mongoing
中文社区

基于MongoDB的弹性文档服务

杨林
TEG云架构平台部

Tencent

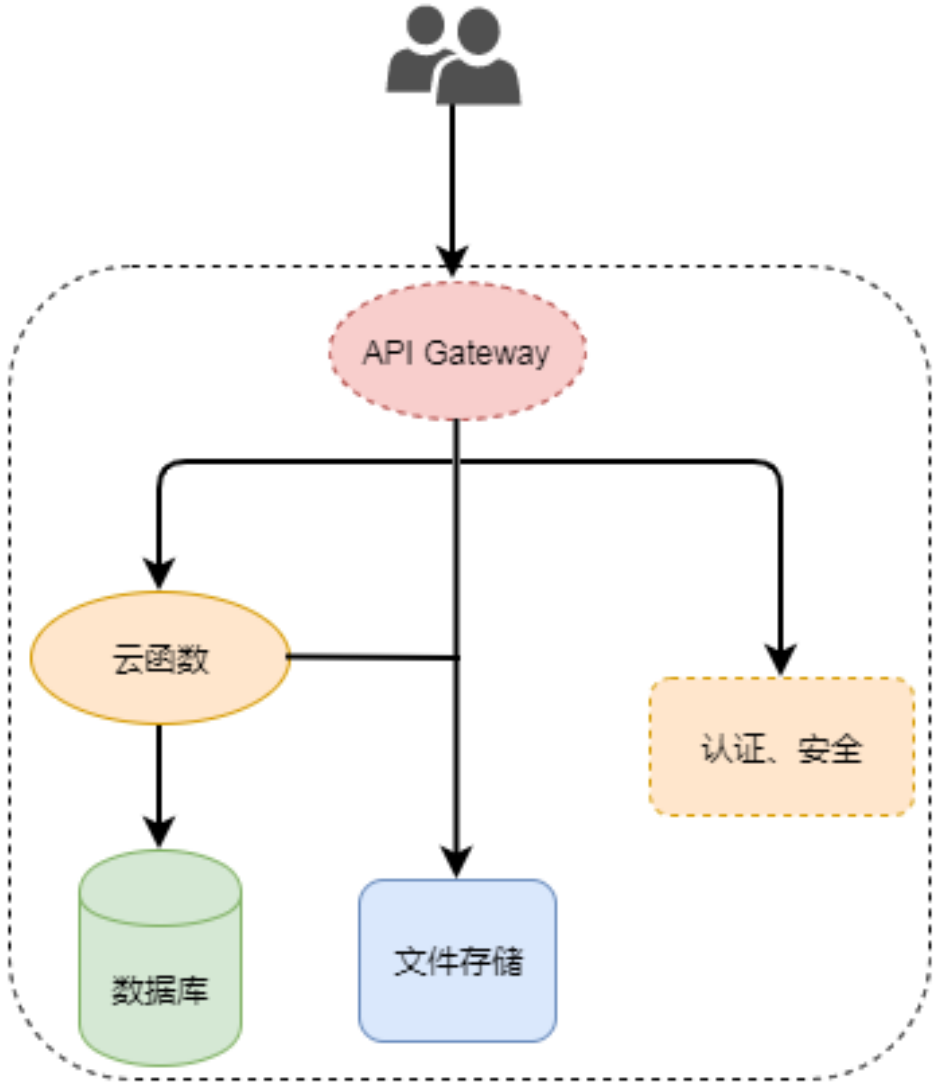
CONTENTS

一、背景

什么是云开发?

优势:

- 一站式后端服务
- 免运维
- 专注业务逻辑，快速迭代



三大基础能力:

- **云函数**：计算单元，编写并部署在云端的函数，随时调用
- **存储管理**：文件上传/下载及管理功能
- **数据库**：用户核心数据存储，前端或云函数中对数据库进行读写

一、背景

Tencent

对数据库的要求?

- 灵活、易扩展
- 数据高可靠
- 成本低，最好支持按量付费
- API简单且相对丰富
- Restful API
- 低延时，高性能
- NoSQL

- …

Tencent

CONTENTS

二、整体架构

Tencent

方案选型

行业调研

产品	特性
Dynamodb	完全托管的NoSQL 良好的扩展性 REST API支持
Firestore	REST、RPC等多种API 支持文档模型 离线读写 弹性扩展
MongoDB Stitch	支持mongo协议 MongoDB生态

