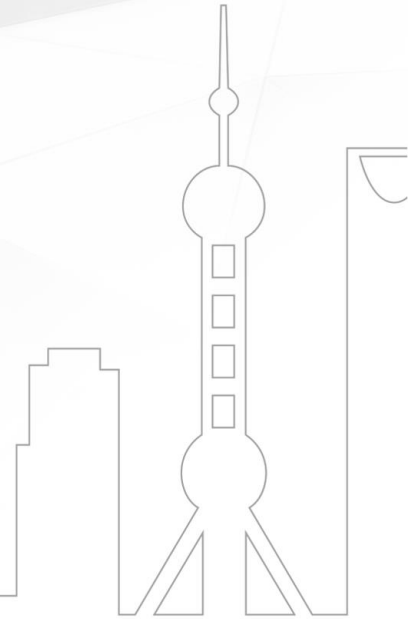
QCon[上海站]
全球软件开发大会2016

携程REDIS多数据中心实践

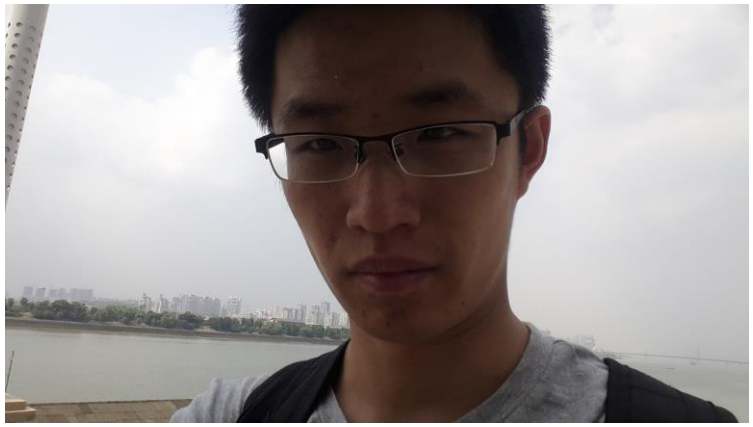
SPEAKER

孟文超



International Software
Development Conference

主办方 Geekbang. 极客邦科技 InfoQ



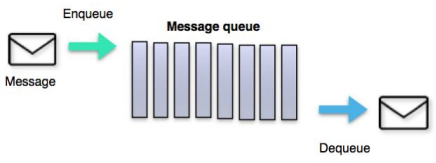
Ctrip 携程

大众点评 dianping.com

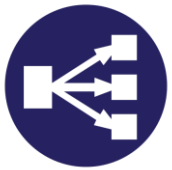
redis

宝信软件 BAOSIGHT

BAOSTEEL



Software Load Balance



广泛使用的REDIS

- Redis
  - 支持多种数据结构
  - 高性能、高可用
- 携程使用情况
  - 2000多组实例
  - 每秒访问量200万(写入10万)

REDIS多数据中心

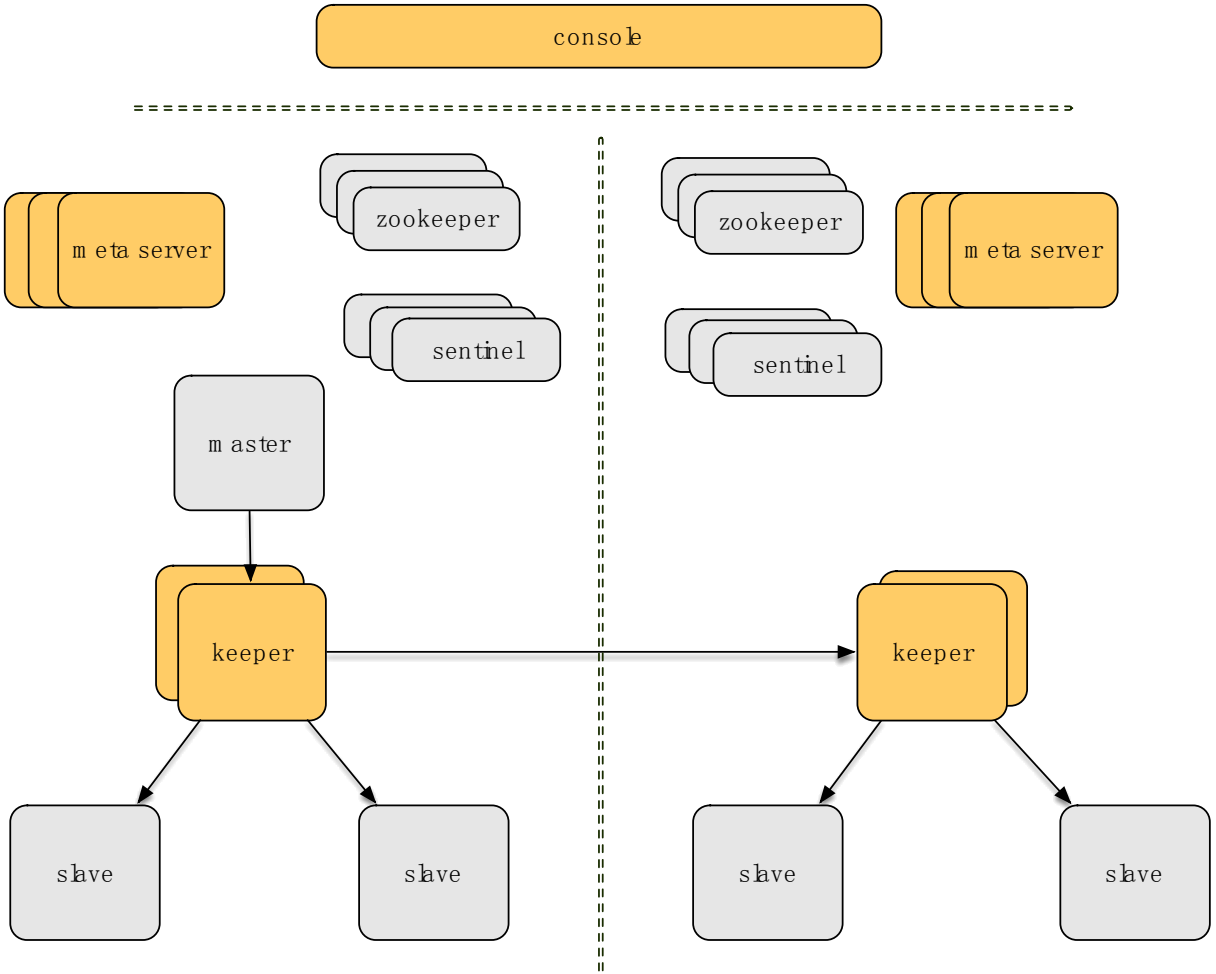
- 多数据中心
  - 业务压力
  - 高可用
- 多数据中心阶段
  - 备份
  - 双活
  - 多活

- Redis多数据中心?
  - Cache
  - 内存数据库

需要解决的问题

- 数据复制
  - 一致性
  - 实时性
- 高可用
  - 复制高可用
  - redis failover
- 机房切换
  - 反向复制

整体方案(X-PIPE)


console
meta server
zookeeper
zookeeper
meta server
sentinel
sentinel
master
keeper
keeper
slave
slave
slave
slave
控制流
数据流
X-Pipe
Others

