DBAplus

数据库技术沙龙-上海站

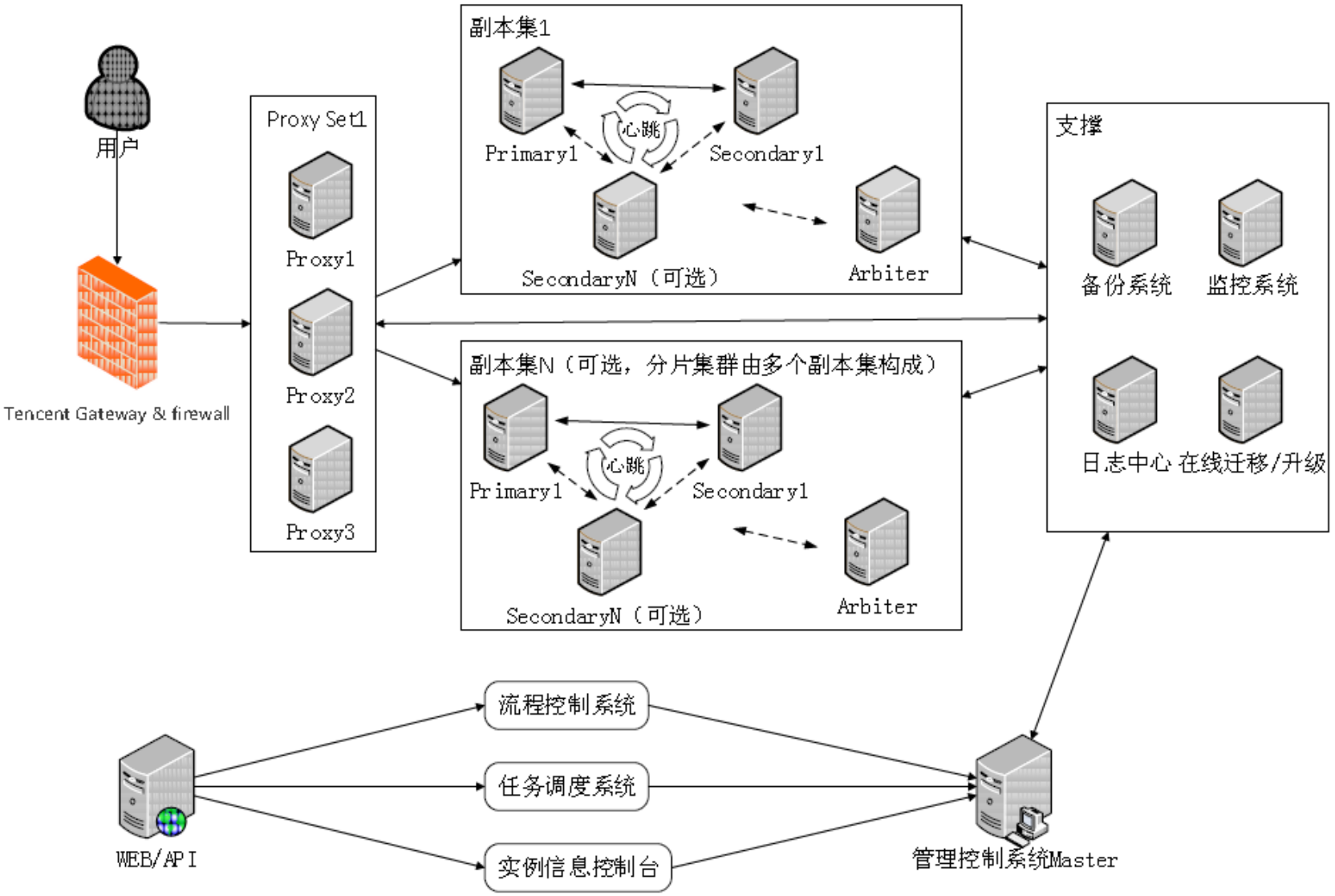
MongoDB云上服务实践

腾讯云MongoDB架构师
wolf.kong

目录

- 云MongoDB(CMongo)现有架构
- CMongo vs MongoDB
  - 搬迁与回档到任意时间点
  - 分片一致性备份与回档
- 多引擎支持
  - Mmap/WiredTiger/MongoRocks
- 运营中的问题与优化
  - Geo查询的效率低
  - 短连接场景CPU高
  - 新增节点追不上Oplog

系统架构


用户
Tencent Gateway & firewall
Proxy Set1
Proxy1
Proxy2
Proxy3
副本集1
Primary1
心跳
Secondary1
SecondaryN（可选）
Arbiter
副本集N（可选，分片集群由多个副本集构成）
Primary1
心跳
Secondary1
SecondaryN（可选）
Arbiter
支撑
备份系统
监控系统
日志中心
在线迁移/升级
流程控制系统
任务调度系统
实例信息控制台
WEB/API
管理控制系统Master

