DTCC

第十一届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2020

架构革新 高效可控

IT168.com ChinaUnix.net ITPUB

北京国际会议中心 | 2020/12/21-12/23

腾讯CMongo架构与优化实践

杨林
腾讯-云架构平台部

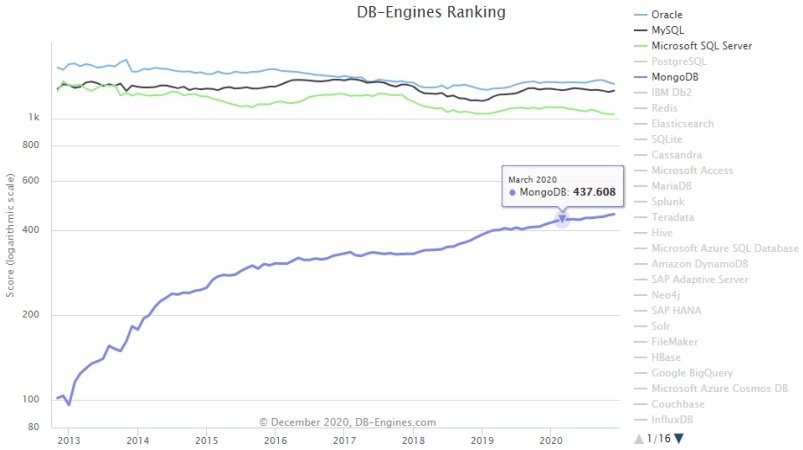
01 CMongo简介

02 整体架构

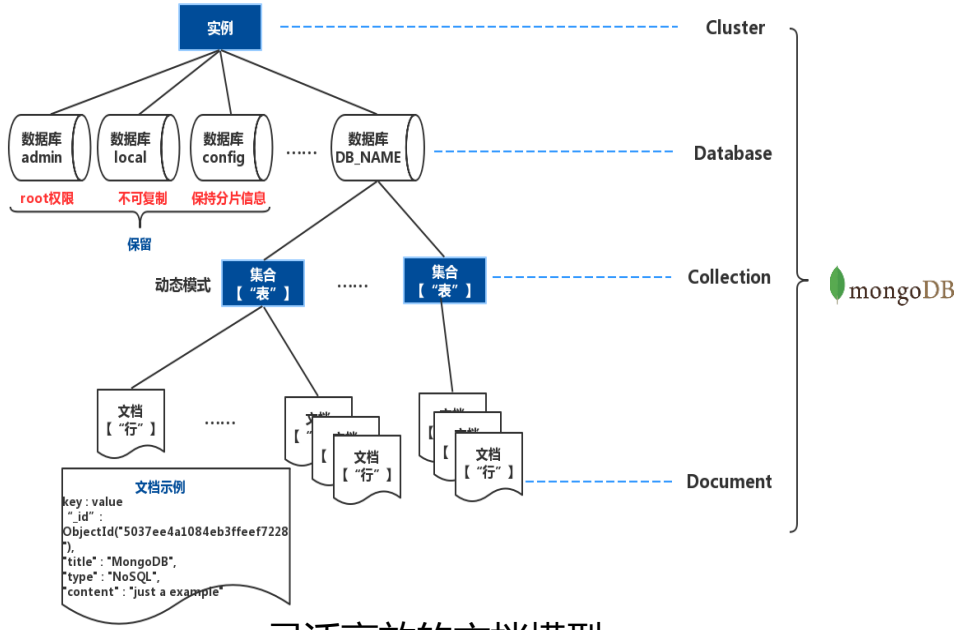
03 功能特性&最佳实践

04 未来规划

MongoDB



NoSQL领头羊，增长势头持续、强劲



灵活高效的文档模型

CMongo简介

灵活性

灵活的表结构：Schema-free
丰富的索引类型
Binary-json友好的交互方式

易扩展

优秀的 横向+纵向 扩缩容能力
分片之间自动负载均衡，对用户使用透明
灵活可配的迁移策略，将性能影响降到最低

企业级功能

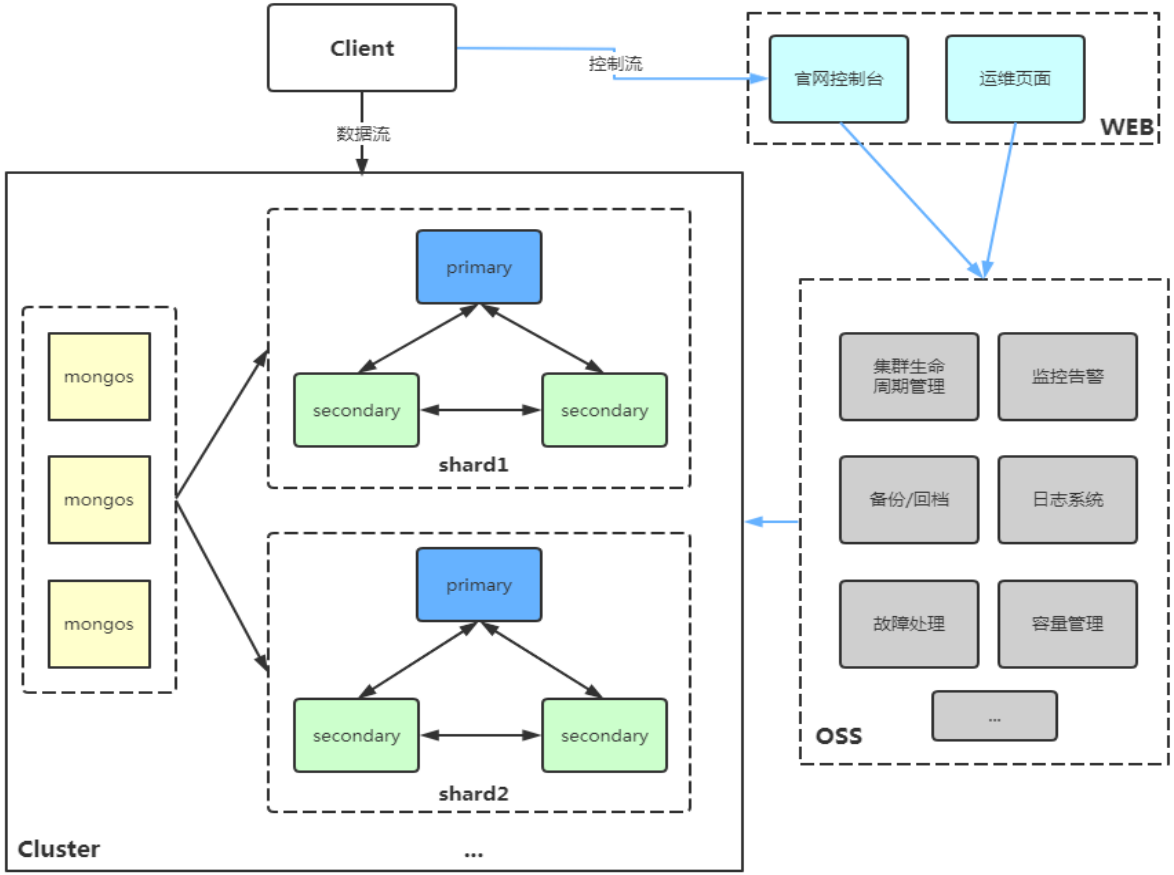
审计：优于官方企业版的审计能力
存储加密
全链路限流
中文全文索引

高性能

专业内核团队：贴近业务定制内核特性
多种插件式引擎：
WiredTiger/RocksDB/InMemory/Mmap
优秀的读写性能：
具体参考腾讯云 官方测试报告