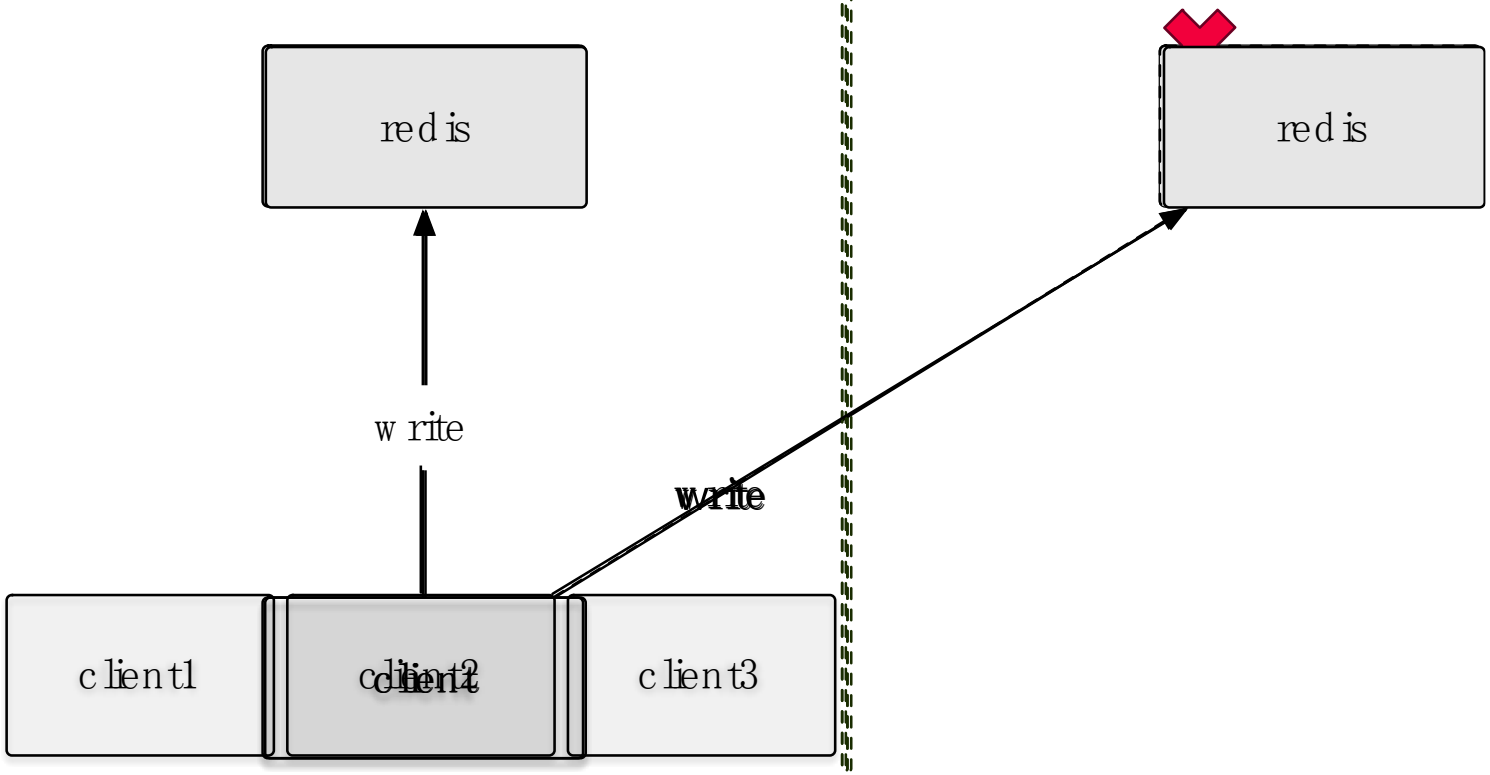
整体方案

- Console
  - 元信息配置、监控、报警
- Meta Server
  - Keeper Leader选举、状态切换
  - HA
- Keeper
  - Redis replication log数据管理
- Zookeeper
  - Meta server集群协调

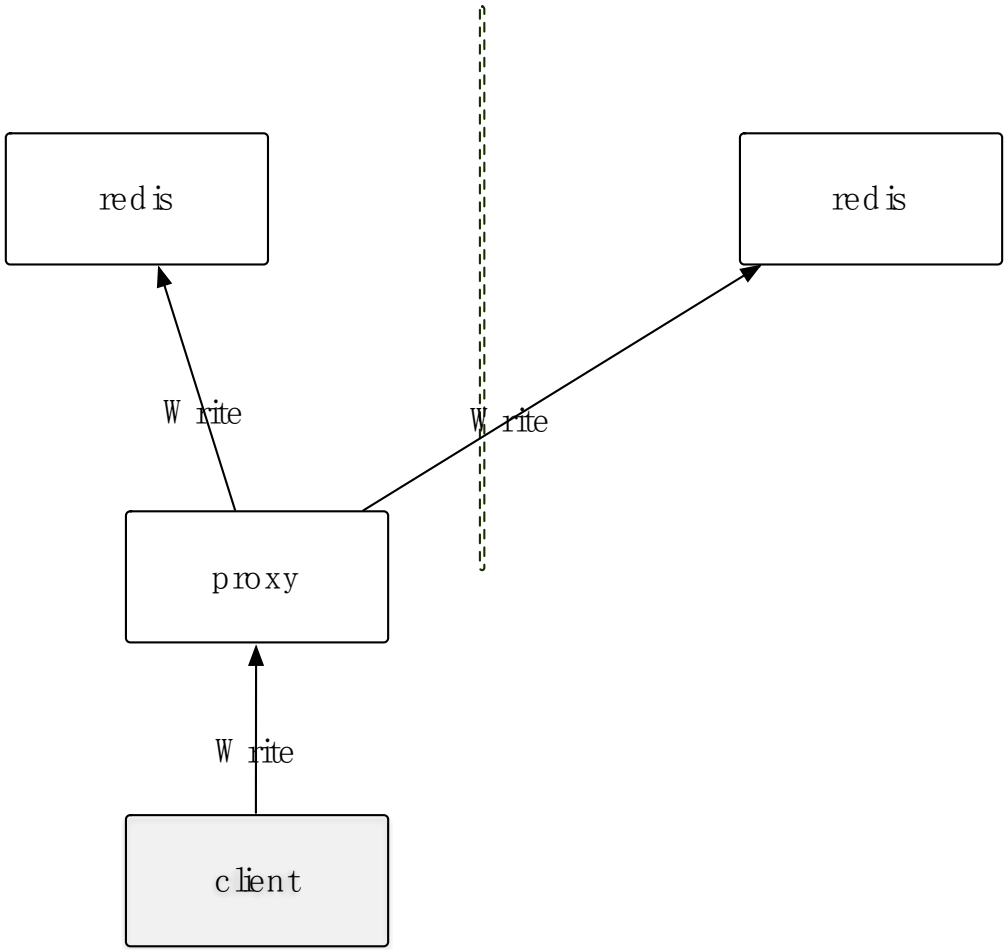
需要解决的问题

- 数据复制
  - 一致性
  - 实时性
- 高可用
  - 复制高可用
  - redis failover
- 机房切换
  - 反向复制

数据复制-双写?