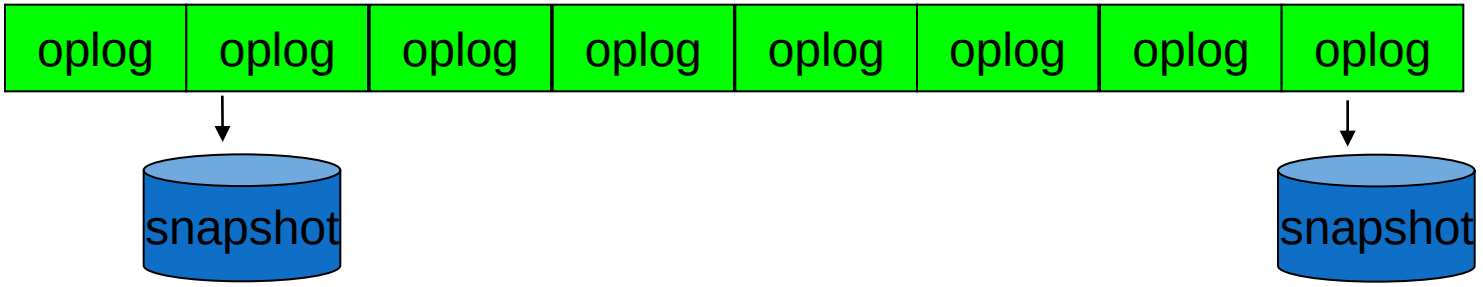
系统架构-接入层

- ✓ 自研兼容mongo协议的Proxy
- ✓ 协议转发. 解析client的请求，根据路由信息转发到对应的mongo；
- ✓ 自动加载路由信息，探测mongo的主从信息；
- ✓ 无状态，可按需扩展；

系统架构-存储层

- ✓ 同一分片内部一主多从. 从库数量可以按需添加,支持同城多机房部署容灾
- ✓ 支持两种集群模式:
  - ➢ 副本集模式. 只有一个分片,完全兼容mongo协议;
  - ➢ 分片模式:
    - ■ 数据根据shard key散列到不同的分片上面;
    - ■ 可以按需添加新分片,扩展集群能力

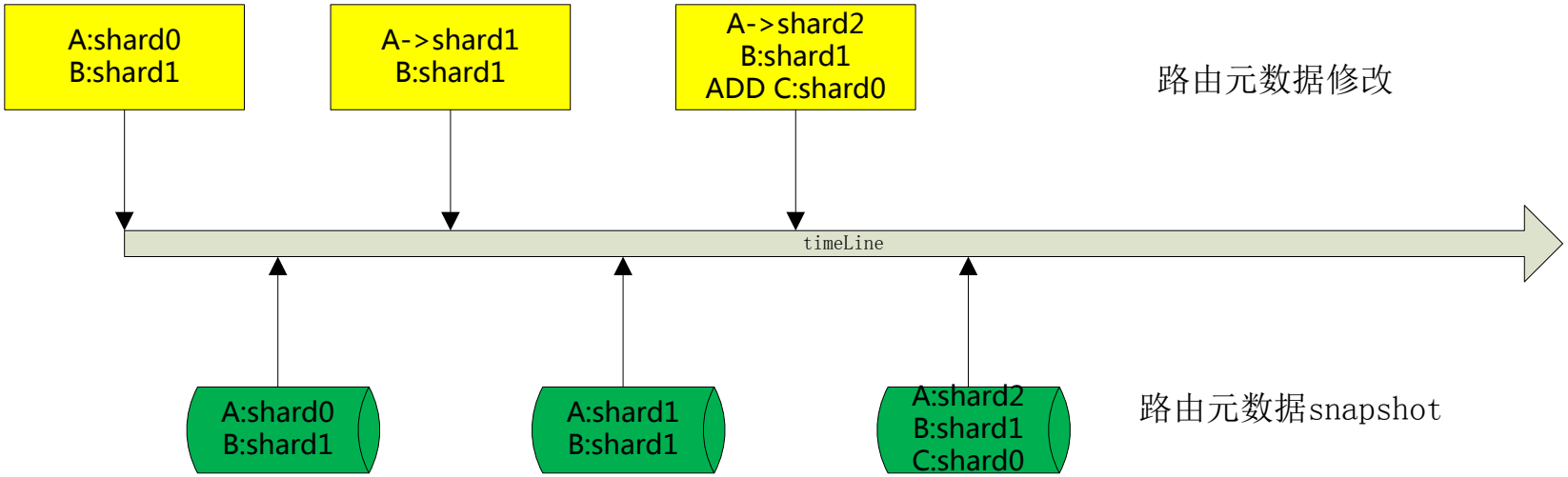
备份与回档



- ✓ oplog做持续性的备份,同时每个备份周期(7天)内对整个分片做一次全量备份(snapshot)
- ✓ 所有的备份数据以多副本(默认2)存储在hadoop
- ✓ 可以基于一个snapshot,重放后续的oplog,将整个分片的数据回档到任意时间点(7天内)