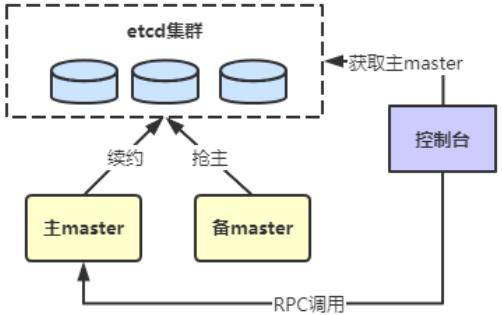
整体架构

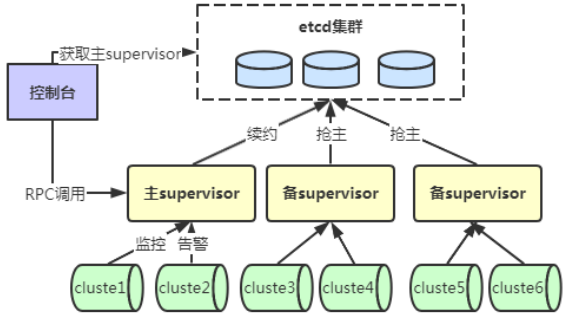
OSS
+
内核
+
运营系统

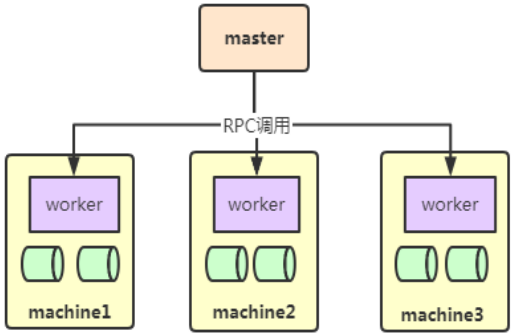


管控架构

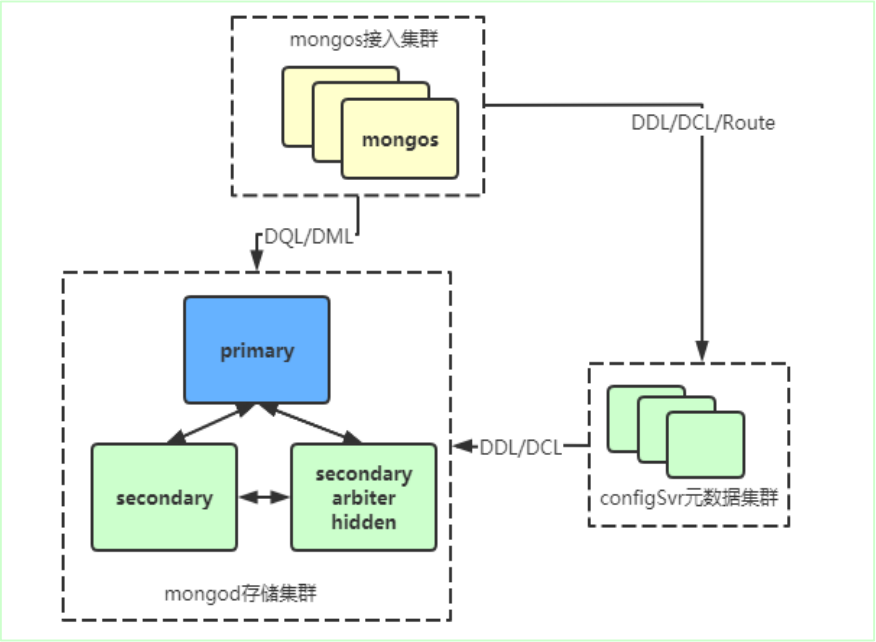
Master	Supervisor	Worker
管控核心模块 集群生命周期管理 对外RPC接口 资源管理	集群拨测、告警 巡检、健康状态报告 自动化运维	宿主机agent 进程管理 机器资源管理



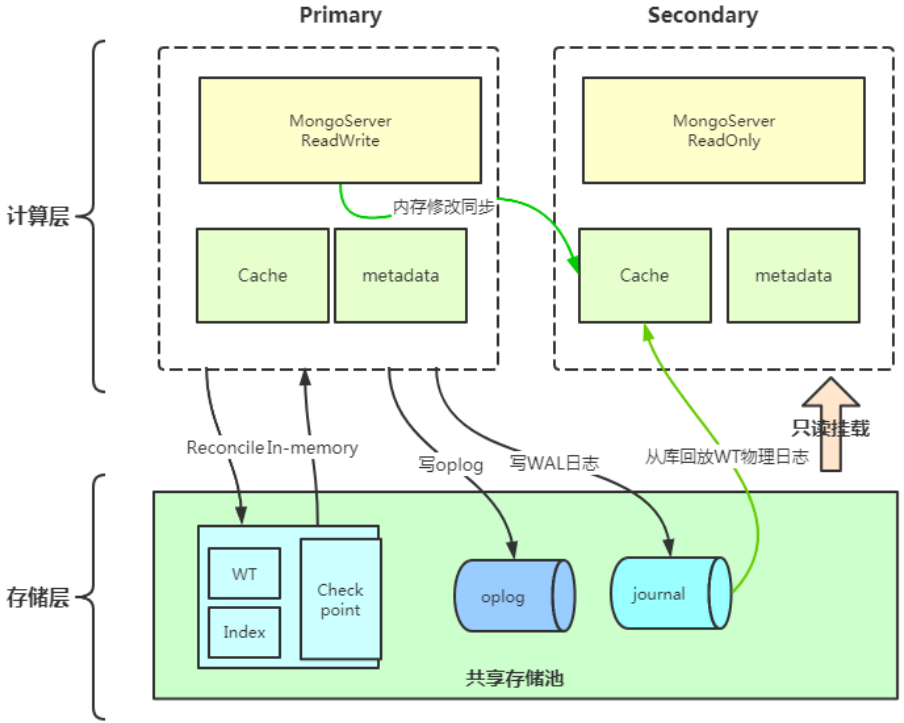




内核架构



原生架构



计算存储分离架构（coming）

功能特性

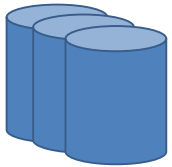
管控

- ✓ 集群创建、删除、扩缩容
- ✓ 备份、回档
- ✓ 故障处理
- ✓ 监控、告警
- ✓ 智能诊断
- ✓ 资源隔离
- ✓ 容量、连接数控制
- ✓ 跨地域容灾
- ✓ 资源管理
- ✓ ...

