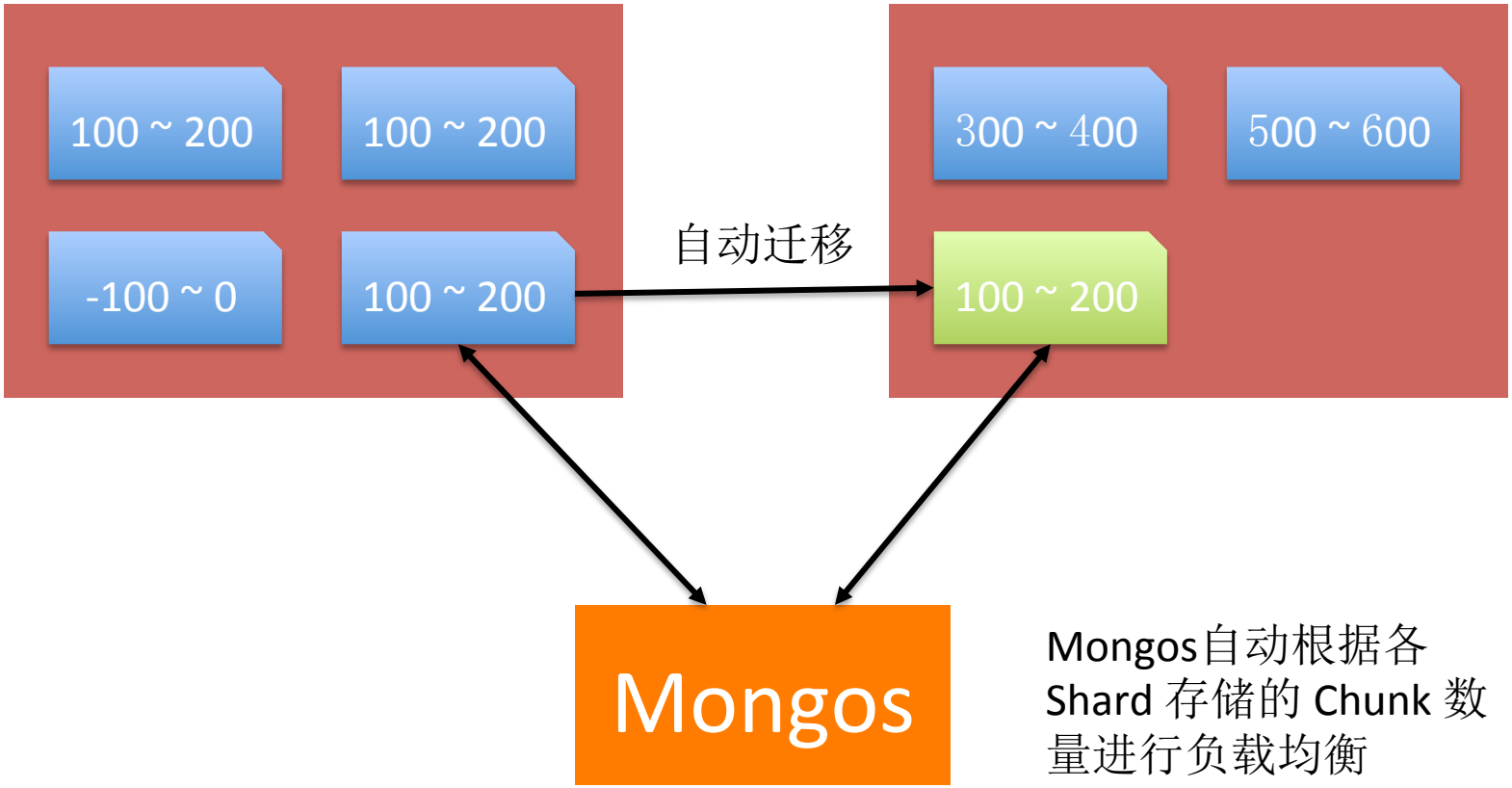


- 根据某个字段的hash值，顺序划分为多个范围，每个范围对应一个 Shard，能将数据均匀的分布到各个 Shard

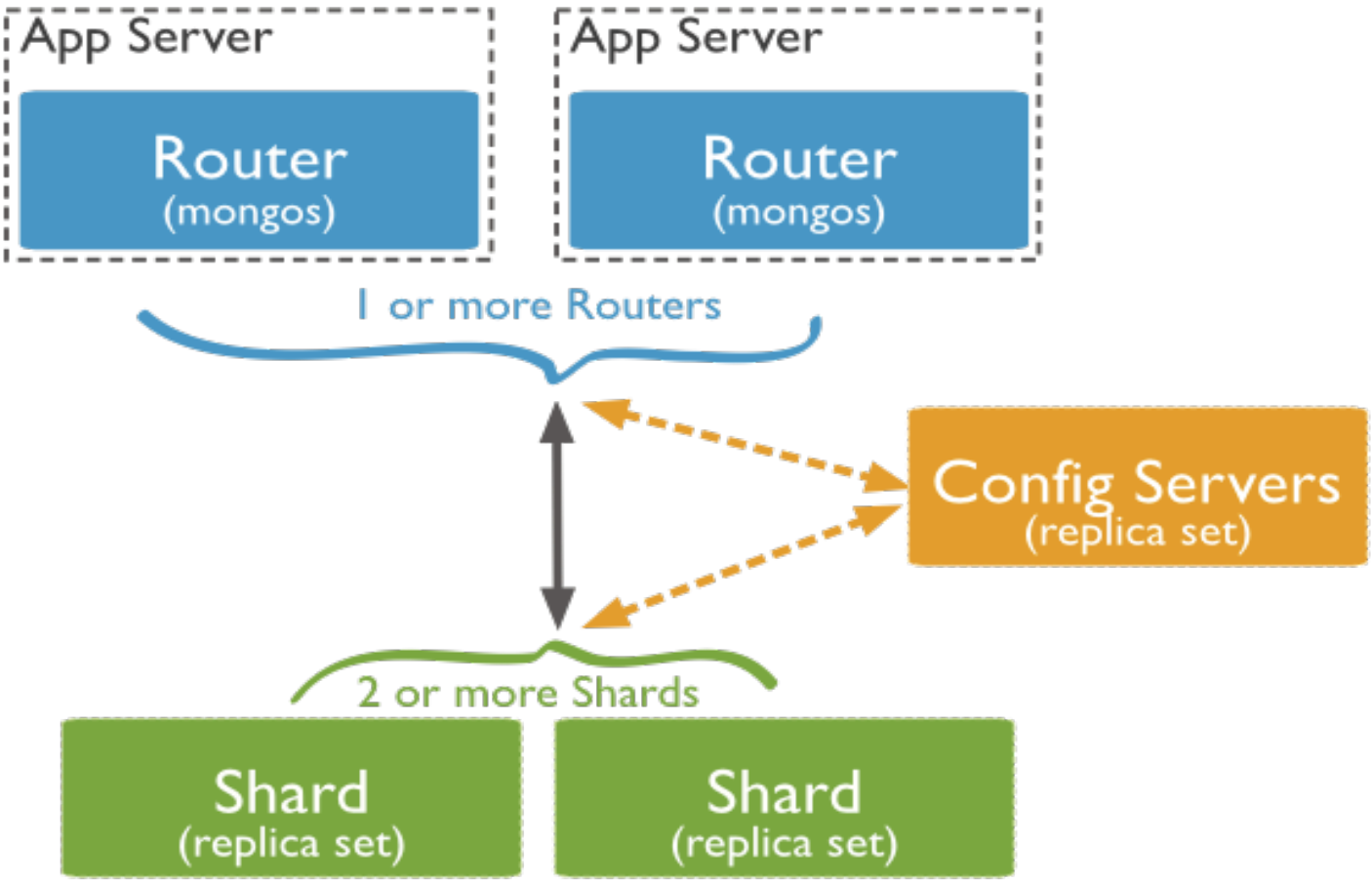
增加、删除 Shard


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)

MongoDB选项参考

应用特征	Yes/No?
应用不需要事务支持及复杂 join	必须 Yes
新应用，需求会变，数据模型无法确定，想快速迭代开发	?
应用需要2000-3000以上的读写QPS	?
应用需要TB甚至 PB 级别数据存储	?
应用发展迅速，需要能快速水平扩展	?
应用要求存储的数据不丢失	?
应用需要99.999%高可用	?
需要大量的地理位置查询	?

1个yes：可以考虑MongoDB
2个及以上yes：不会后悔的选择！

- 无需事务支持
- 需求多变
- 高性能
- 海量存储
- 水平扩展
- 数据高可靠
- 服务高可用
- 强大功能

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。

https://www.aliyun.com/product/mongodb

Thanks!

Q & A