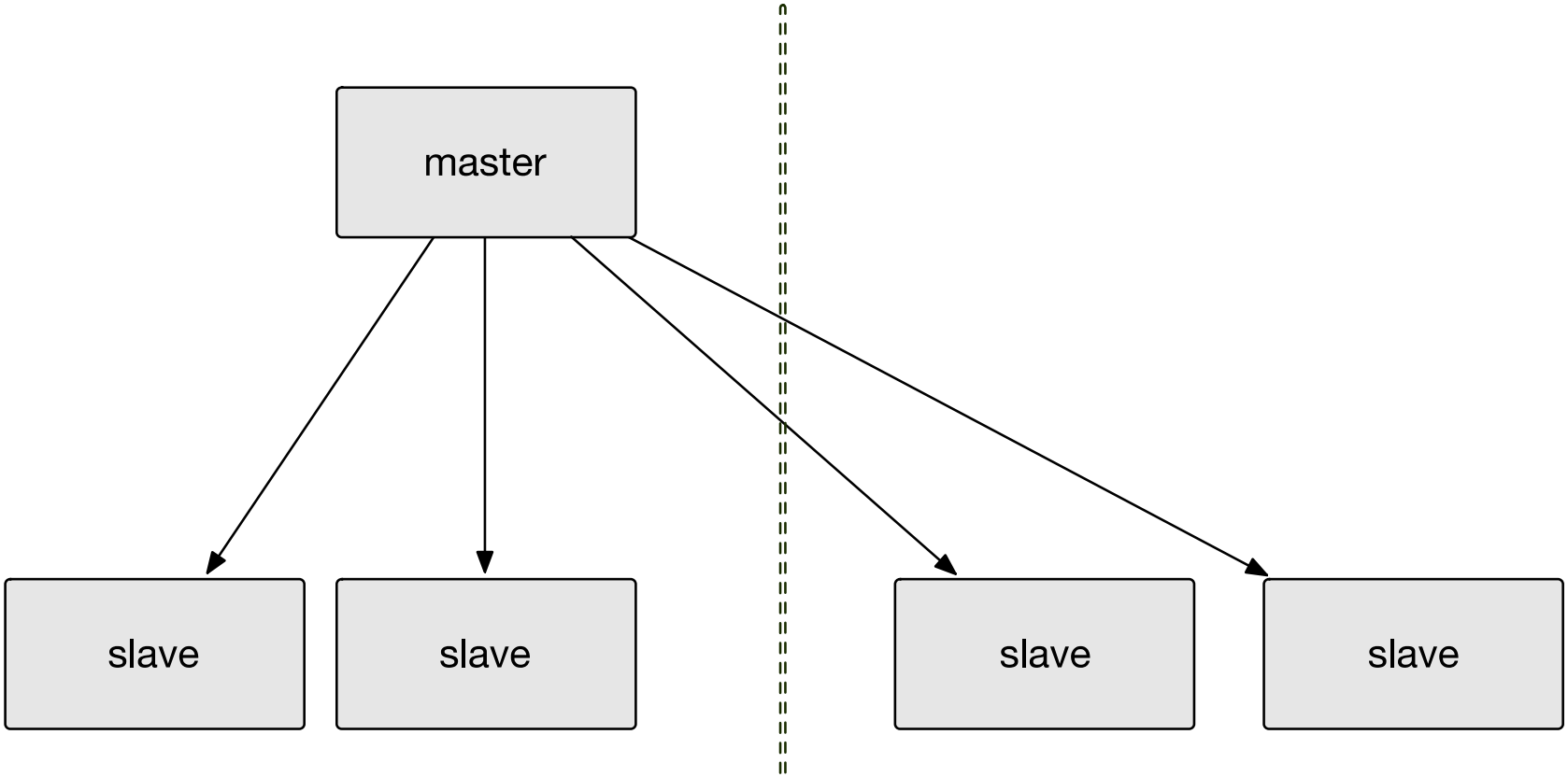
数据复制-PROXY



数据复制-PROXY

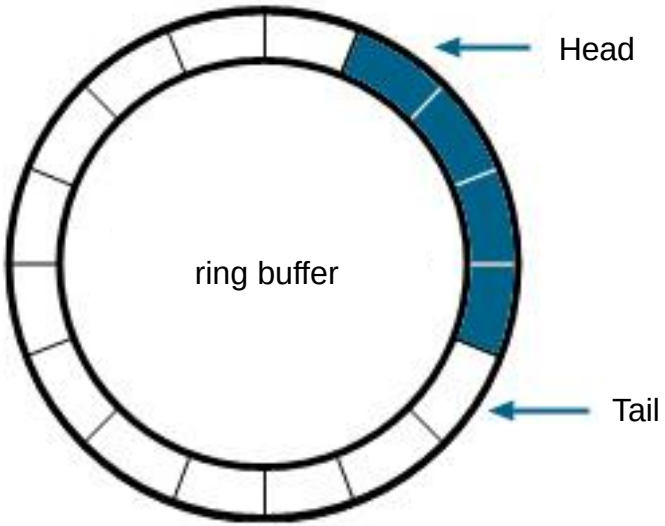
- 多个redis写同时成功?
- 特殊命令处理
  - 系统时间相关
    - time
  - 随机数
    - randomkey

数据复制-MASTER复制

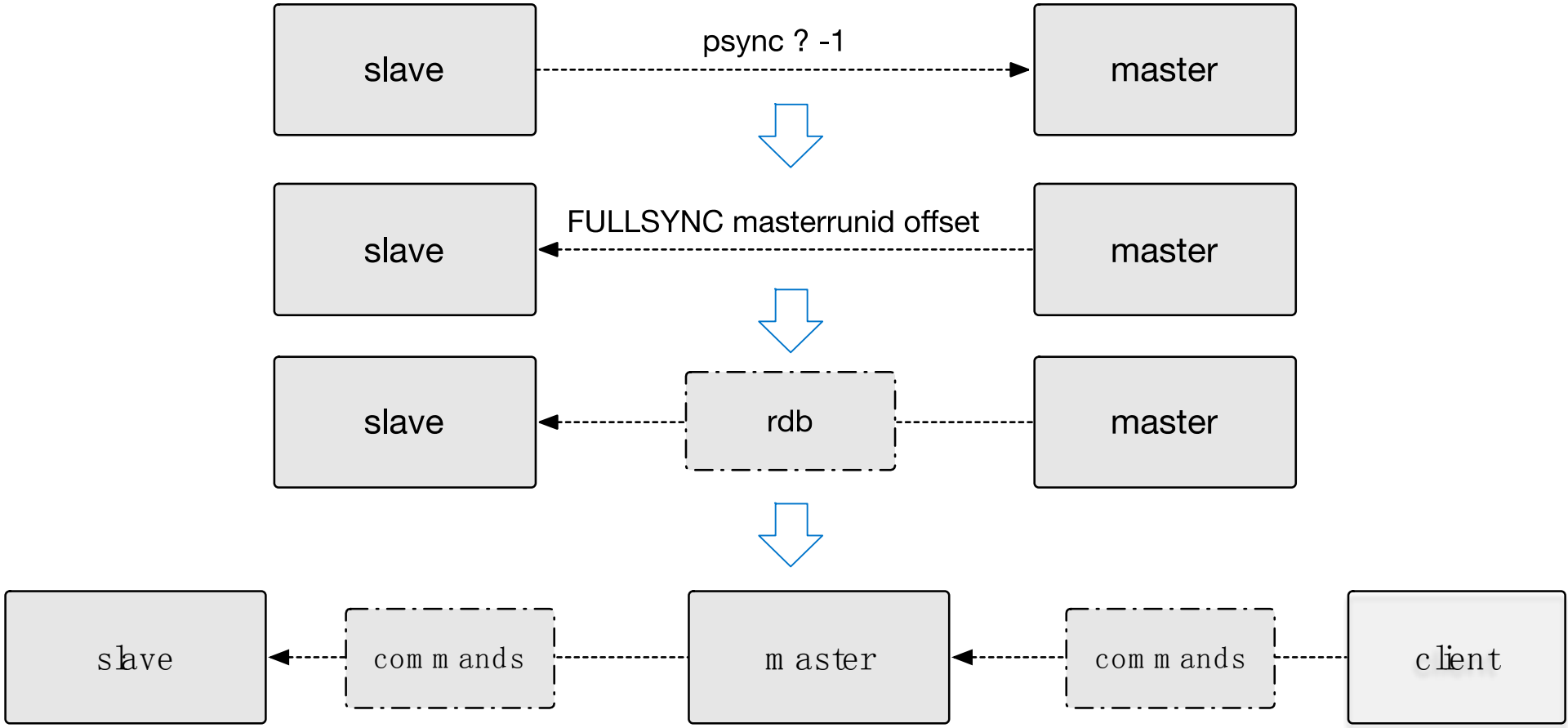


REDIS复制原理

- runid
  - redis运行时生成随机id
  - 重启之后变化
  - 标记唯一的redis进程
- RDB
  - redis内存数据映像
  - 可以导出到磁盘
- replication log(复制日志)
  - 增量命令日志
  - 内存
  - Ring buffer
- replication log offset(复制日志偏移量)
  - redis启动时设置
  - 随着写入命令增加而增加