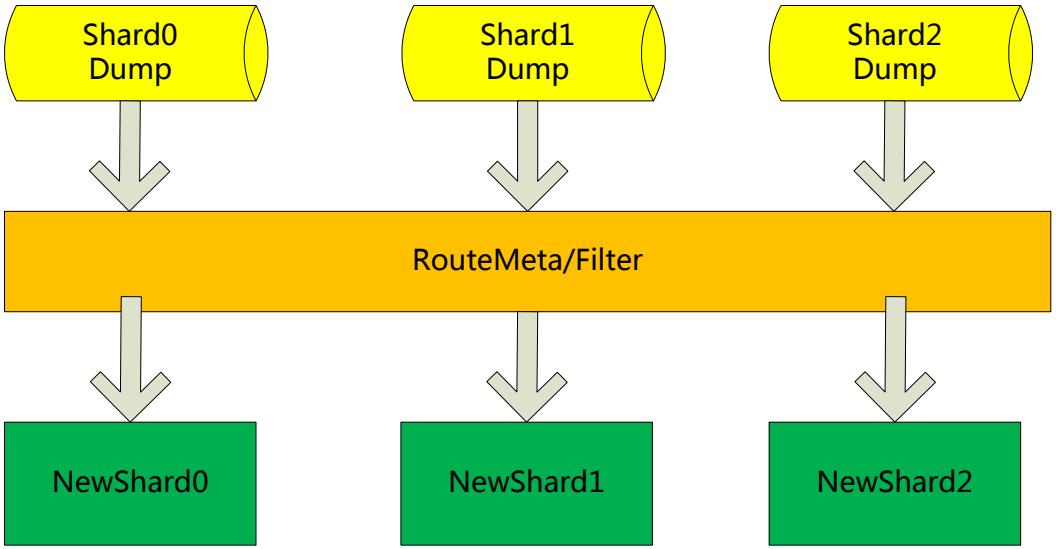
备份与回档

- 分片一致性备份与回档
  - 多分片集群存在路由变更的过程
  - 元数据增量merge成元数据的snapshot



备份与回档

- 回档时回放路由元数据更改,生成RouteMeta
- 回档时根据RouteMeta过滤脏数据(迁移时未删除的数据)

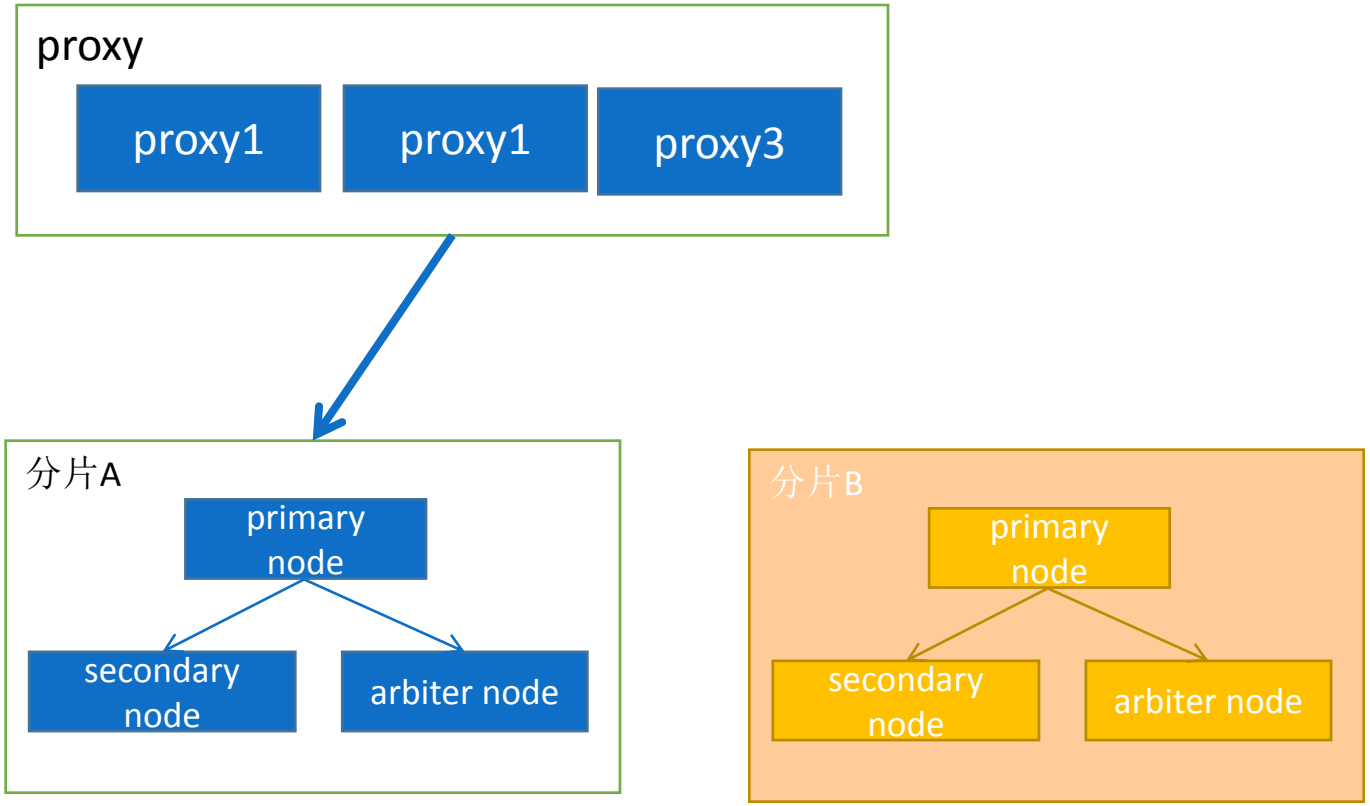


水平扩展

- 分片集群容量不够,需要添加新的分片,同时进行集群内数据搬迁,均衡容量和流量,主要步骤:
  - 基于备份的数据导入;
  - 增量数据oplog同步;
  - 路由切换;
  - 删除迁移残留数据;