内核

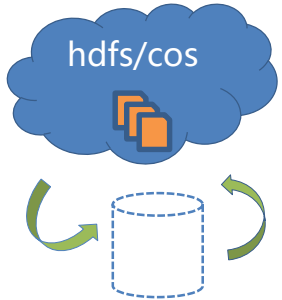
- ✓ 3.2开始支持从库snapshot读（官方4.0版本才支持该特性）
- ✓ geoNear优化（相比原生性能提升10倍）
- ✓ MongoRocks优化
- ✓ 基于checkpoint的不停服物理备份
- ✓ 大量短连接下随机数生成算法优化
- ✓ 白名单免密
- ✓ 动态resize oplog（代码已被官方接受）
- ✓ TTL索引优化
- ✓ 审计、加密
- ✓ 内核全链路过载保护（业界独家）
- ✓ skip + limit优化
- ✓ ...

管控-企业级数据安全



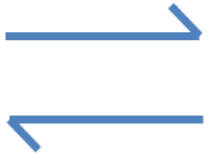
多副本

数据多副本，高效HA



备份/回档

高效备份回档，有“备”无患，兜底数据安全



只读/灾备

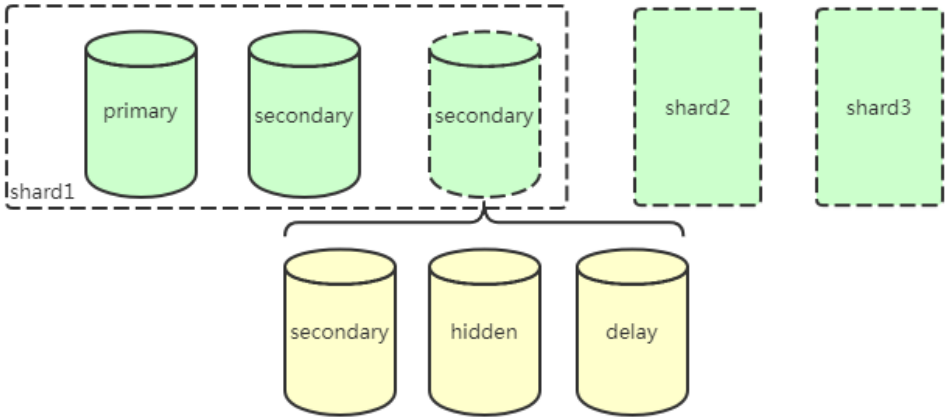
实时同步，镜像集群

审计、加密

请求可追踪，整链路数据加密

多副本

- **多种类型**
  secondary、hidden、delay、arbiter等
- **支持50个从节点**
- **自动HA容灾**
- **节点状态定期巡检**



挑战：

- ✓ 资源利用最大化？
- ✓ 多副本高效容灾
- ✓ 节点角色持久化

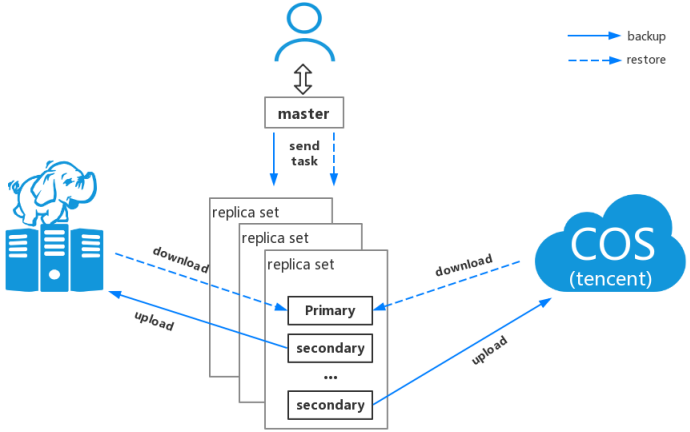
解决方案：

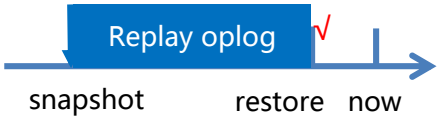
- ✓ 机房、机架、机器多维度容灾
- ✓ 降序最佳适应算法(BFD)，避免碎片，提升资源利用率
- ✓ 节点迁移、变更带状态