数据库选型

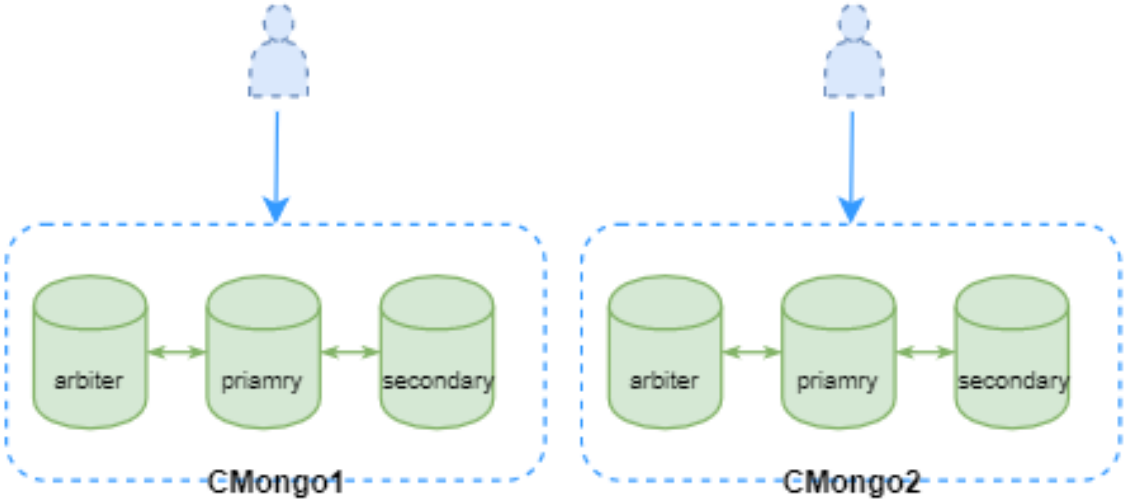
数据库形态	优势	劣势
SQL	SQL语法完善 成熟的事务支持	扩展性差
NoSQL	扩展性好 天然分布式 灵活的kv模型，对齐业界竞品	事务能力较弱

最终方案：基于CMongo自研

二、整体架构

FlexDB的两种架构

独占型



优点：

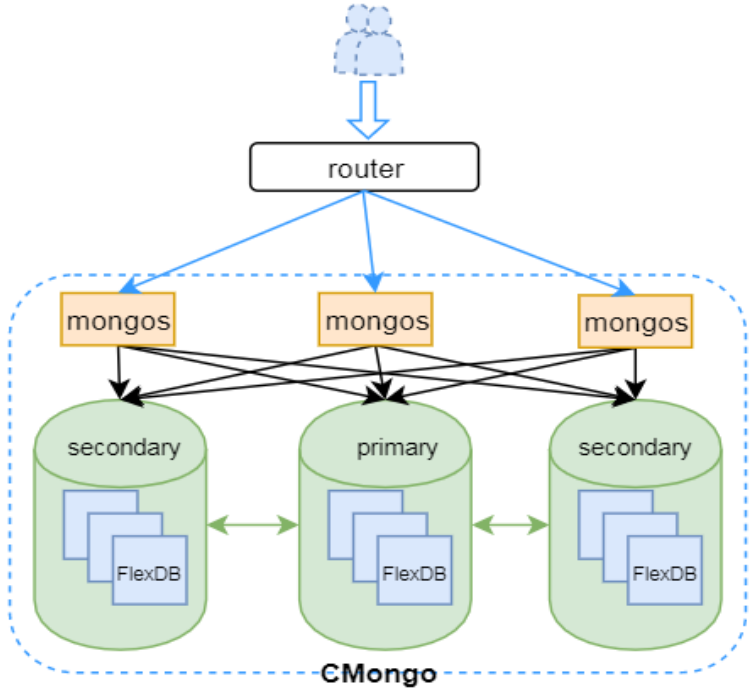
- 一对一模型，隔离性好
- 改动小，监控、扩容等功能都可以复用
- 大用户友好