REDIS复制原理-全量



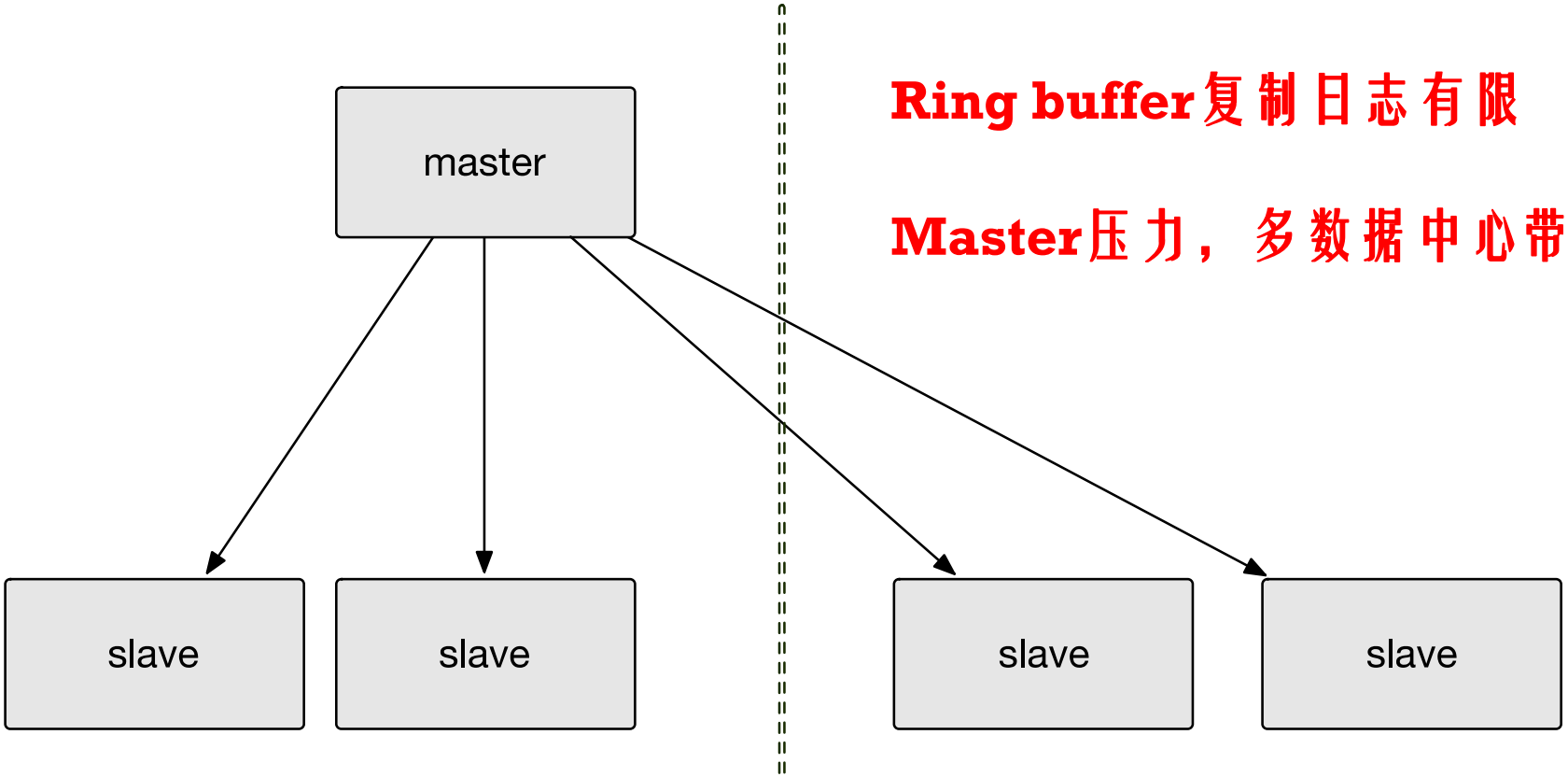
REDIS复制原理-全量

- 初次(全量同步)
  - PSYNC ? -1
  - FULLSYNC masterrunid offset
  - master导出rdb，向slave传输
  - slave接收rdb完成，加载数据至内存
  (**无法对外提供服务**)
  - 向slave传输replication log
- Slave不可用时间
  - 10GB内存**1-2min**

REDIS复制原理-增量

- 网络中断，重连(增量同步成功)
  - PSYNC masterrunid currentOffset
  - CONTINUE
- 网络中断，重连(增量成功失败)
  - 复制日志缺失
  - masterrunid不一致

数据一致性-MASTER复制？

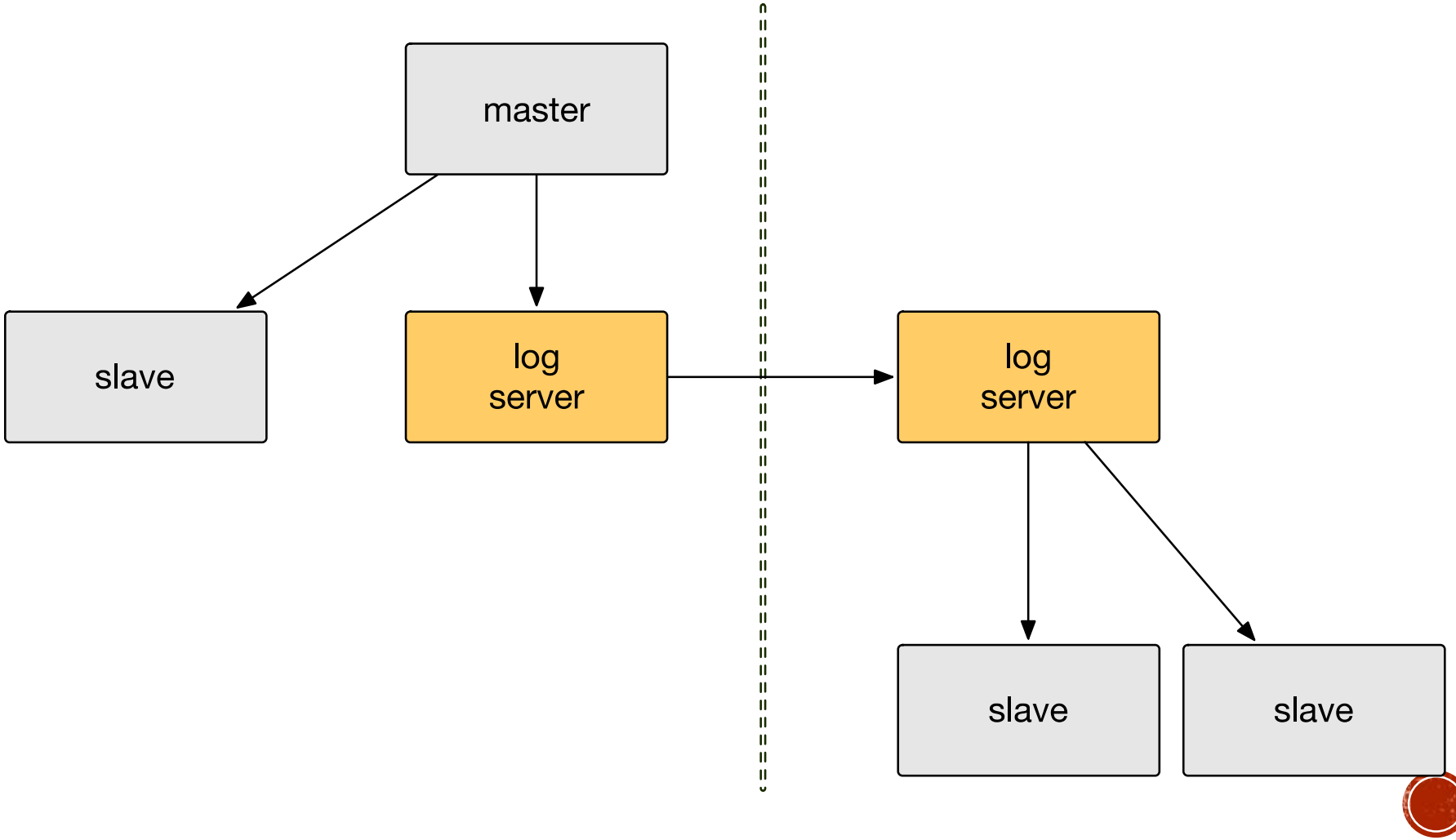


Ring buffer复制日志有限

Master压力，多数据中心带宽

复制日志磁盘保存?