备份回档

- 逻辑备份 VS. 物理备份
- 全量备份 VS. 增量备份（oplog）
- 动态调整备份间隔，保证备份连续覆盖7天时间

- 整实例回档 VS. 库表回档
- 可回档到7天任意时刻



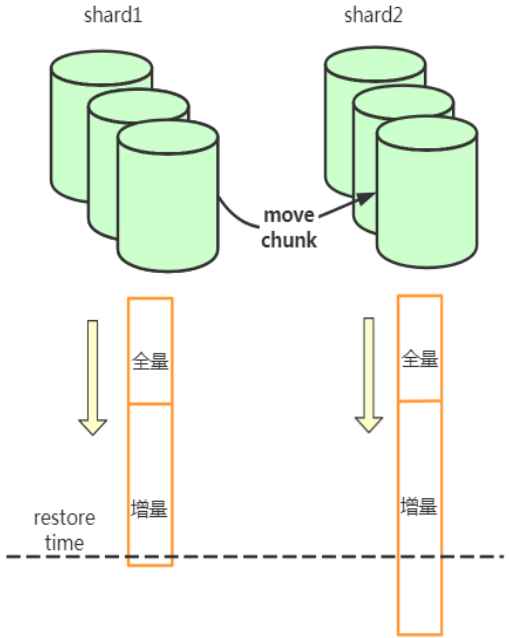


分片回档

分片集群回档数据+路由

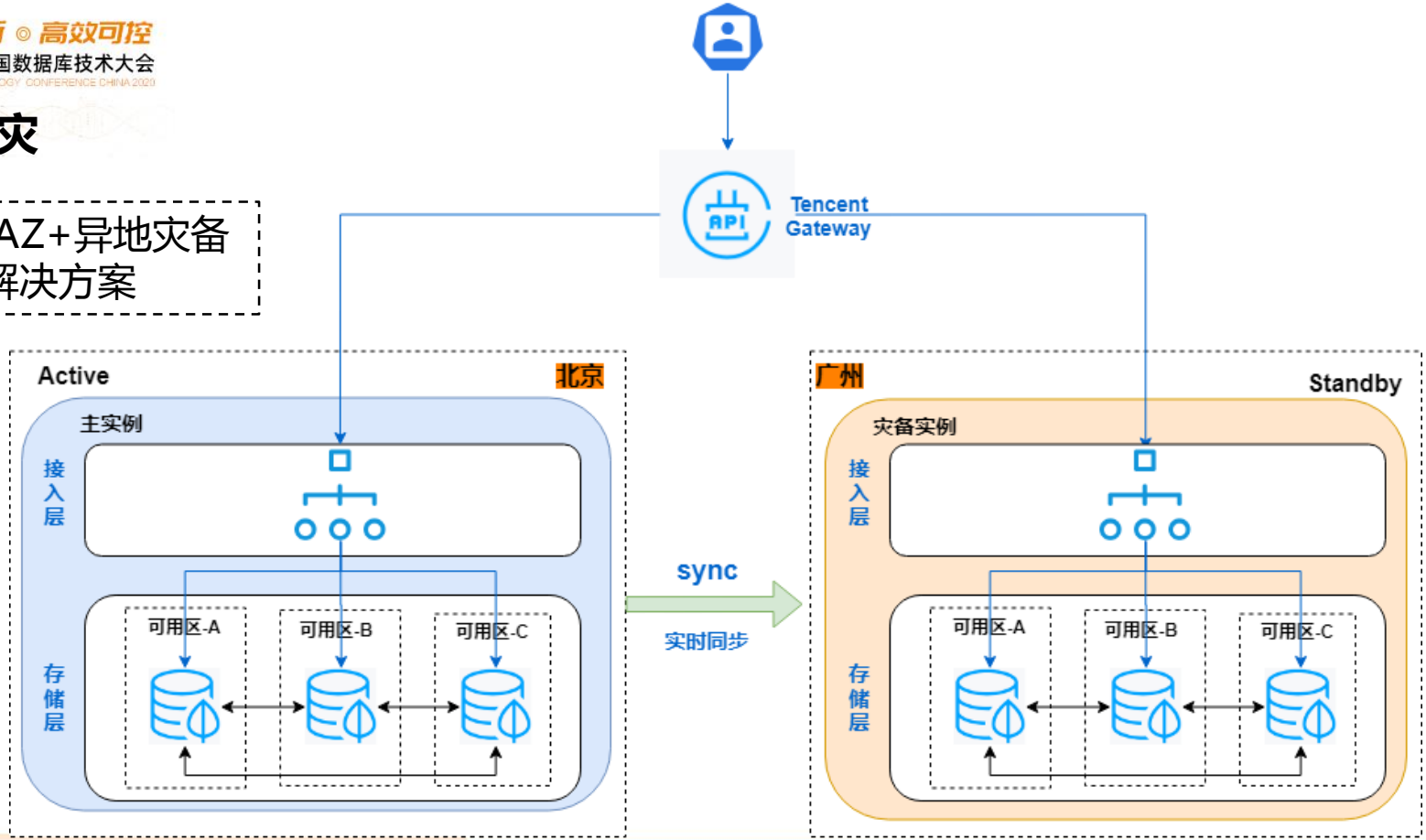
分片集群回档时间全局有效性

回档“脏数据”过滤



跨地域容灾

同城多AZ+异地灾备
解决方案

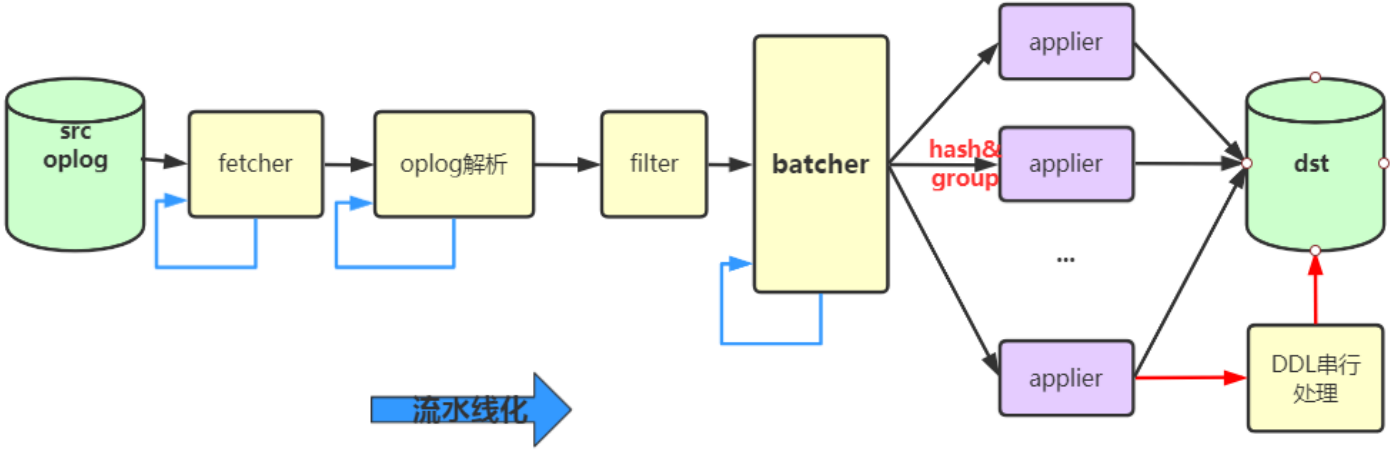


跨地域容灾

挑战：

- 同步速度
- 数据一致性如何保证?
- 异常处理

  断点续传
  状态监控&故障恢复



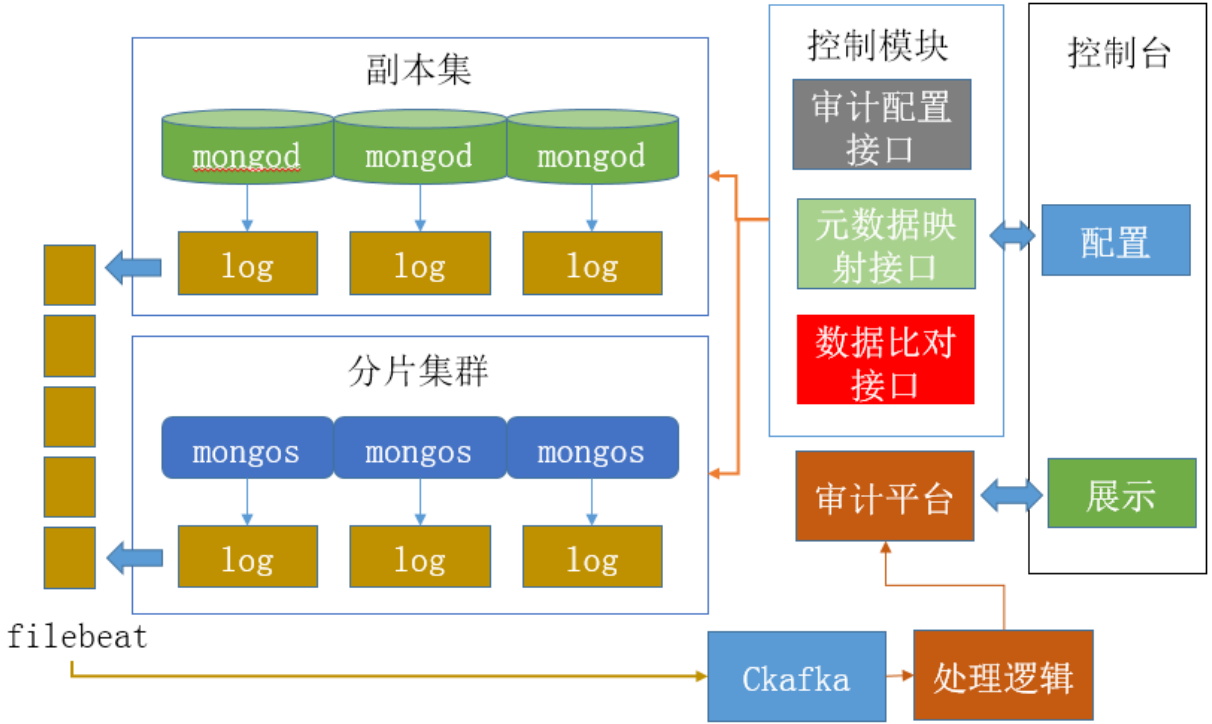
- ✓ oplog由串行改并行回放，速度提升N（并发数）倍
- ✓ DDL串行处理