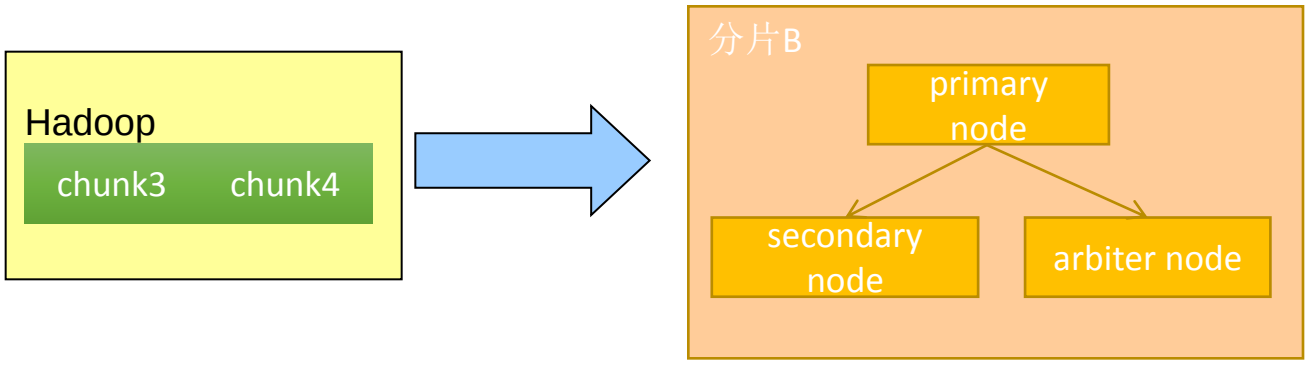
水平扩展-示例

目标: 分片A 上表db.t1有4个chunk(chunk1,chunk2,chunk3,chunk4), 需要迁移chunk3, chunk4到分片B



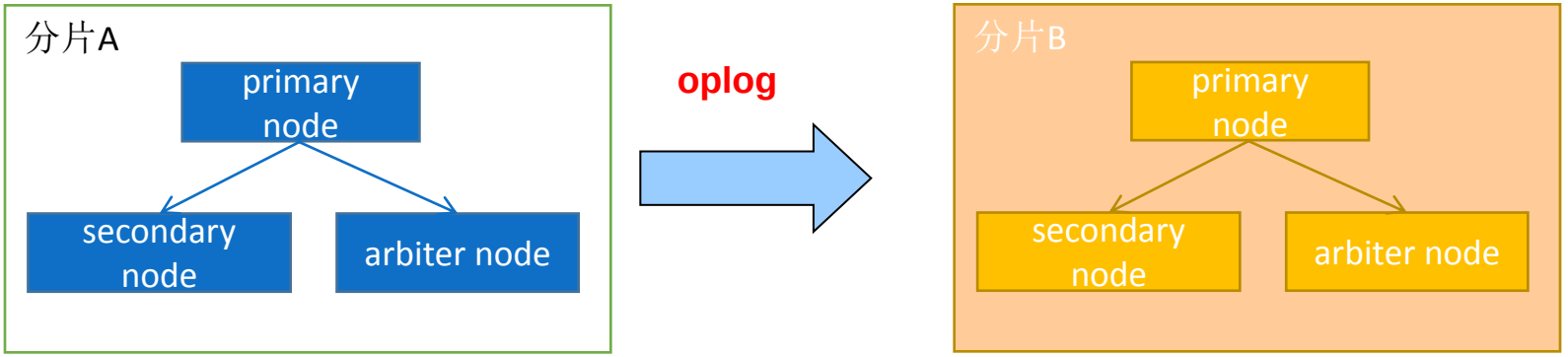
水平扩展-备份数据导入

- ✓ 备份数据多worker并发导入目标分片;
- ✓ 备份数据导入完成后会记录最后一条oplog的时间戳,用于后续增量数据oplog同步;
- ✓ 基于备份数据进行导入的好处是:
  - ➢ 备份数据是现成的, 无需再从源分片上面去dump数据(非常耗时)
  - ➢ 对源分片无任何影响