数据复制-REPLICATION LOG SERVER

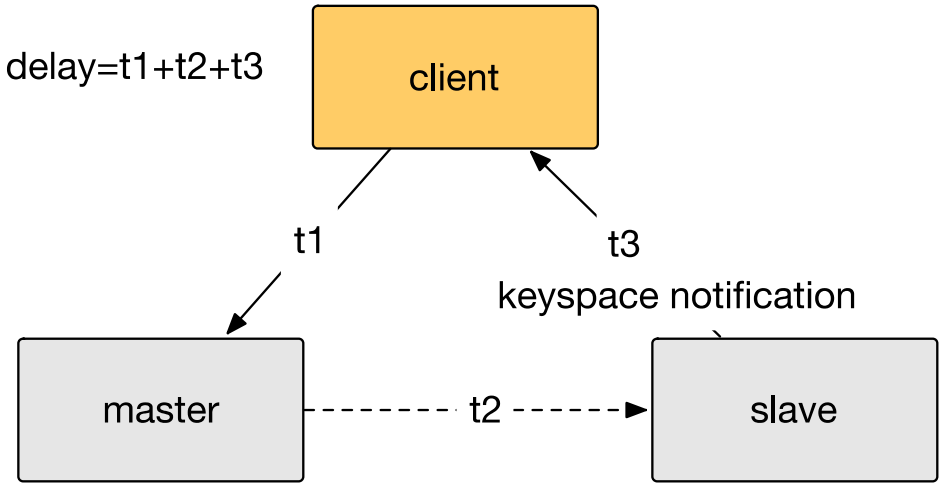


数据复制-REPLICATION LOG SERVER

- 磁盘缓存log数据
- 多机房Log server—log server数据传输
  - 数据中心内redis通过log server获取数据
  - 数据中心之间通过keeper复制
    - 压缩
- Replication log server
  - 命名：**keeper**

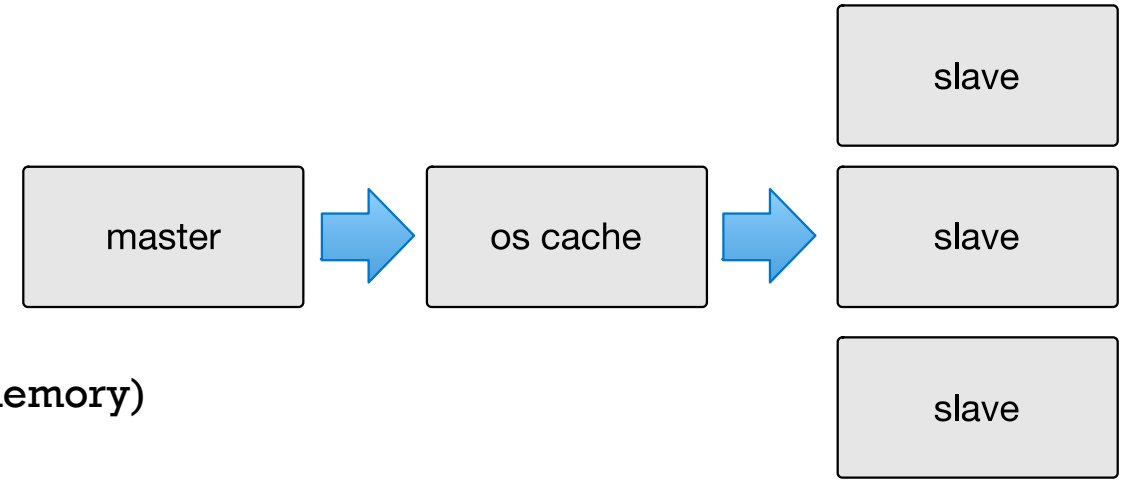
实时性

- 测试
  - 5w QPS，value 100字节
- 测试结果
  - master->slave 0.2ms
  - master->keeper->slave 0.3ms
- 跨机房延时
  - 同城几毫秒
  - 异地十几、几十毫秒



实时性

- 实现方案
  - 流式处理
  - 减少gc
    - Netty(pooled direct memory)
    - 操作系统(ZeroCopy)
  - 通知
    - JDK
      - AbstractQueuedSynchronizer
      - (volatile long)



需要解决的问题

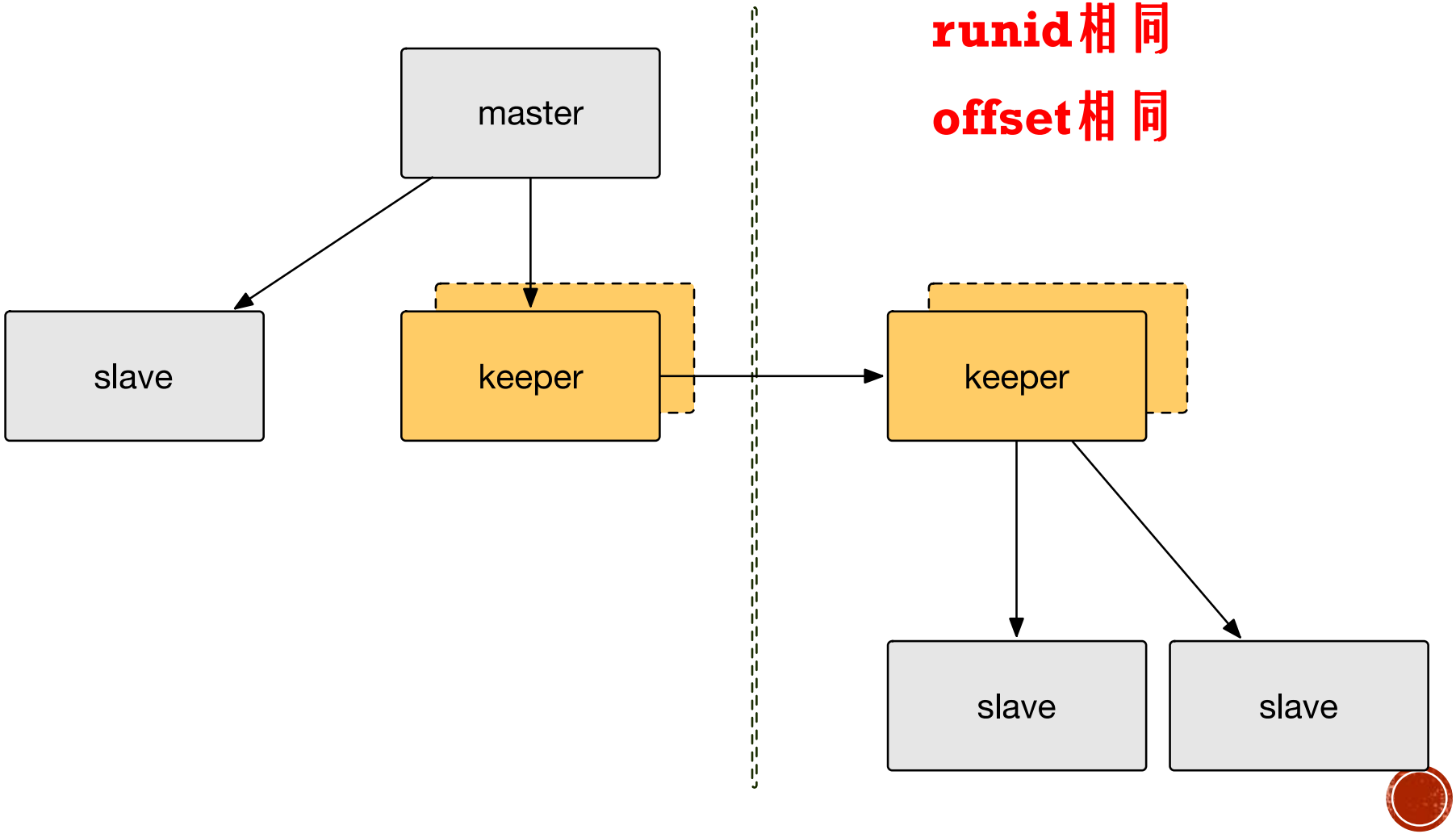
- 数据复制
  - 一致性
  - 实时性
- 高可用
  - 复制高可用
  - redis failover
- 机房切换
  - 反向复制