- ✓ 源和目标定期比对、校验
- ✓ rollback的优雅处理
- ✓ 灾备集群只读权限控制

- ✓ 增量阶段oplog ts持久化
- ✓ 同步状态实时上报

审计

- **审计范围**
  - DDL
  - CRUD可选开启
- **审计规则**
  - 支持动态增删改查
  - 灵活性高(细粒度)
- **亮点**
  - 审计实时监控
  - 支持同步/异步两种模式
  - 性能损耗5%以内



管控-核心PaaS服务能力

秒级监控

异常告警

定期巡检

过载保护

弹性伸缩

自动化运维

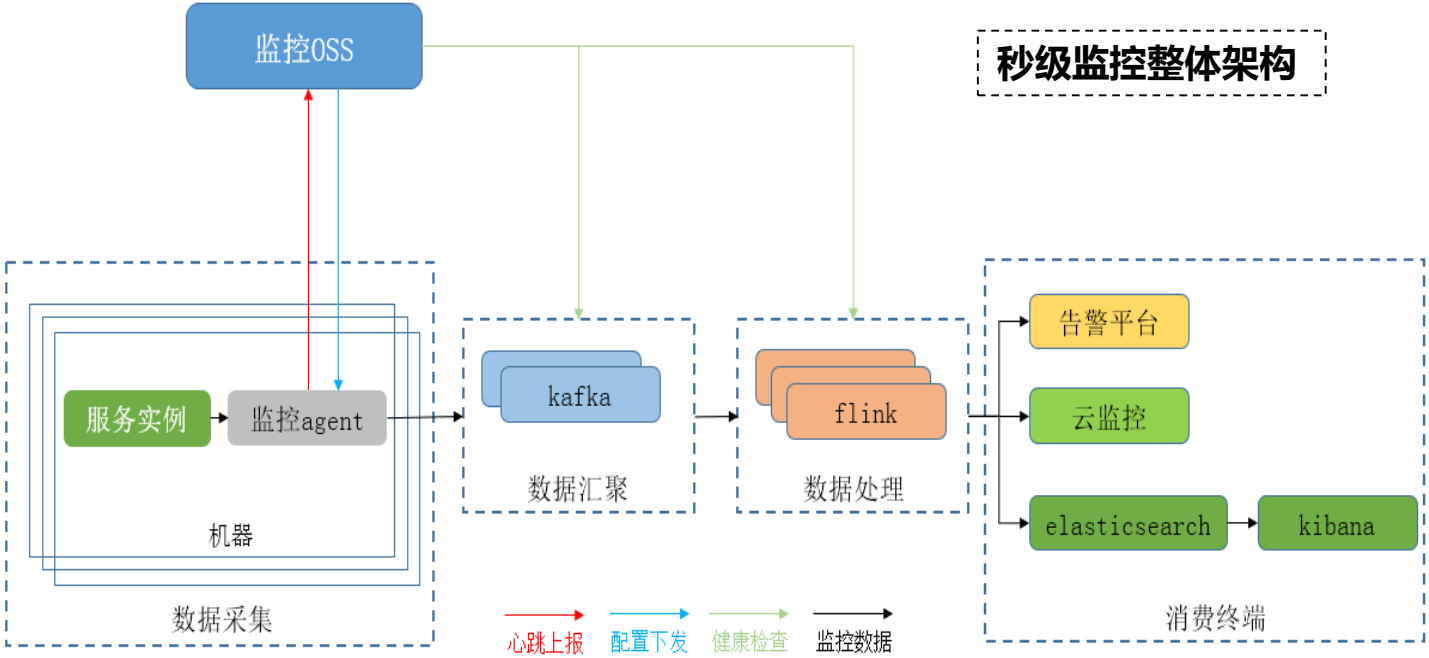
秒级监控

旧版监控：

- 监控粒度粗
- 缺乏关键指标
- 集中式上报，可用性低
- 扩展性差

新版监控：

- 秒级监控
- 系统关键指标上报
- 集中式改为分布式架构



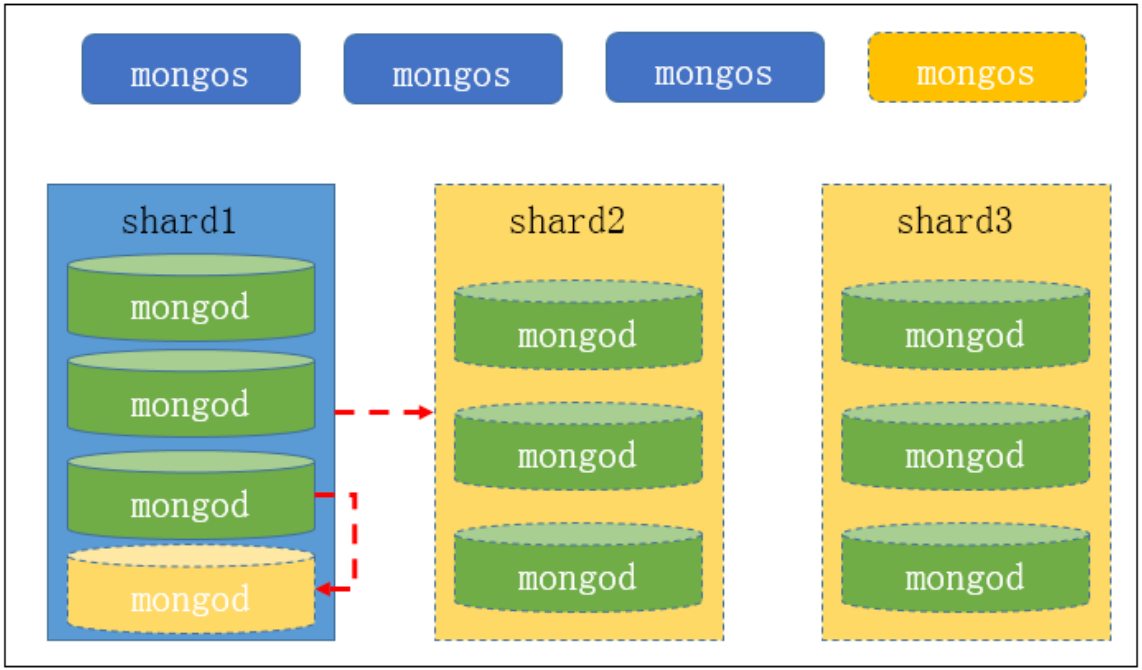
弹性伸缩

Scale up：

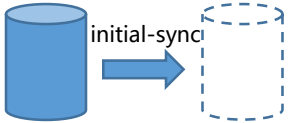