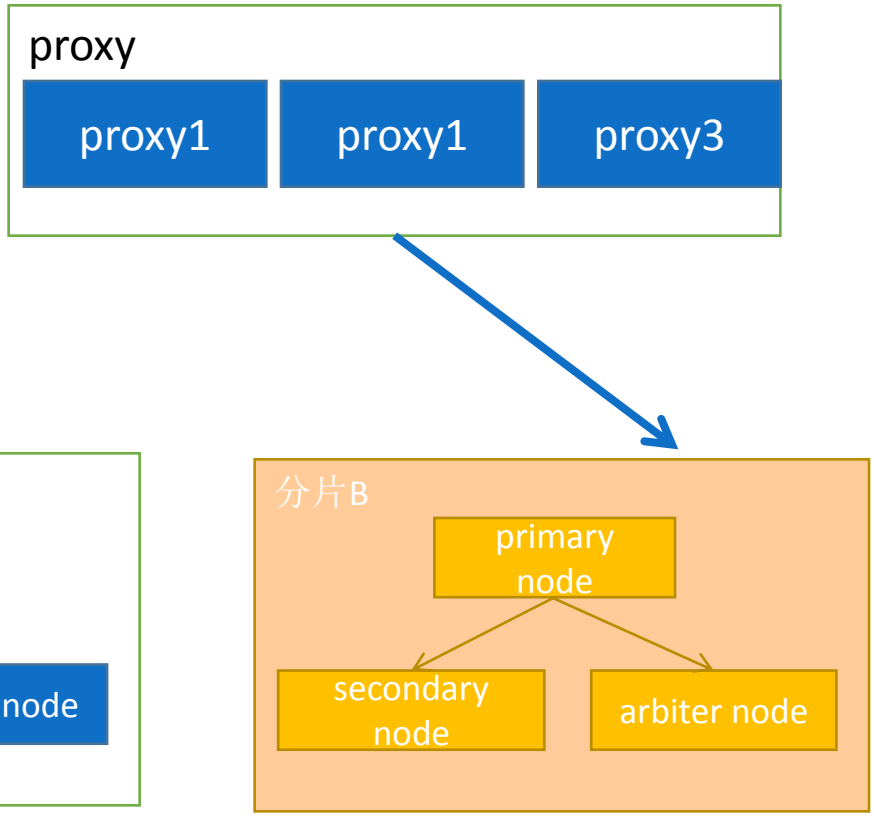
水平扩展-增量数据同步

- ✓ 从备份导入的最后时间戳开始同步源分片的oplog到目标分片
- ✓ 并发多worker,根据_id字段将oplog分桶到不同的worker



水平扩展-路由切换

- ✓ 封禁源分片的待迁移chunk的写请求
- ✓ 等待上一步的增量数据同步完成
- ✓ 更新迁移的chunk的路由到新分片
- ✓ 解禁路由
- ✓ **影响迁移chunk的写,时长预计在3os左右**