高可用-KEEPER

主备切换

runid相同

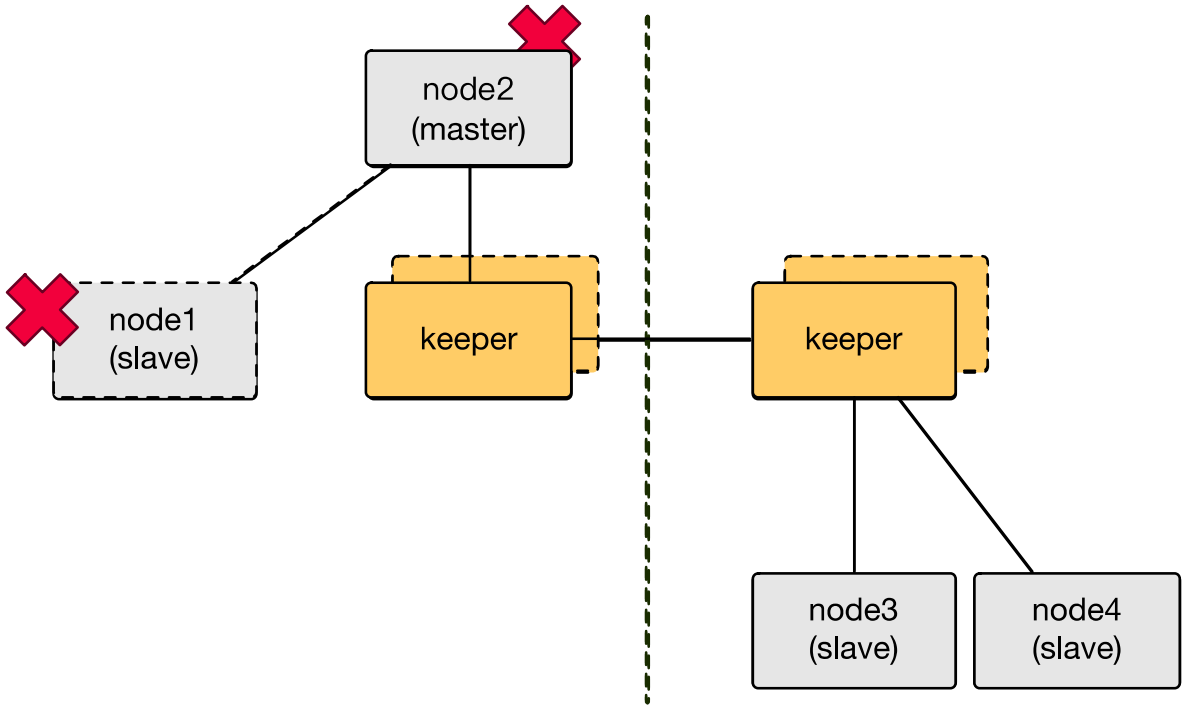
offset相同



高可用-META SERVER

- 功能
  - keeper主备选举
    - Keeper主备用(active/backup)
  - keeper管理
    - 增加、删除
- 多个meta server分片，负责不同的redis实例
  - meta server挂，负责的分片迁移
- Zookeeper
  - Meta server leader选举
  - Meta server分片信息存储

高可用-REDIS FAILOVER



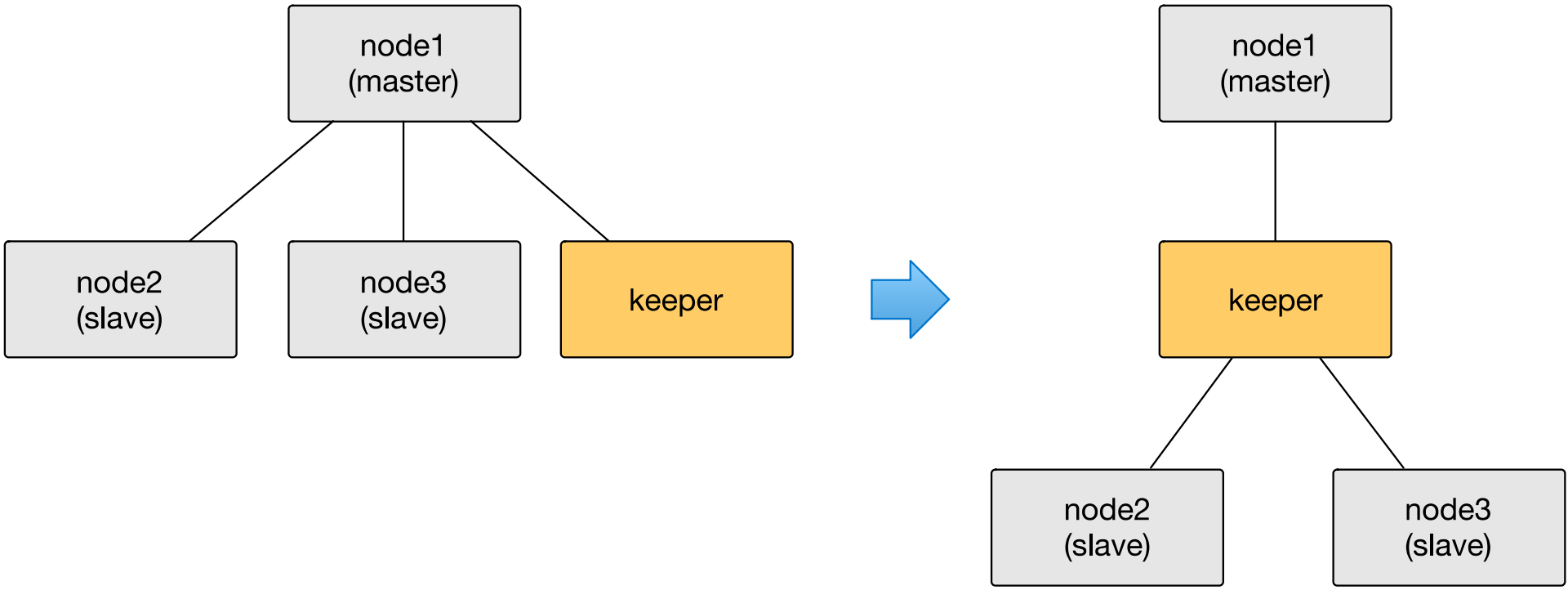
REDIS FAILOVER

- Sentinel
  - 高可用性解决方案
- 流程
  - Sentinel选择一个slave(node2)为新的master
  - Sentinel提升node2为master
  - Keeper从新的master(node2)复制数据
    - 全量同步