缺点：

- 资源浪费严重，使用率低
- 成本太高，无法应对海量用户量
- 集群数太多，难以管理

VS

共享型



优点：

- 多对一的模型，节省资源
- 单用户成本低，超卖操作空间大
- 集群个数少，管理维护方便

缺点：

- 多个租户共享一个集群，资源争抢，热点问题
- 监控、告警、扩容等运维功能需要重新开发

Tencent

CONTENTS

Tencent

DB隔离

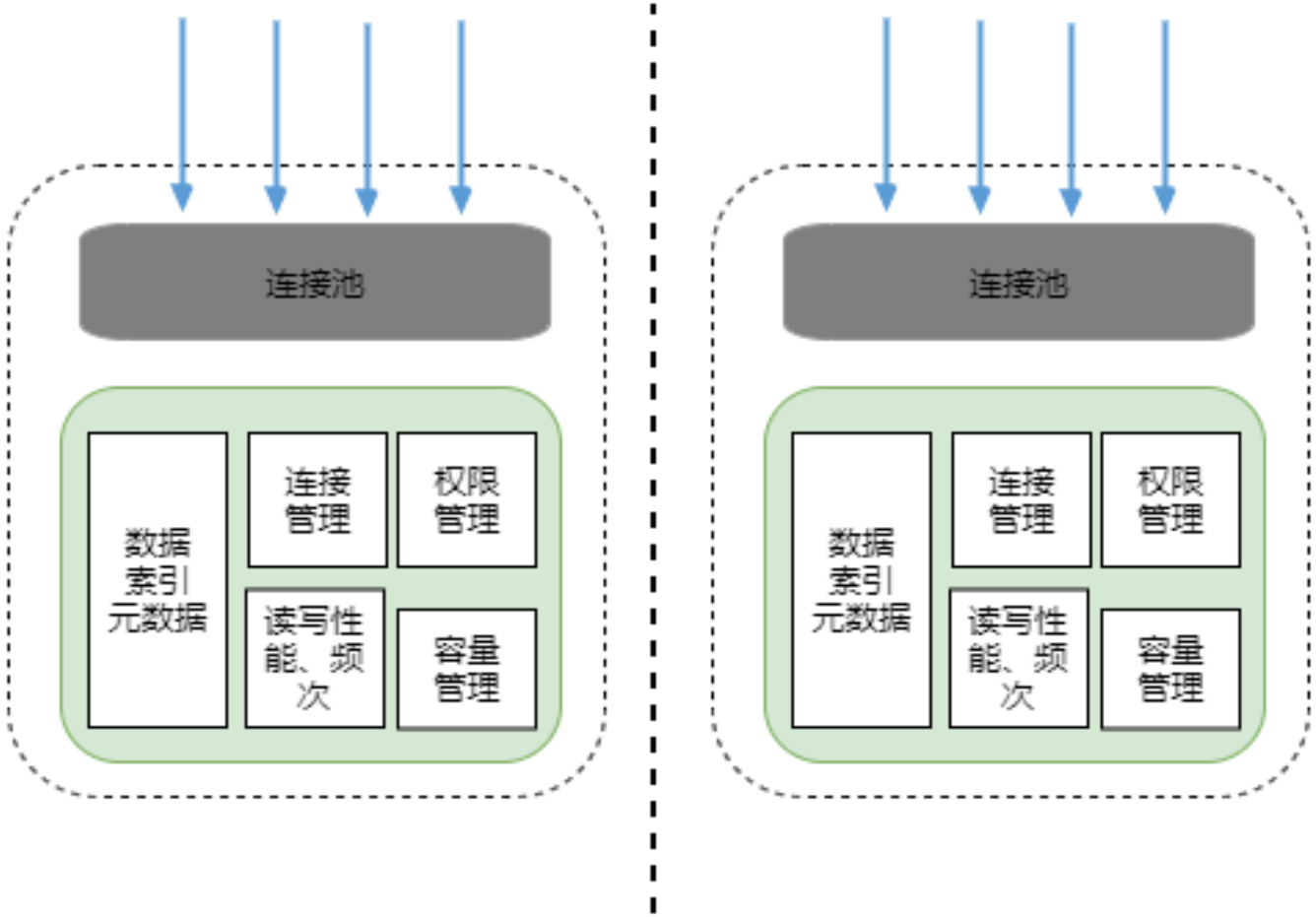
- 权限隔离 √
- QoS限制 ×
- QPS限制 ×
- 存储隔离，容量限制 ×
- 连接数限制 ×
- 按量计费 ×

➢ 原生MongoDB支持