- 扩节点资源
- 扩从节点，上限50

Scale out：

- 扩分片数，上限128

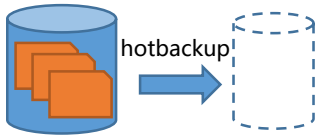


弹性伸缩-加节点



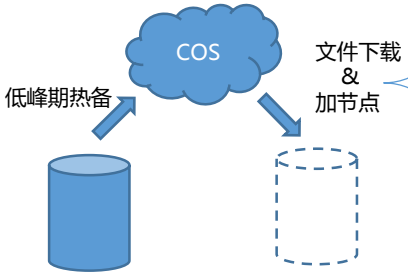
方案一：
基于原生加节点逻辑

- 逻辑同步
- 建索引耗时太久
- 影响源节点



方案二：
hotbackup直接拷贝文件新建节点

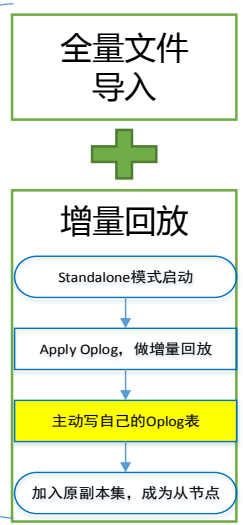
- 影响源节点，尤其高峰期可能导致业务雪崩



最终方案

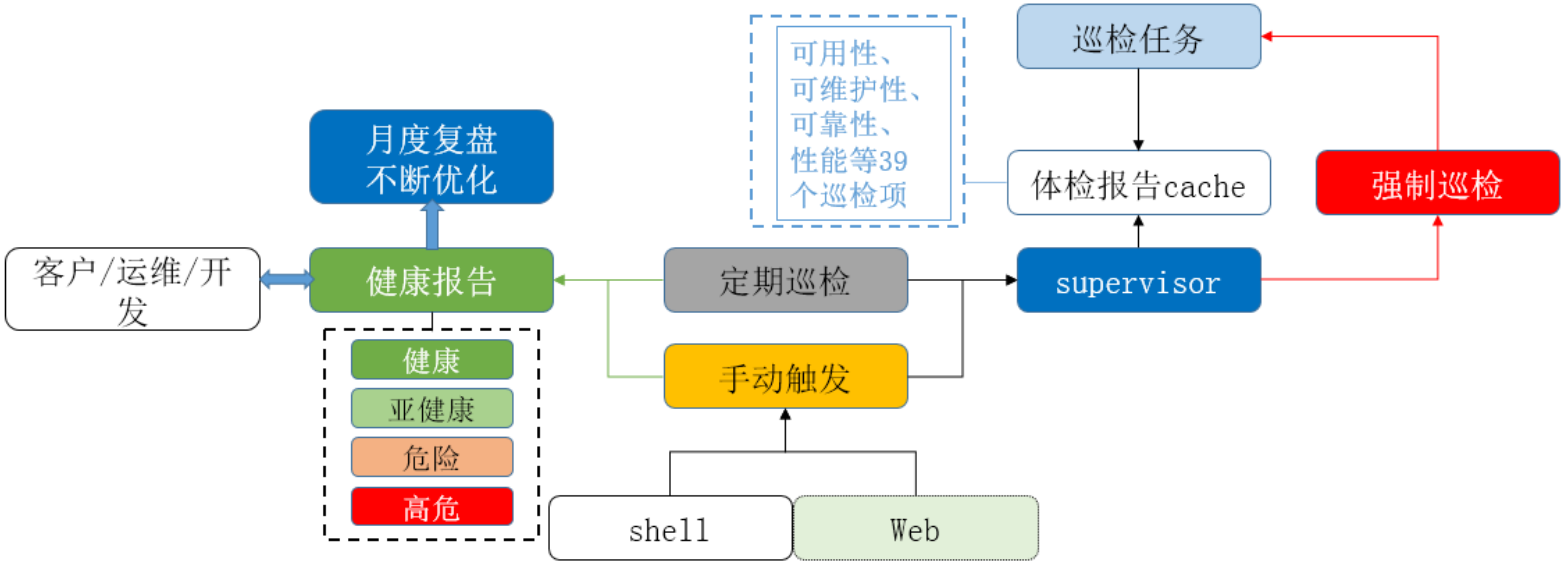
方案三：
基于远端热备文件加节点

- 对源节点无影响
- 充分利用冷备文件
- 尽可能加快恢复速度



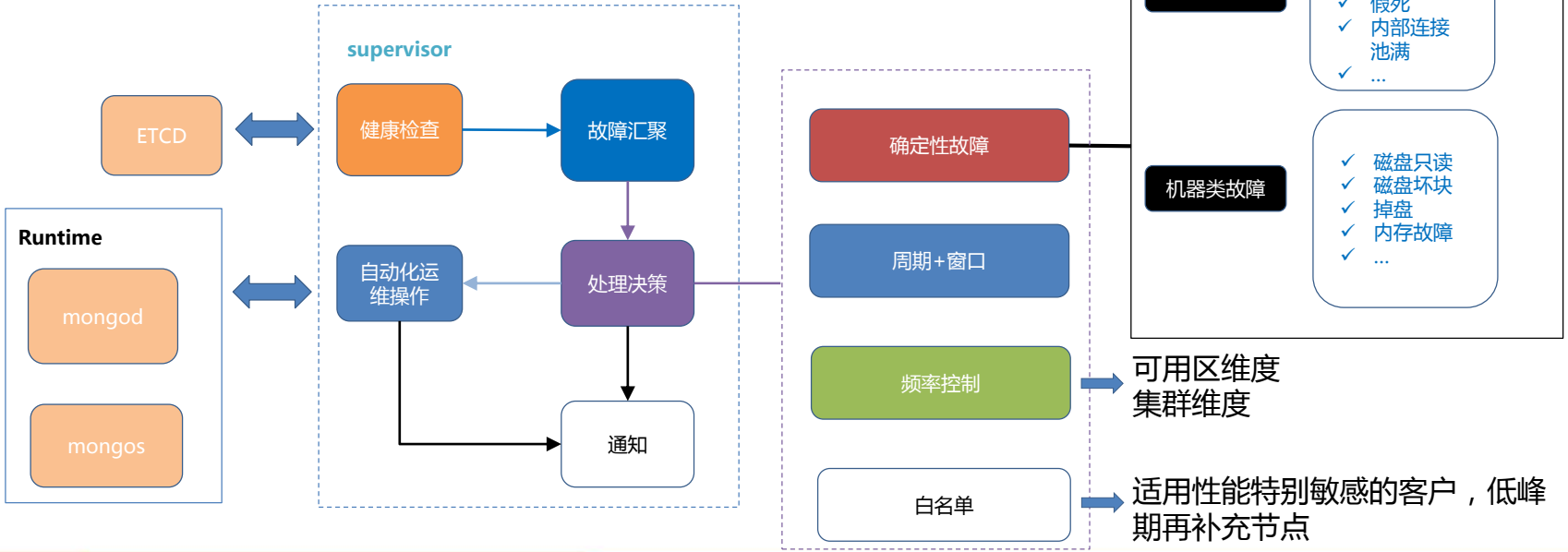
巡检

目标：集群健康体检，提前发现集群潜在风险



自动化运维

自动处理**确定性**故障，快速修复问题，保证集群可用性