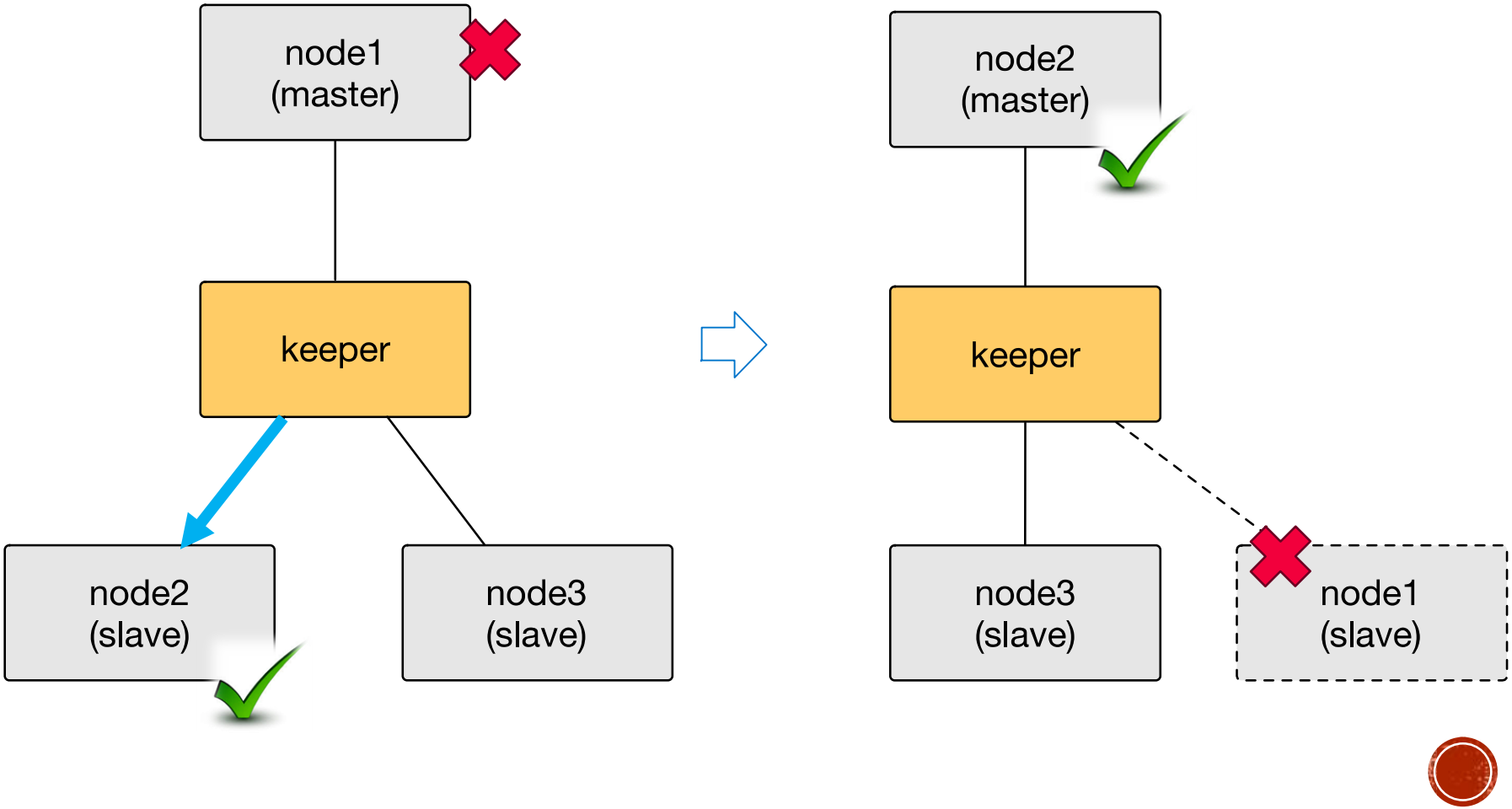
全量同步原因

- redis无全局offset
  - 官方解决方案https://gist.github.com/antirez/ae068f95c0d084891305
    - 全局replication id
  - 尚未实现
- Slave同步速度不一致
  - 同步慢的slave被提升为master

利用KEEPER实现增量



利用KEEPER实现增量



REDIS FAILOVER

- 假设需要提升node2为master
- 等待增量命令完全传输到node2
- keeper从node2继续增量同步
- node3无需任何变化

REDIS FAILOVER(REDIS源码修改)

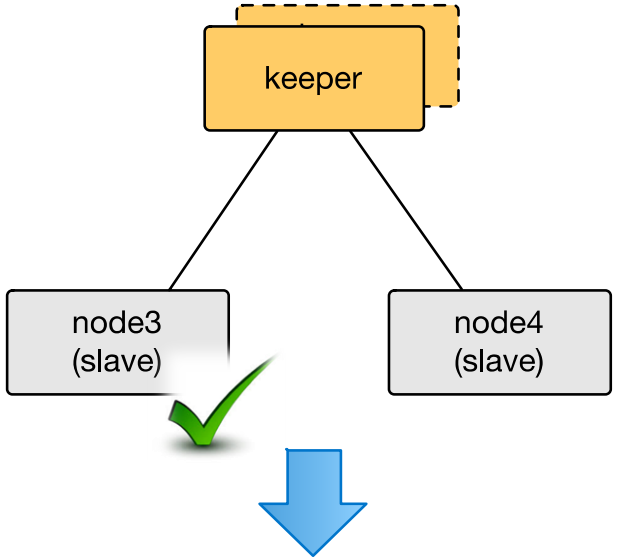
- Sentinel
  - keeper对sentinel透明
  - Setinel提升操作转发给keeper执行
- Redis
  - fsync
    - 只产生replication log