- 同集群不同DB分配不同user，权限隔离

➢ 二次开发

- **DB维度数据统计**：mongos、mongod内核改造，支持请求数，连接数，容量统计
- **DB维度限流**：令牌桶算法
- **管控系统**：实时采集，超过阈值告警、封禁

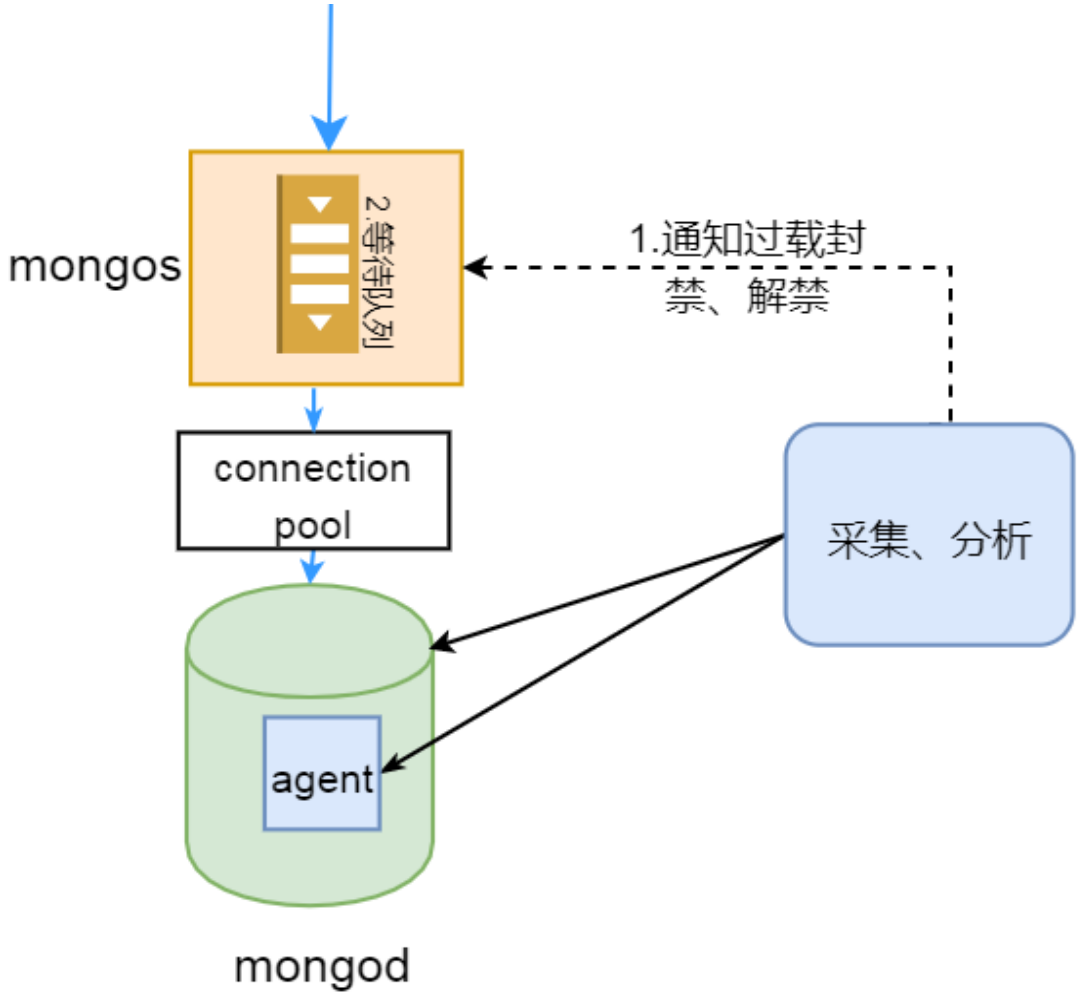


Tencent

集群过载保护

- Why?
  - 多租户，整体过载影响多个用户
  - 缺乏自我保护

- **如何实现?**
  - **mongos**：增加等待队列，判读过载
  - **mongod**：提供运行状态信息
  - **管控系统**：资源实时采集，分析、通知mongos过载保