内核-深度优化定制

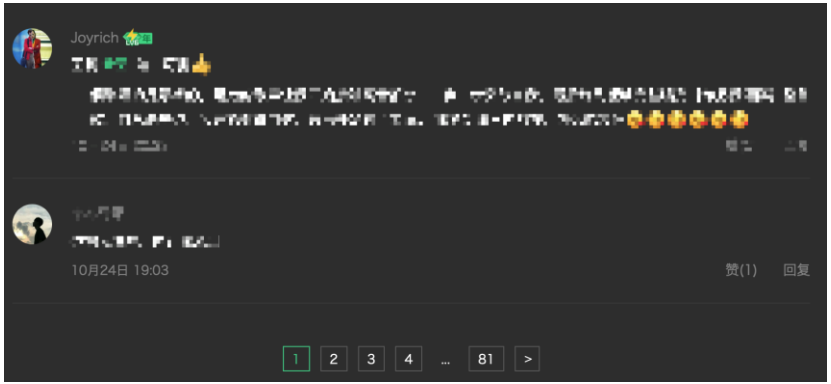
Skip+limit优化

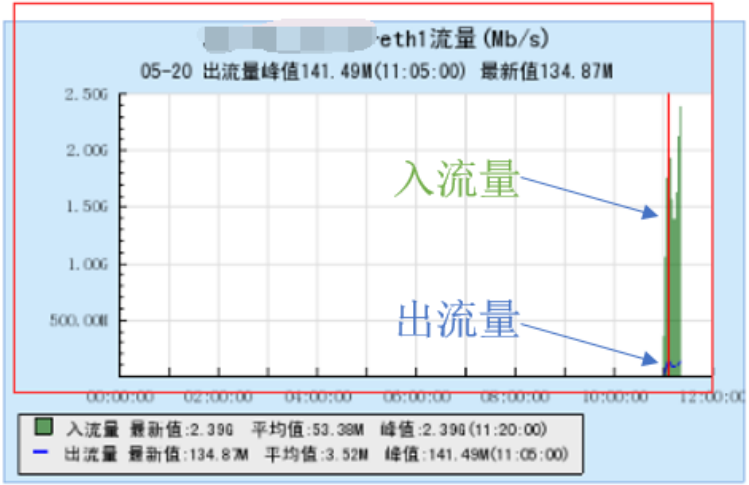
全链路内核流控

skip+limit优化

K歌和Q音的评论、作品“大翻页“：

- 几十甚至上百页的翻页跳转
- cursor只能顺序读，需使用skip+limit



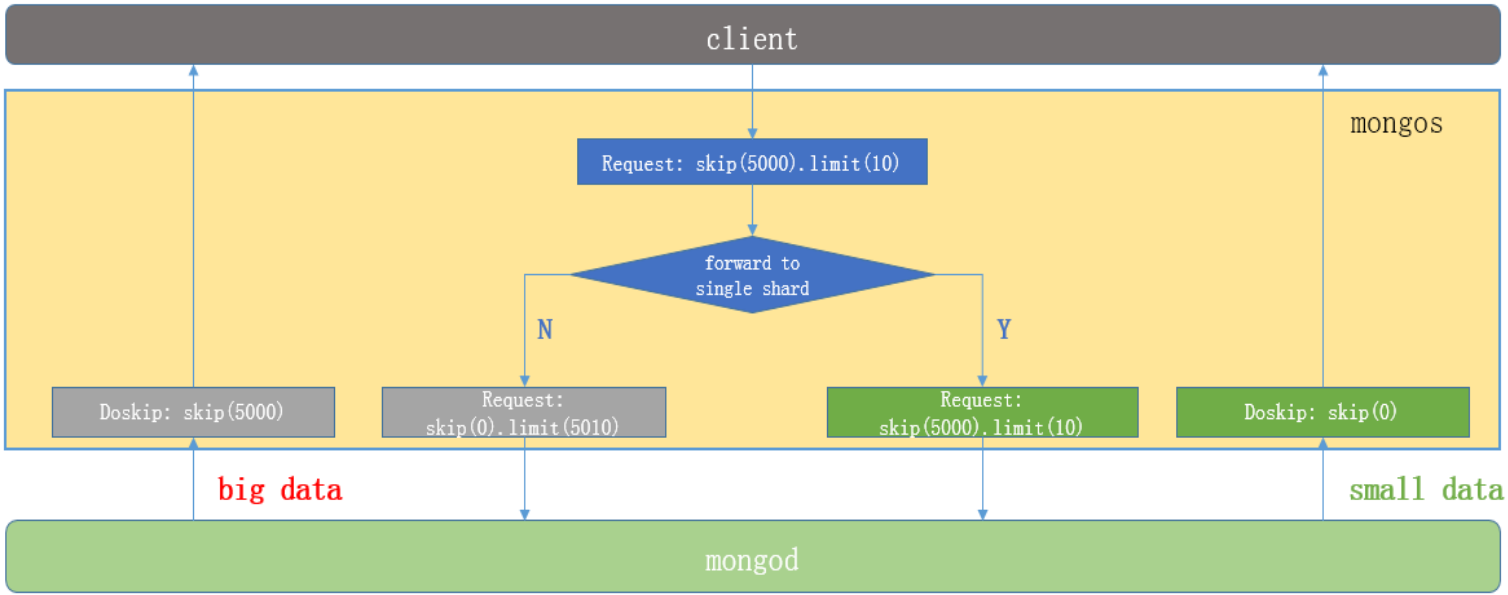


问题：

- 性能差，QPS上不去
- mongos入流量是出流量上百倍
- mongos cpu接近打满

skip+limit优化

- 优化思路：命中单个shard，skip下推到mongod



skip+limit优化

➢ 性能对比

版本对比	请求总数	并发数	耗时	网卡流量	mongos-CPU	mongod-CPU
原生版本	200	5	6.3s	120MB/s	30%	13%
优化版本	200	5	**0.6s**	**<1MB/s**	**1.7%**	14%