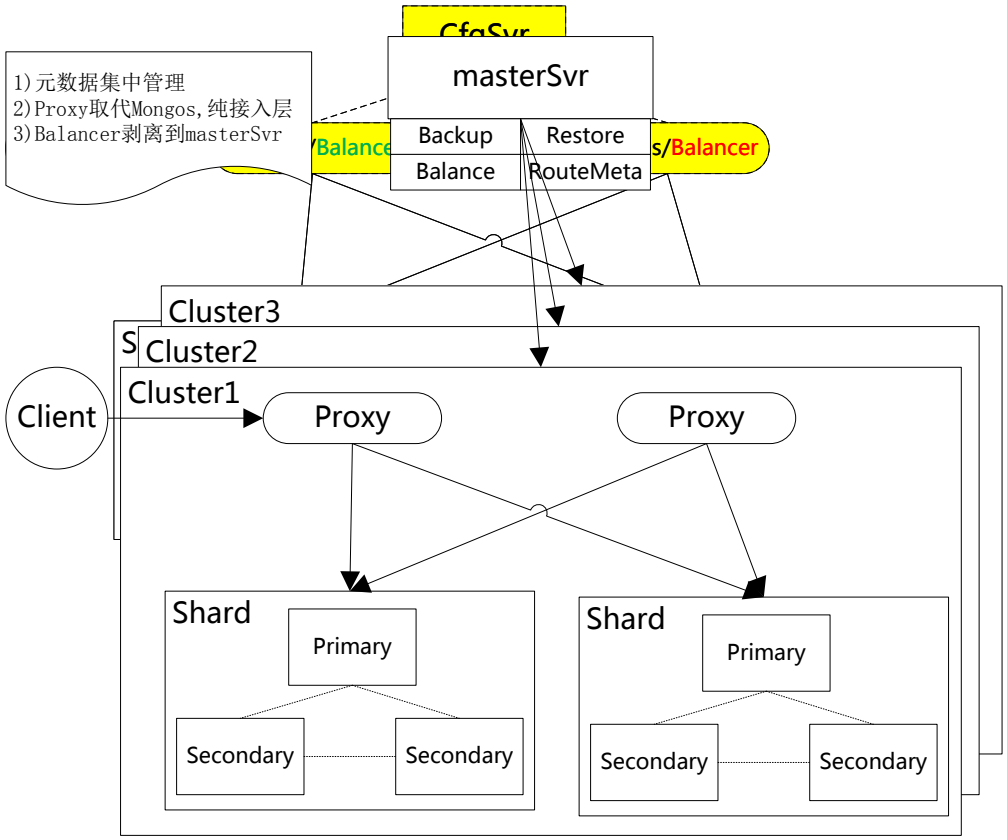


- CMongo整体架构介绍完毕
- 下面介绍CMongo对原生MongoDB的优势

MongoDB作为云服务的困难

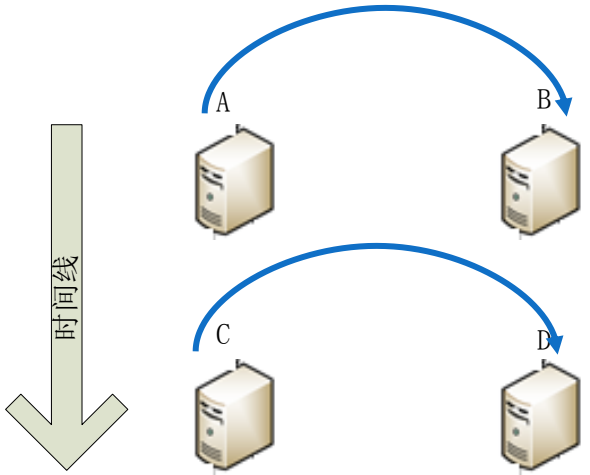
- 每个集群独立元数据
  - 额外三台MongoD组成ConfigSvr，浪费资源
  - 不方便元数据集中管理
- 迁移速度太慢
  - 10GB数据迁移需要以天为单位进行
- 多分片PIT(point in time)备份与回档
  - Balancer在均率时路由数据可能变更

元数据集中管理


1)元数据集中管理
2)Proxy取代Mongos,纯接入层
3)Balancer剥离到masterSvr
masterSvr
Backup
Restore
Balance
RouteMeta
Cluster3
Cluster2
Cluster1
Client
Proxy
Proxy
Shard
Primary
Secondary
Secondary
Shard
Primary
Secondary
Secondary

搬迁多任务并发

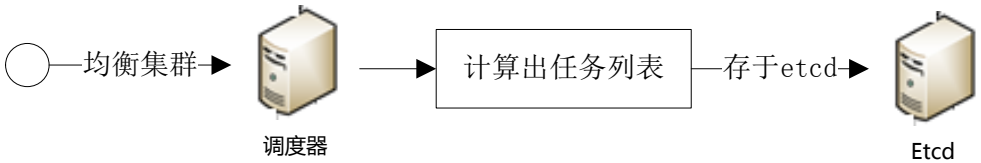
- 原生MongoDB一大劣势在于没有多任务并发



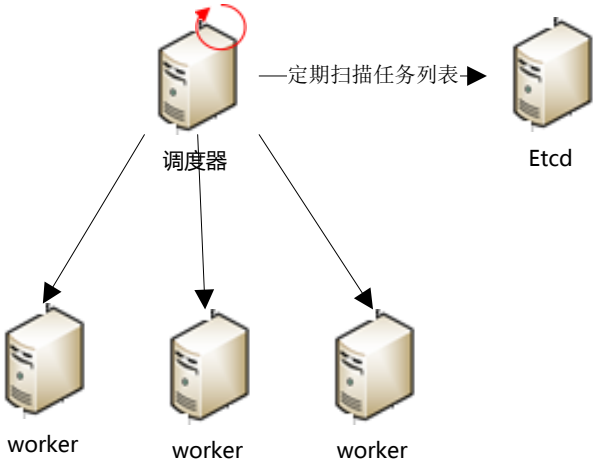
- 完全无关的两个任务依然必须要串行执行
  - 该点在3.4版本才得到改善

搬迁多任务并发

- 考虑因素：
  - 任务互斥
  - 源/目标 当前并发度
  - 节点当前利用率
- 生成任务存于etcd



- 定期扫描etcd获得任务列表，根据当前集群并发度选择任务并下发任务给worker节点



多分片PIT(point in time)备份与回档

	CMongo	MongoDB
备份	多分片一致的备份回档	单分片备份回档
		社区版无多分片一致性的备份回档功能
数据均衡	基于备份数据均衡,对源的影响小	从源读热数据,影响服务
回档	回档到七天内任意时刻 保持PIT的元数据一致性	单分片回档到任意时刻
		社区版无多分片一致性的备份回档功能

多引擎支持

- Mmap
  - 表锁,无明显优势,逐步淘汰
- WiredTiger
  - 行锁,数据压缩,B树模型,主流推广
  - 点查询/区间查询性能均优秀
  - SATA/SSD盘表现俱佳
- MongoRocks
  - 行锁,数据压缩
  - LSM+BloomFilter
  - 有空查询的点查询性能强劲
  - 适用于SSD盘

