

MongoDB应用实践

李岑

主要内容

- 使用介绍
 - 场景、特性、部署、监控、变迁
- 问题分析
 - 通用问题、特定场景问题分析及优化
- Q&A

业务场景

- 主要业务：个人数据，如个人图片元信息、通讯录等；
- 数据类型：结构化可变字段数据；
- 使用方式：实时在线业务，多字段查询，范围查询等（OLTP）；
- 数据规模：近万亿条目，日近10亿写入；

业务需求

- 可靠性： 99.99999999%
- 可用性： 99.97%
- 扩展性： 支持快速数据扩展；
- 功能： CRUD + aggregate
- 其他： 吞吐性能、多业务管理、运维业务透明等；

Why MongoDB?

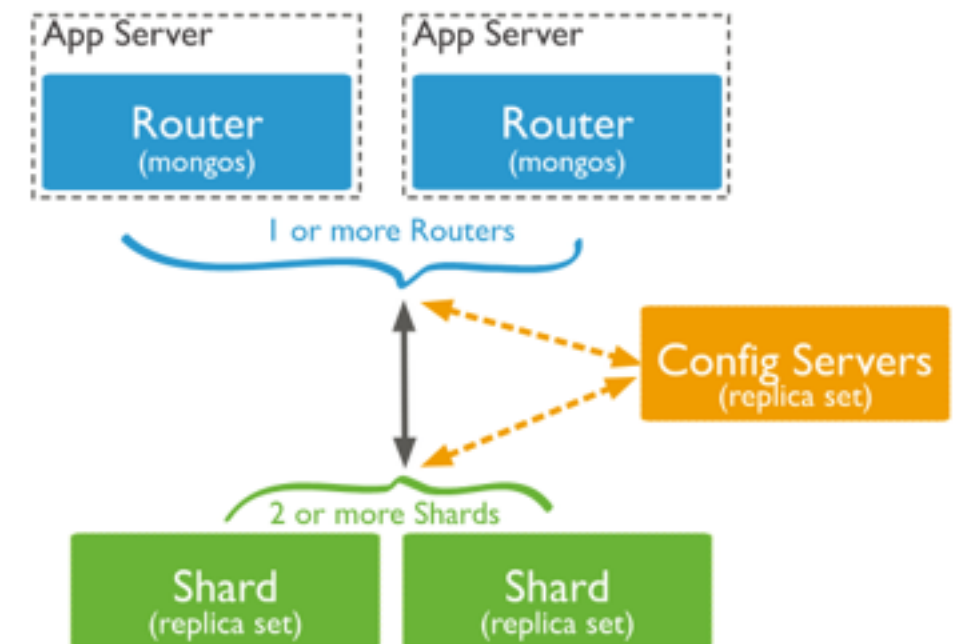
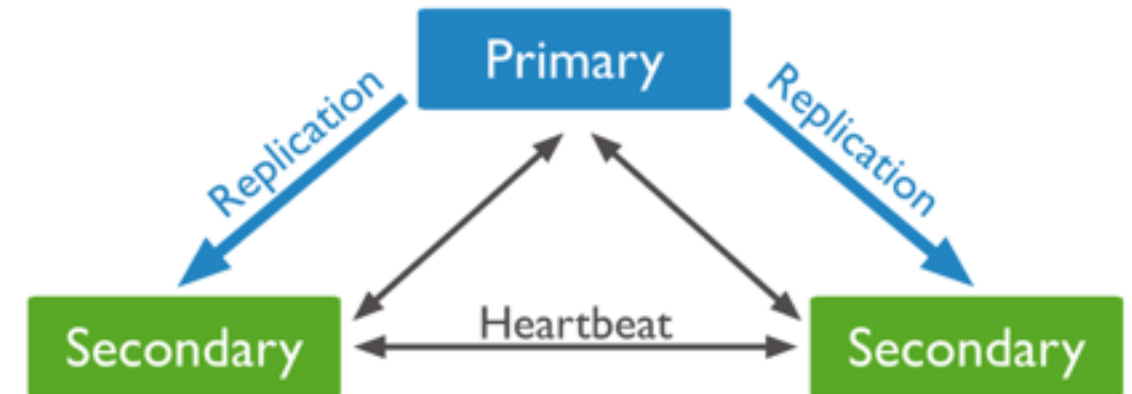
- 文档型 + schema-free
 - 文档型语意描述真实对象更自然
 - 数据建模相对简单，开发人员友好
 - 字段扩展，适合快速迭代开发

Why MongoDB?