Tencent

监控告警

监控维度

- 监控维度太少
  - mongos维度 √
  - mongod维度 （少）
  - db/table维度 X
- 分钟粒度监控

指标类型

- 慢日志分类 X
- 常用命令单独统计 X
- 平均延迟指标 X

Tencent

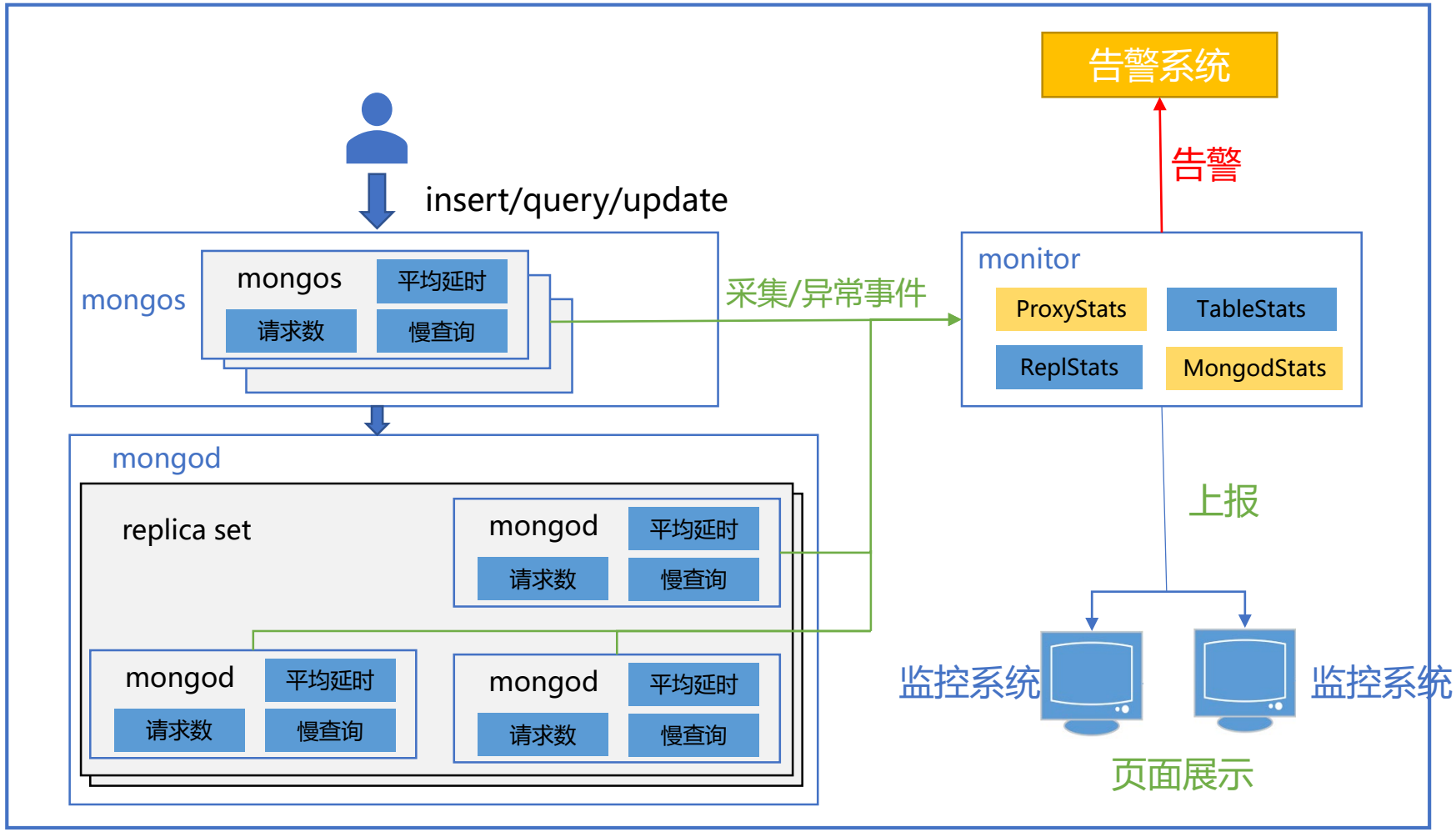
监控告警

监控事项：

- 基本CMongo集群指标
- **DB维度**基本指标
- 集群内DB**冷热**情况

告警策略：

- 分时段、级别、类别
- 收敛规则

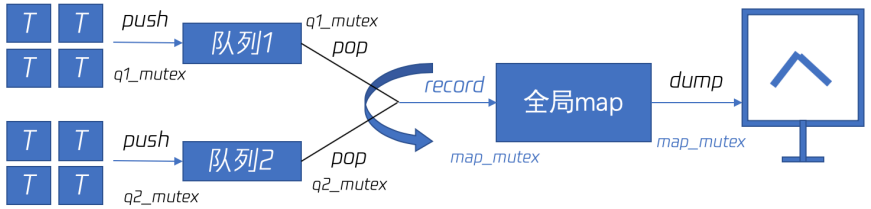


Tencent

难点与亮点

- 35+项新指标
- 集群冷热监控
- 库表维度统计
- 全自研

- 用户请求路径上添加钩子
- 采集数据耗时长
- unordered_map降低时延
- “无锁”队列降低锁争抢



- 新增命令统计结果

- 更细粒度
- 更频繁统计操作

- 数据量更大
- 计算分析要求更高