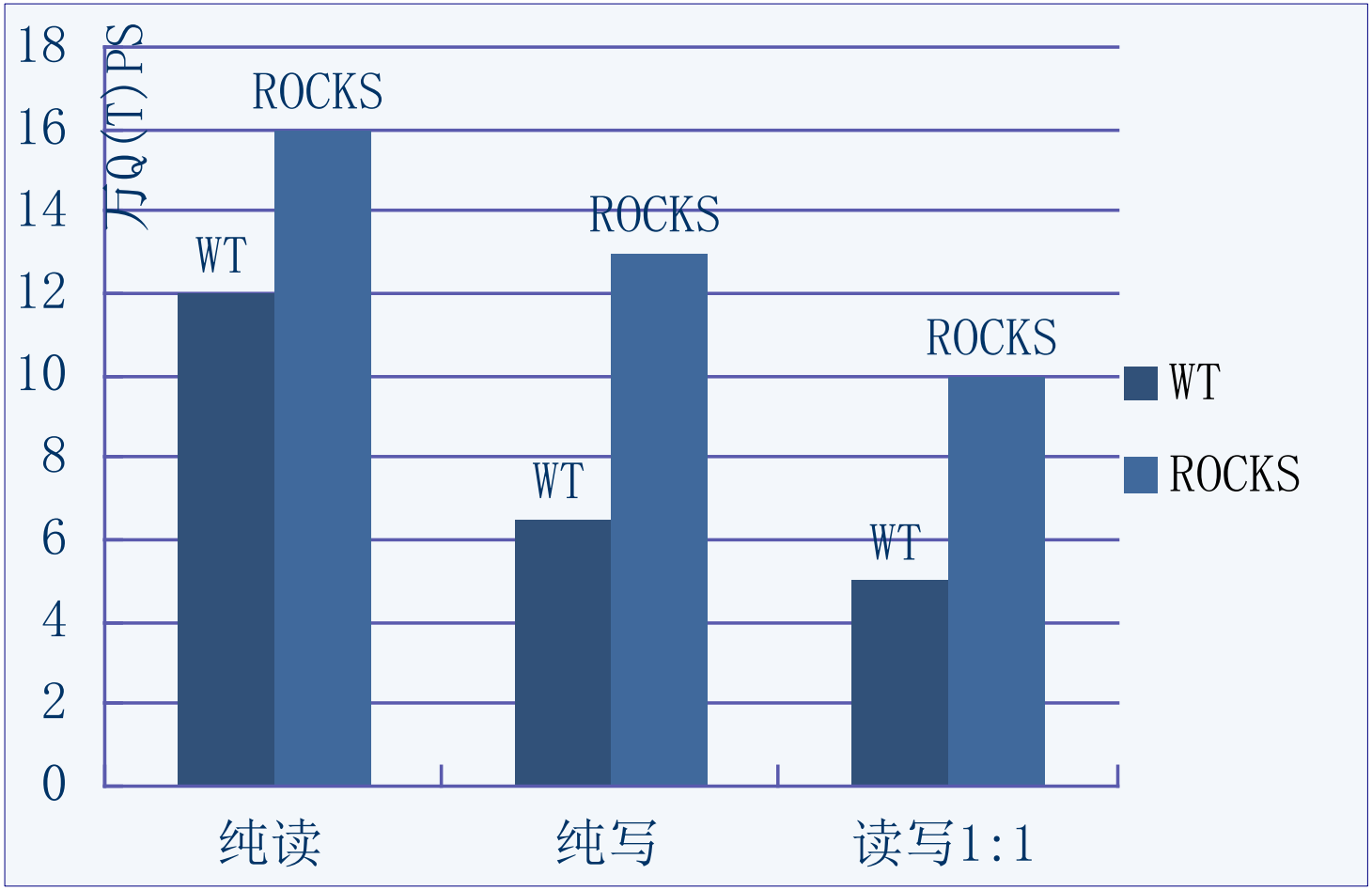
MongoRocks改造

- 单ColumnFamily->多ColumnFamily
  - CF资源隔离/memtable/compactionThreads
  - 删表/删索引物理删除,不需要compact
- Oplog标记删除->Sst文件物理删除
  - 减少Oplog写放大
- 开启rowCache/减少blockCache
  - 针对kv业务提高点查询命中率
  - 牺牲range查询的性能