- 完善的分布式方案，高可用、高可靠、维护成本低
 - 复制集：故障检测、故障切换
 - 集群化：数据分区、数据均衡
- 生态&社区
 - 完善生态环境（driver、部署、统计监控、平台）
 - 社区活跃

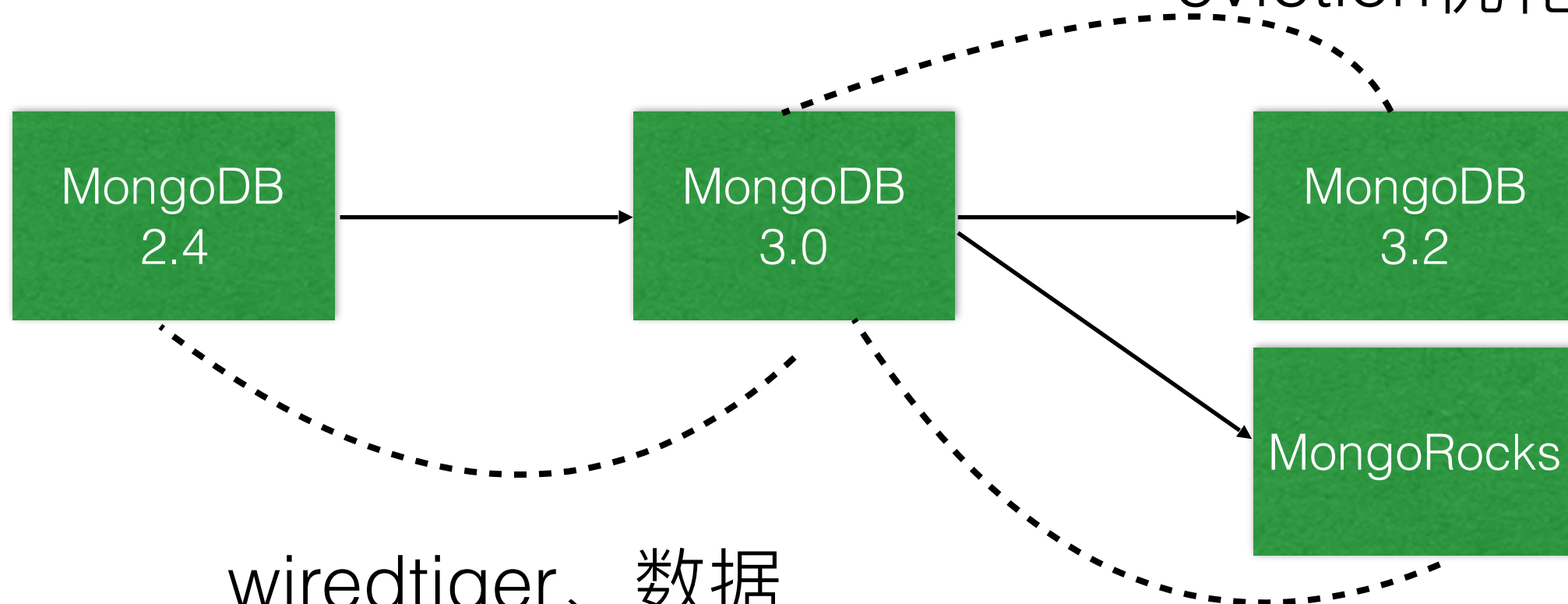
部署

- 硬件&系统
 - 12 core + 32G + 1.7T SSD
 - ext4
- 复制集
 - 1 + 2 + 2 (+1)
- 集群结构
 - mongos + configserver + shards