- 延时更敏感

Tencent

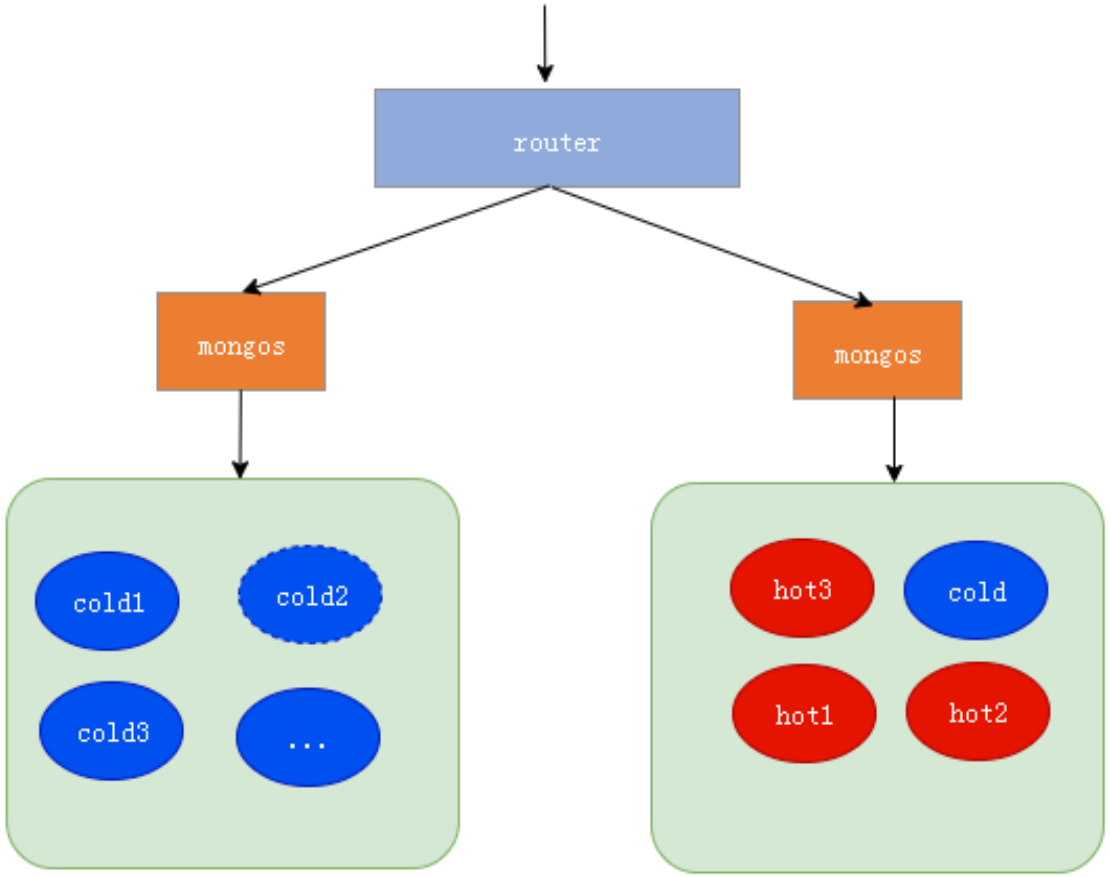
热迁移

现状

- 资源分配算法单一、固定
- 预计非活跃用户占比超80%
- 用户（即DB）存在明显**冷热分布**
- 不支持大用户独占

痛点

- 多租户共享一个CMongo集群，存在资源争抢
- 无法感知用户冷热变化
- **不支持用户迁移**



Tencent

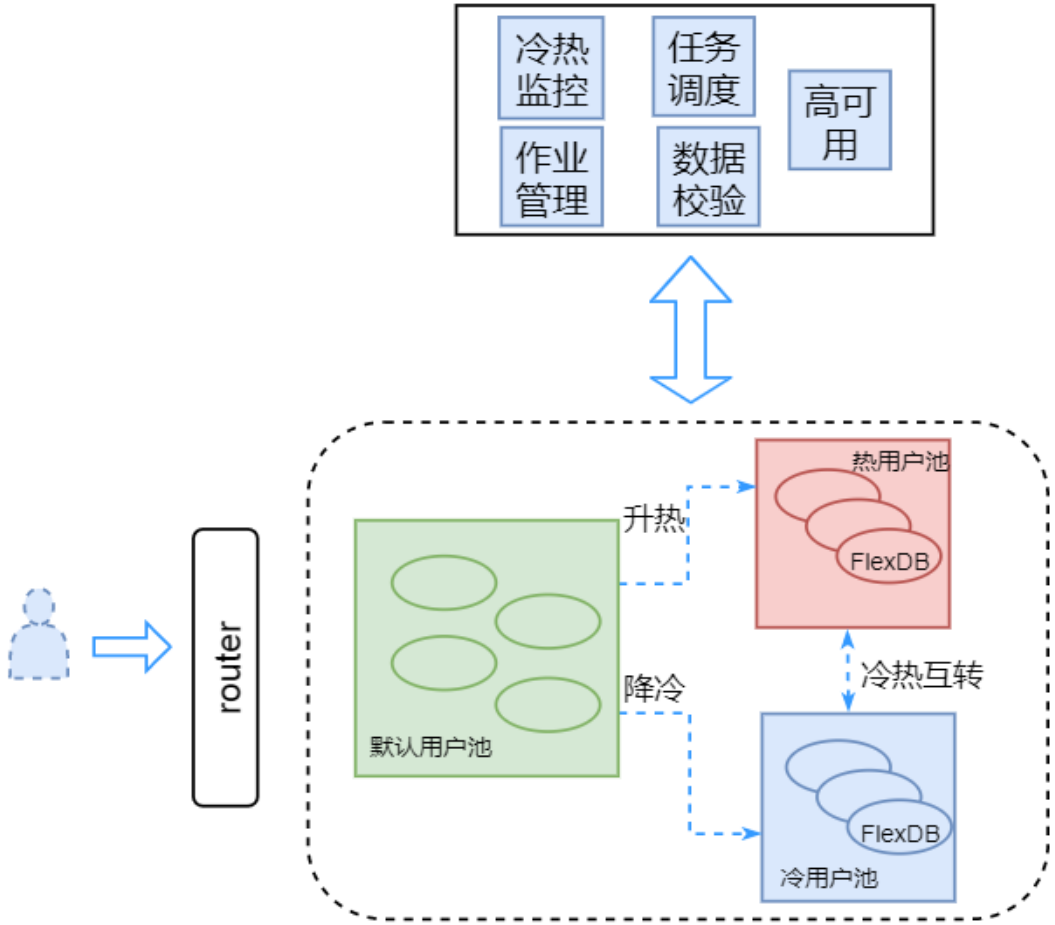
冷热分离

分配算法优化

- 冷热实例规则定义
- 首次分配，默认所有用户都分配在温用户池
- 不同用户池的CMongo集群规格不同

热迁移

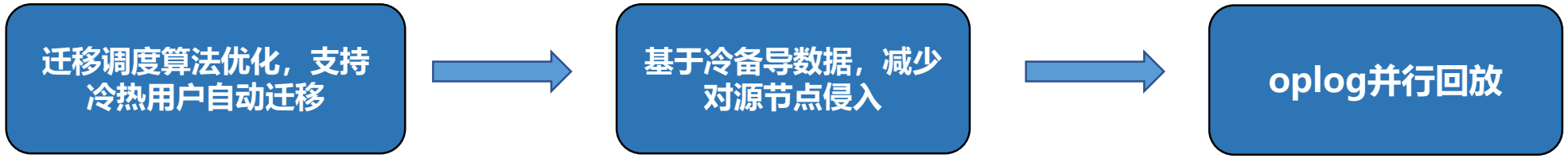
- 温降冷、温升热、冷热互转调度
- 数据正确性校验
- 用户无感知



DB全量同步 → 全量产生的oplog同步 → oplog逐次逼近 → “停写”+最后一次同步 → 修改路由 → 访问目标集群

三、挑战与实现

热迁移优化