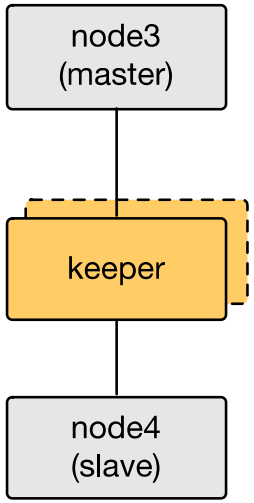
需要解决的问题

- 数据复制
  - 一致性
  - 实时性
- 高可用
  - 复制高可用
  - redis failover
- 机房切换
  - 反向复制

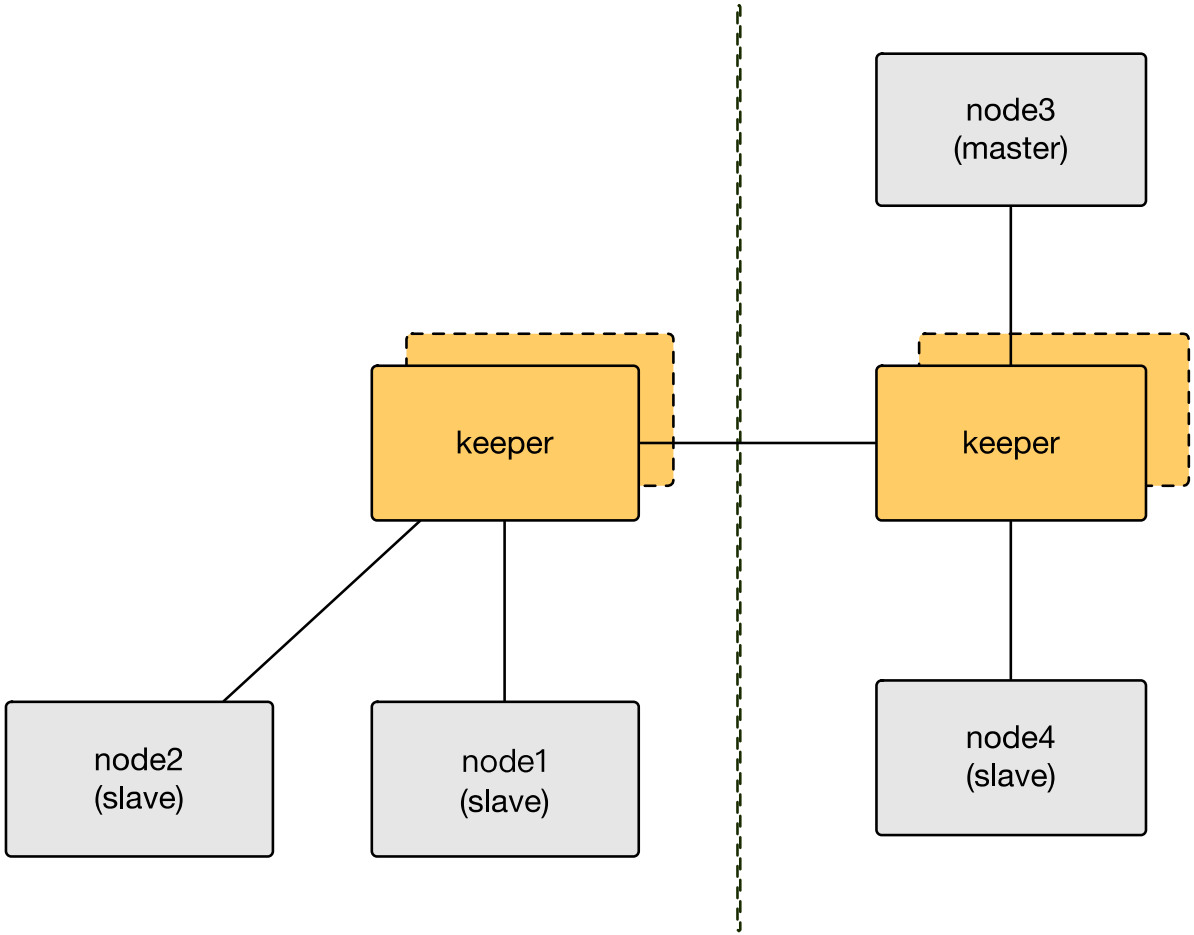
机房切换

- 切换机房
  - 将node3转化为master
  - keeper从node3同步数据
  - Node4不变

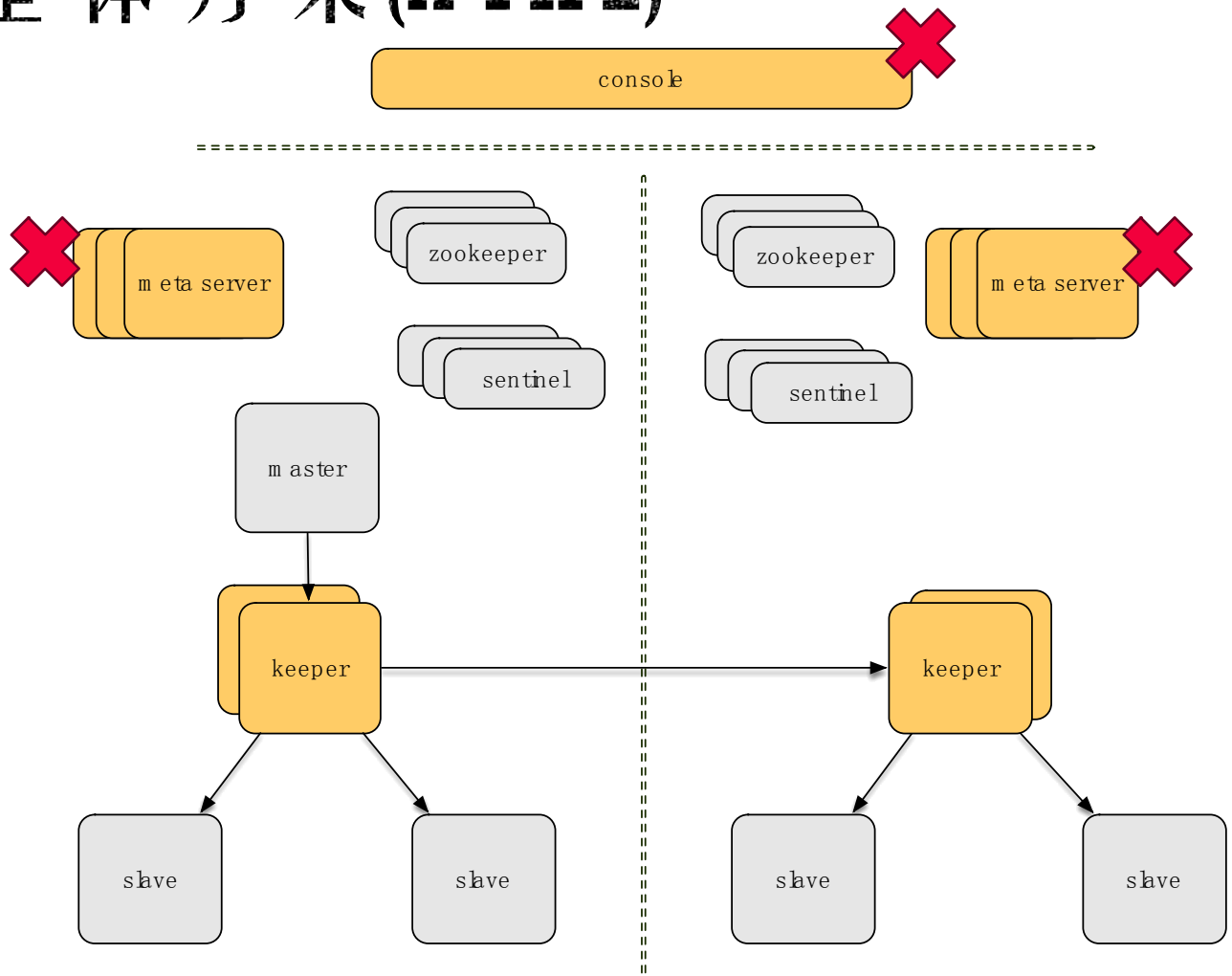




机房切换

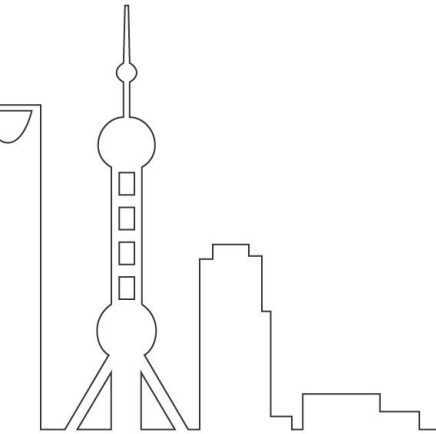

node3
(master)
keeper
keeper
node2
(slave)
node1
(slave)
node4
(slave)

整体方案(X-PIPE)


console
meta server
zookeeper
zookeeper
meta server
sentinel
sentinel
master
keeper
keeper
slave
slave
slave
slave
控制流
数据流
X-Pipe
Others

FUTURE

- 正在开源
  - https://github.com/ctripcorp/x-pipe
- Redis cluster支持
- keeper数据利用
  - 增量数据订阅
  - 拆分
  - 合并
- redis数据持久保存
  - rdb
  - aof(rewrite)



Thanks!

International Software Development Conference

Mail: oytmfc@gmail.com

主办方 Geekbang 极客邦科技 InfoQ