引擎

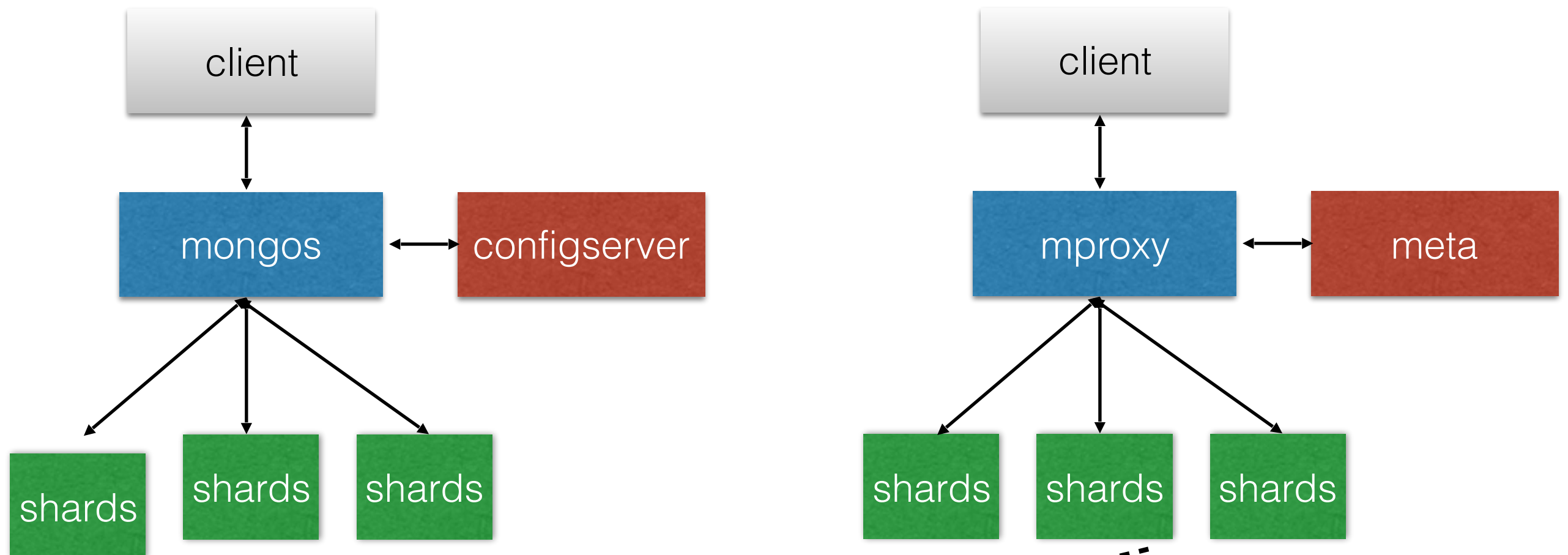
raft协议、
configserver复
制集、3.0 page
eviction优化



wiredtiger、数据
压缩、pluggable
storage engine

rocksdb引擎

架构演进



流控：流量封禁

流量管理：业务、分片纬度流量统计监控

连接管理：短长连接转换

容错：单库故障

扩容：性能 + 可管理性

外围建设

- 统计监控
 - 机器：cpu、mem、disk等；
 - 实例：请求qps、qr、连接数、主从延时等；
 - 业务维度：业务请求qps、业务用量审计；
- 运维管理
 - 部署平台、慢查询查杀、轮转升级/建索引、数据迁移/均衡；

shardkey的选择

- 范围分片
 - 优点： 范围查询；
 - 缺点： 请求热点， 如新用户段；
- hash分片
 - 优点： 数据&请求分布均衡；
 - 缺点： 范围查询性能差；

TIPS

- shardkey值空间尽量分散，避免值空间集中、有限；
- 避免shardkey热点，单shardkey数据过多容易导致单chunk太大（jumbo chunk）；同时也容易形成热点分片；

索引

- 索引使用优化
 - 冗余索引：清理冗余索引，节省空间；
 - 索引性能：根据业务请求合理选择索引字段及顺序；
 - 索引分析工具：dex(索引选择、平均耗时、长尾等)；
- 索引建立
 - 在线索引创建：锁 + iops
 - 离线索引创建：rolling index building

长连接or短连接

- 客户端请求流程：connect->auth->request->close
- 服务端服务模型：thread per connection
- 长连接
 - 优点：节省网络开销 + 线程开销
 - 缺点：容易导致不均衡，有状态（cursor cache）
- 短连接：
 - 优点：无状态，均衡
 - 缺点：性能低；

稳定性—连接爆炸

- 表现
 - mongos DoS：集群故障；
- 分析
 - mongos服务模型不感知客户端关闭动作，请求持续执行到完成；
 - 单分片变慢 + 客户端重试耗光mongos连接，扩散到整个集群；
- 优化
 - mongos扩容；
 - mongos实现后端连接池限制后端连接数（connection cursor）；

稳定性—慢请求

- 常见类型
 - scan
 - aggregation
 - large-skip query
 - 索引不合理
- killOp

稳定性 — mongod hang

- show dbs; dropDatabase等（禁止or离线操作）；
- 前台建索引（轮转建索引）；
- wrietiger page eviction（升级到3.2.10+）；

空间回收

- 类型
 - delete records: 非物理回收
 - drop collection: 非物理回收
 - drop database: 物理回收
- 处理
 - repair
 - draining moveChunk + dropDatabase

扩容

- 原生balance
- moveChunk + noCleanup
- draining moveChunk + dropDatabase
- paralyzed draining moveChunk + dropDatabase

冷热数据

- mongorocks + HDD
 - 低成本
 - 高写入、低读取性能
- 适用场景：
 - 冷数据存储
 - hidden secondary

Q&A