手动迁移 --->自动迁移

充分利用冷备资源，做到对源节点几乎无影响

解析oplog，根据_id并行回放，加速流水同步

测试

- 单集群**200**个租户，每个数据大小**5GB**
- cpu负载**100%**

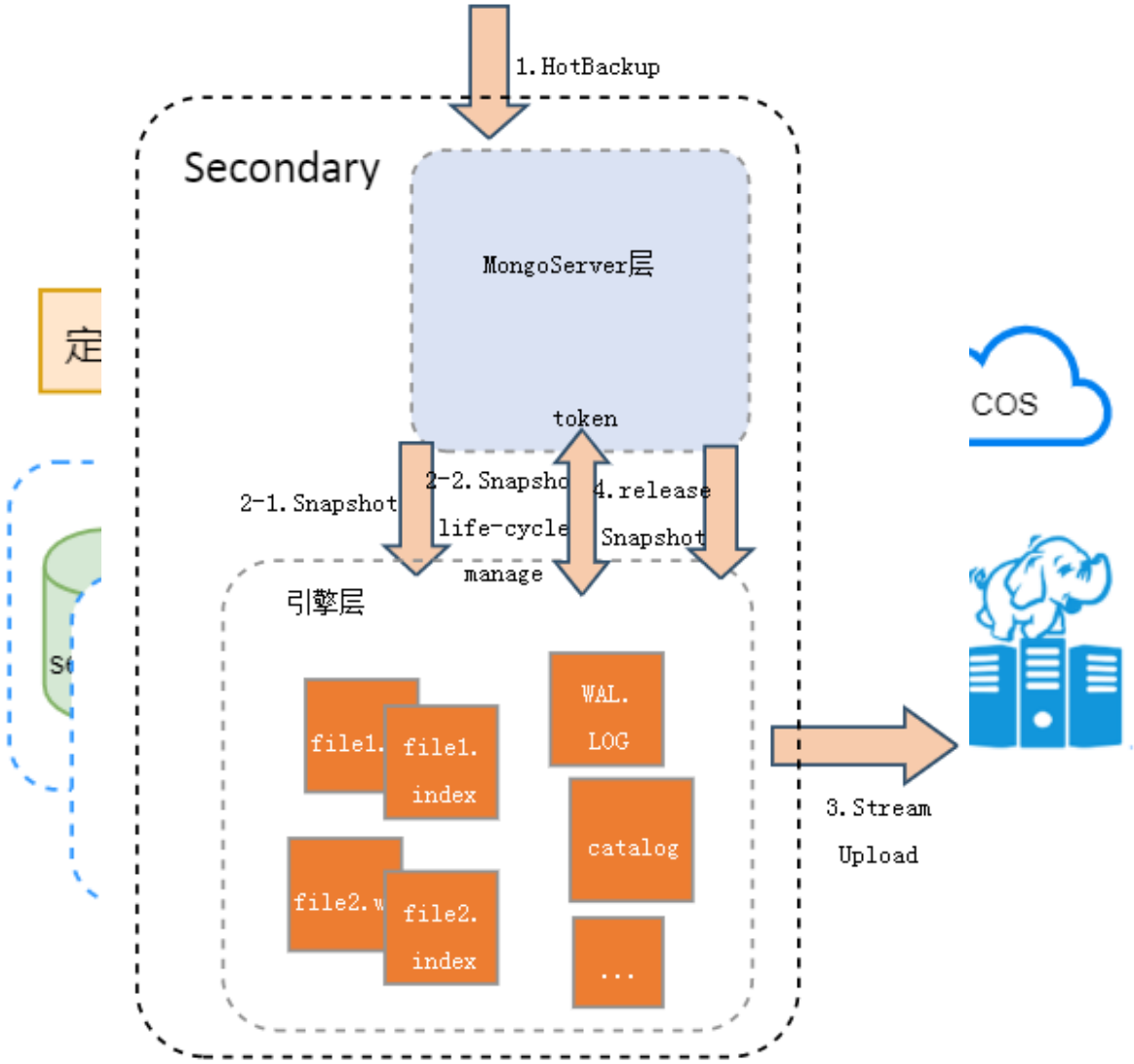
测试场景条件	迁移时长	迁移期间读写成功率
99%请求来自一个租户	117秒	100%
50%请求来自一个租户	125秒	100%
请求均衡来自所有租户	124秒	100%

Tencent

备份&回档

备份

- 逻辑备份 VS 物理备份
- 全量备份 VS 增量备份
- 整实例备份 VS 库表备份
- 全量备份间隔可调，流水持续备份，保证备份连续性



Tencent

备份&回档

回档

- 集群回档 VS 库表回档
- 可回档到最近14天内的任意时间点

难点

- 分片集群回档数据+路由
- oplog并行回放
- 回档与数据均衡互斥
- 备份可能有“脏”数据，需过滤
- 回档后重命名操作如何处理?