MongoRocks改造

- 优化后MongoRocks与WT性能对比

64GB/24core/nvme*4/raid5

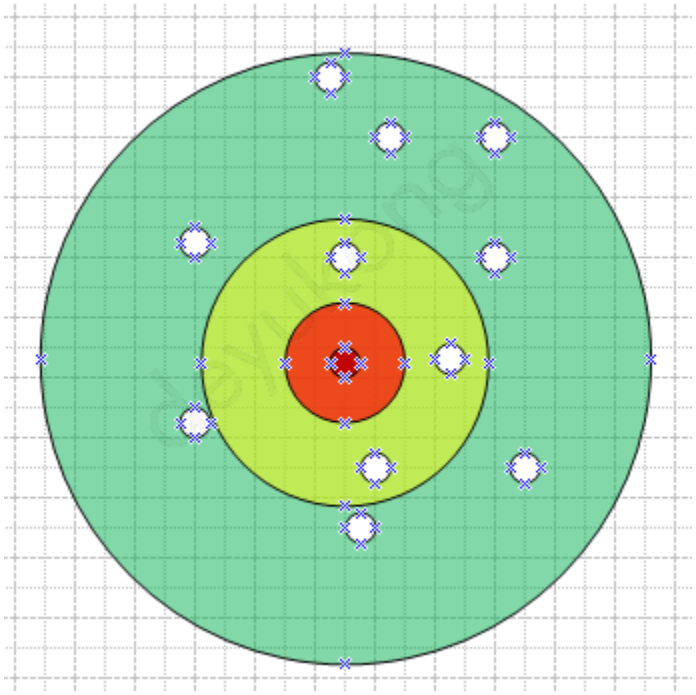


运营中的问题与优化

- GeoNear查询的效率低
- 短连接场景CPU高
- 新增节点容易追不上Oplog

GeoNear查询的效率低

- 从起始点开始的不断向外扩散的环形搜索过程
- 参数：
  - 初始步长/圆环增量半径



GeoNear查询的效率低

- 初始步长设置
  - 与中心点距离最近的点的距离的三倍
- 迭代步长增量设置
  - 内环搜索结果小于600个，则步长*2

GeoNear查询的效率低

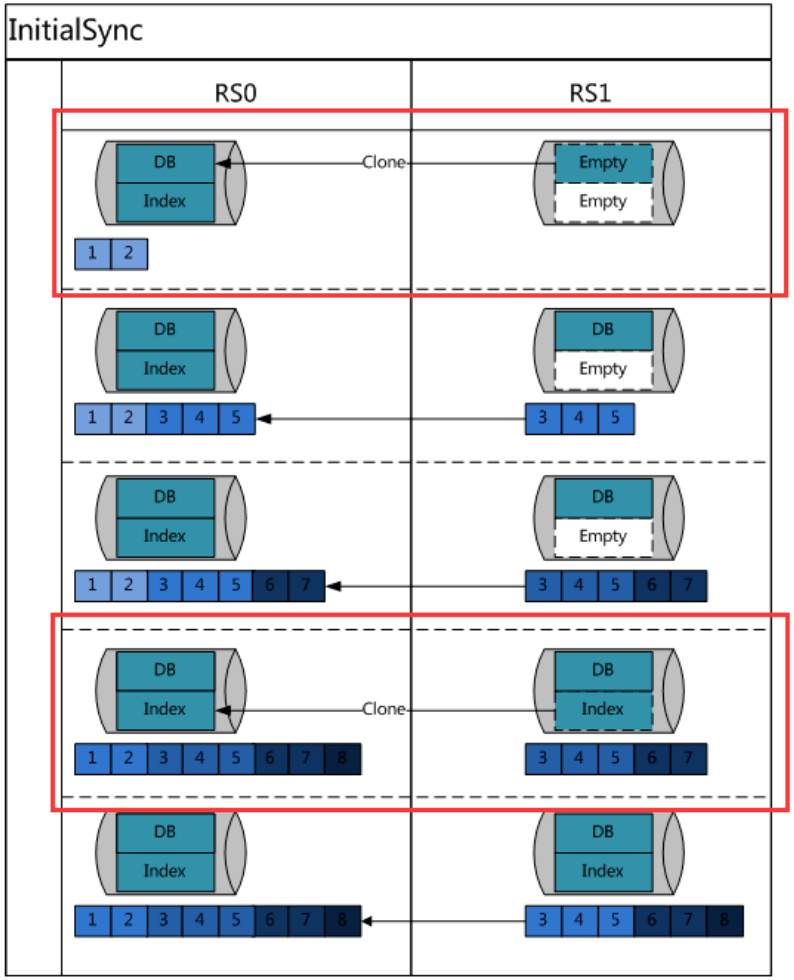
- 搜索结果小于600，步长*2
  - 太奔放
  - LBS应用一般只需要几十个结果
  - 迭代步长可配置，默认最小步长迭代
- 稠密点集下，一个圆环需要扫描的数据多
  - 需要保证严格的"最近"，圆环内必须全部扫描
  - 设置一个圆环的搜索结果个数阀值
  - 超过阀值则牺牲正确性，保证效率
- 修改内核，增加superGeoNear接口
  - db.runCommand({"superGeoNear": "coll", "near":[x,y],...})

短连接场景CPU高

- MongoDB连接认证使用/dev/urandom作为随机源
  - 内核spinLock
  - 多线程争抢严重
- 应对办法
  - 修改MongoDB内核，替换随机算法

新增节点追不上Oplog

- MongoDB新增节点通过Dump源数据/追Oplog方式

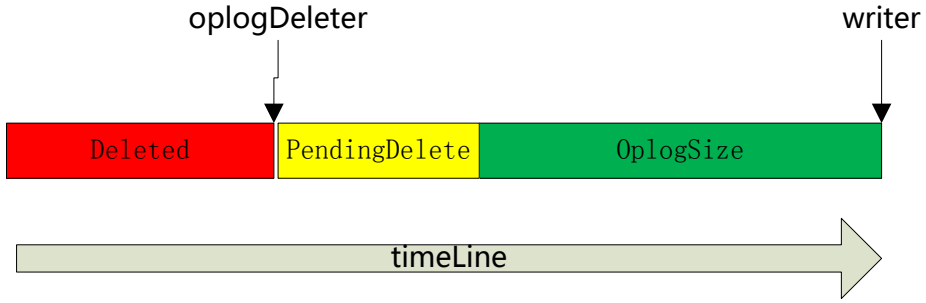


1.拷贝数据
2.追Oplog
3.追Oplog
4.构建索引
5.追Oplog

拷贝数据/构建索引太慢，导致Oplog被冲刷掉

新增节点追不上Oplog

- wiredTiger/mongoRocks
  - 后台线程淘汰最旧的Oplog
  - 保持Oplog数量稳定
- 解决方法：
  - resizeOplog(特色功能)
  - 动态在线变更oplog大小
    (https://github.com/wolfkdy/mongo/commit/eacf2ffd5206435b1b2b5e0abd3d86275e345973)



未来展望

- 更贴心的使用体验
  - 慢日志查询与聚合
  - 备份数据可提供下载
- 内核级的性能优化
  - 从库读的稳定性优化(SERVER-20328)


DBAplus
www.dbaplus.cn
THANK YOU