（备注：测试场景为查询数据落在单分片）

- ✓ 网卡流量下降 2 个数量级
- ✓ CPU利用率（peak）降低 1 个数量级
- ✓ 性能提升 1 个数量级

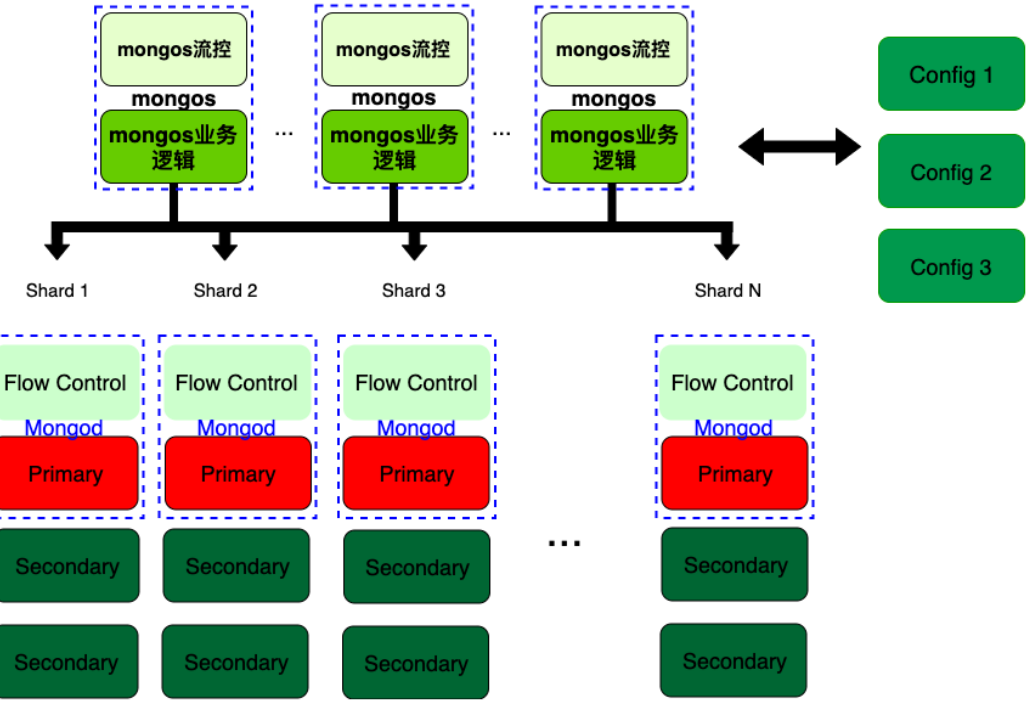
全链路流控

线上问题：

- 原生集群缺少流控
- 短时大量请求打垮集群

流控方案：

- 嵌入式设计，无需额外模块
- 自适应平滑限流
- 全内核逻辑



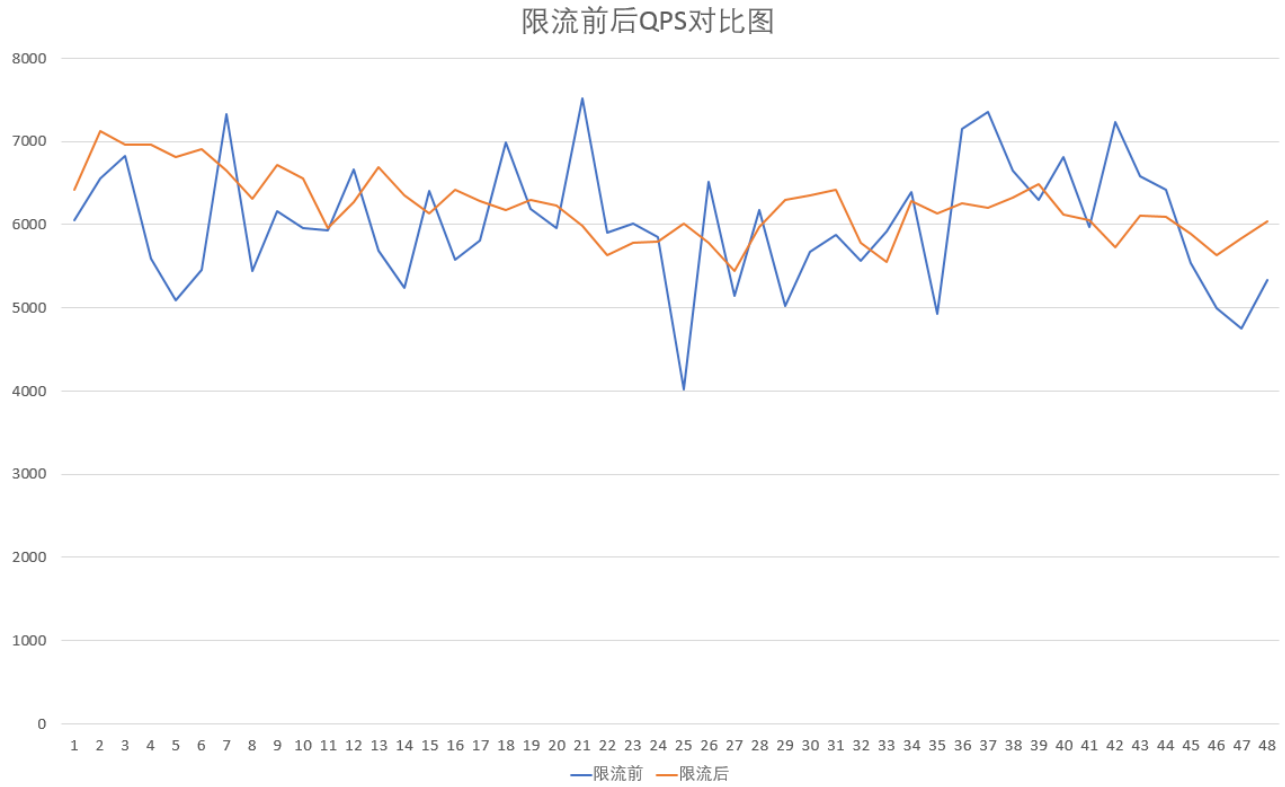
全链路流控

限流因子：

- 资源 – cpu/mem/io/net
- 锁 – ar/aw、qr/qw等
- QPS – tcp Vegas

效果：

- 更加平滑稳定的QPS
- 业务请求突发，系统不雪崩
- 限流因子阈值动态可配
- 动态开关



未来规划

技术及产品持续建设

业务场景

MongoDB托管服务
- 公有云
- 内部云
- 多模态支持

弹性文档服务
- ServerLess
- TCB（云开发）

生态融合

内核

性能
- 鉴权优化
- evict优化
- 定制内核

新特性
- 计算存储分离
- MongoRocks4.2
- 审计、加密

可用性
- 过载保护
- 连接控制
- 库表粒度Qos

成本
- 冷热分离
- 内存优化

管控平台

管控操作
- 资源管理
- 实例管理
- 作业调度

运维系统
- 运维web
- 工具系统
- 健康巡检

作业调度
- 定时任务
- 备份回档
- 生命周期管理

监控告警
- 读写拨测
- 故障检测
- 采集上报

THANKS