Tencent

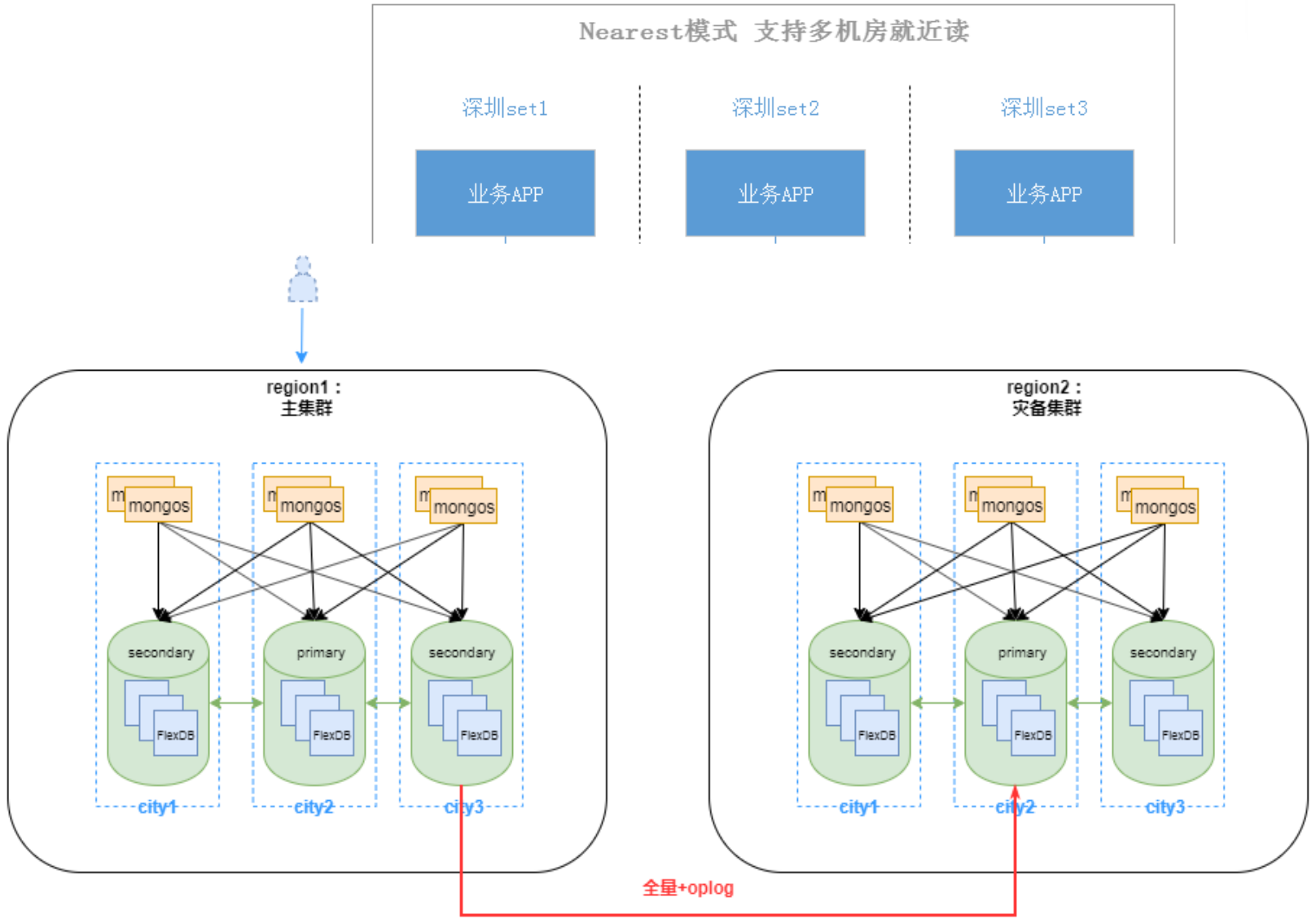
容灾

同城容灾

- 同城RTT 2~3ms
- 客户端就近接入
- CMongo装箱算法适配
- 强制跨3机房部署

灾备实例

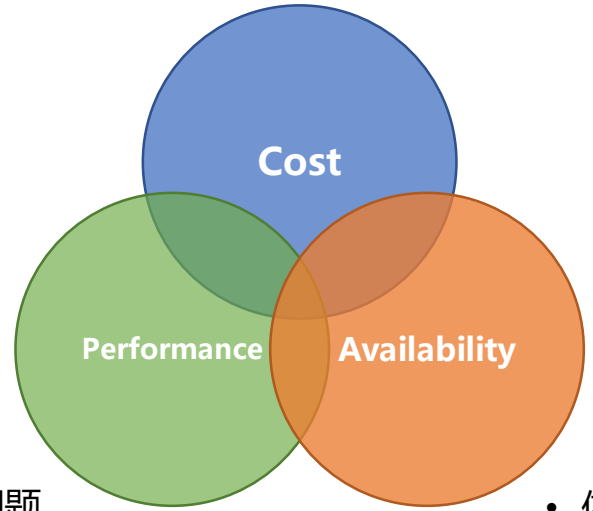
- 断点续传
- 同步进程高可用
- 同步秒级延迟
  - 全量数据按表并发
  - 增量oplog并行回放
  - 同步延迟阈值告警



Tencent

总结

- **多租户分DB**的模型，降低了单用户管理成本
- 通过**冷热分离**，降低存储成本



- 通过DB间**热迁移**，解决热点租户的问题
- **过载保护**，保证集群不会雪崩
- DB维度**隔离**，保证单个DB性能

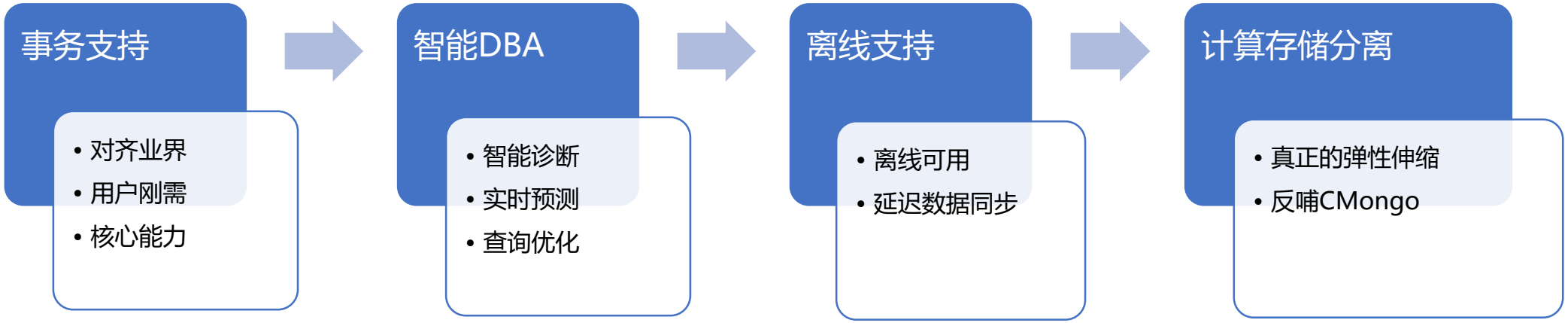
- 依托CMongo的3副本，故障时快速切换
- **跨机房容灾**，解决单机房故障导致的不可用
- **灾备**集群，保证区域性机房故障数据可用
- 定期备份，是保证数据安全的底线

Tencent

CONTENTS

Tencent

未来规划



Tencent

Q&A

demonlyang@tencent.com

腾讯云MongoDB

欢迎关注MongoDB中文社区

MongoDB中文社区网址：
http://www.mongoing.com/

MongoDB中文社区技术交流群：
1.请关注微信服务号
2.点击“社区互联”
3.选择 “联系我们”

MongoDB中文社区微信服务号+订阅号

mongoing.com