ArchSummit
全球架构师峰会2017

主办方 Geekbang 极客邦科技 InfoQ

PaxosStore：微信高可用、强一致存储系统

郑建军（RockZheng）

2017年12月8日

极客时间

重拾极客精神·提升技术认知

下载极客时间App

获取有声IT新闻、技术产品专栏，每日更新

扫一扫下载极客时间App

主办方 Geekbang. 极客邦科技 InfoQ

AiCon

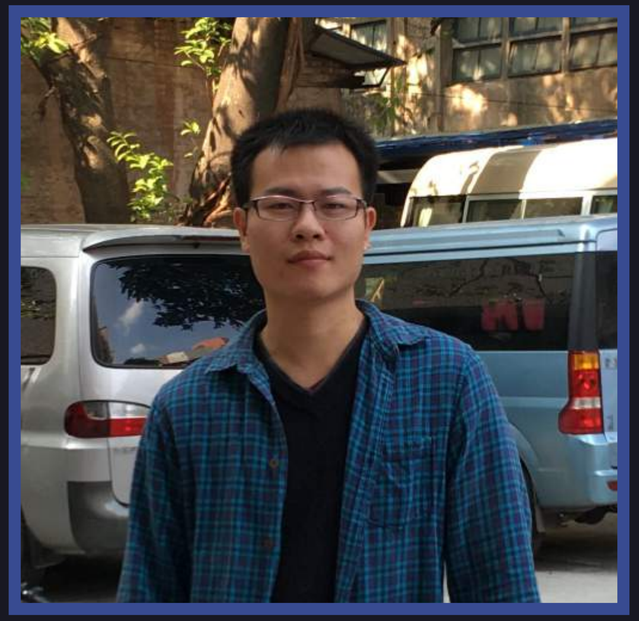
全球人工智能与机器学习技术大会

助力人工智能落地

2018.1.13 - 1.14　北京国际会议中心

扫描关注大会官网

SPEAKER
INTRODUCE



郑建军（RockZheng）

腾讯高级工程师

本人目前在腾讯负责微信基础存储的研发。14年加入微信后台团队，参与了多个大型分布式系统的架构设计和研发工作，其中作为微信核心存储PaxosStore主创人员之一，对微信核心存储系统（消息、朋友圈、好友关系链等）进行升级改造，提升了服务的可用性和数据的安全性。

TABLE OF
CONTENTS 大纲

- 项目介绍
- PaxosStore架构设计
- 具体案例分析
- 项目的挑战和发展

项目介绍

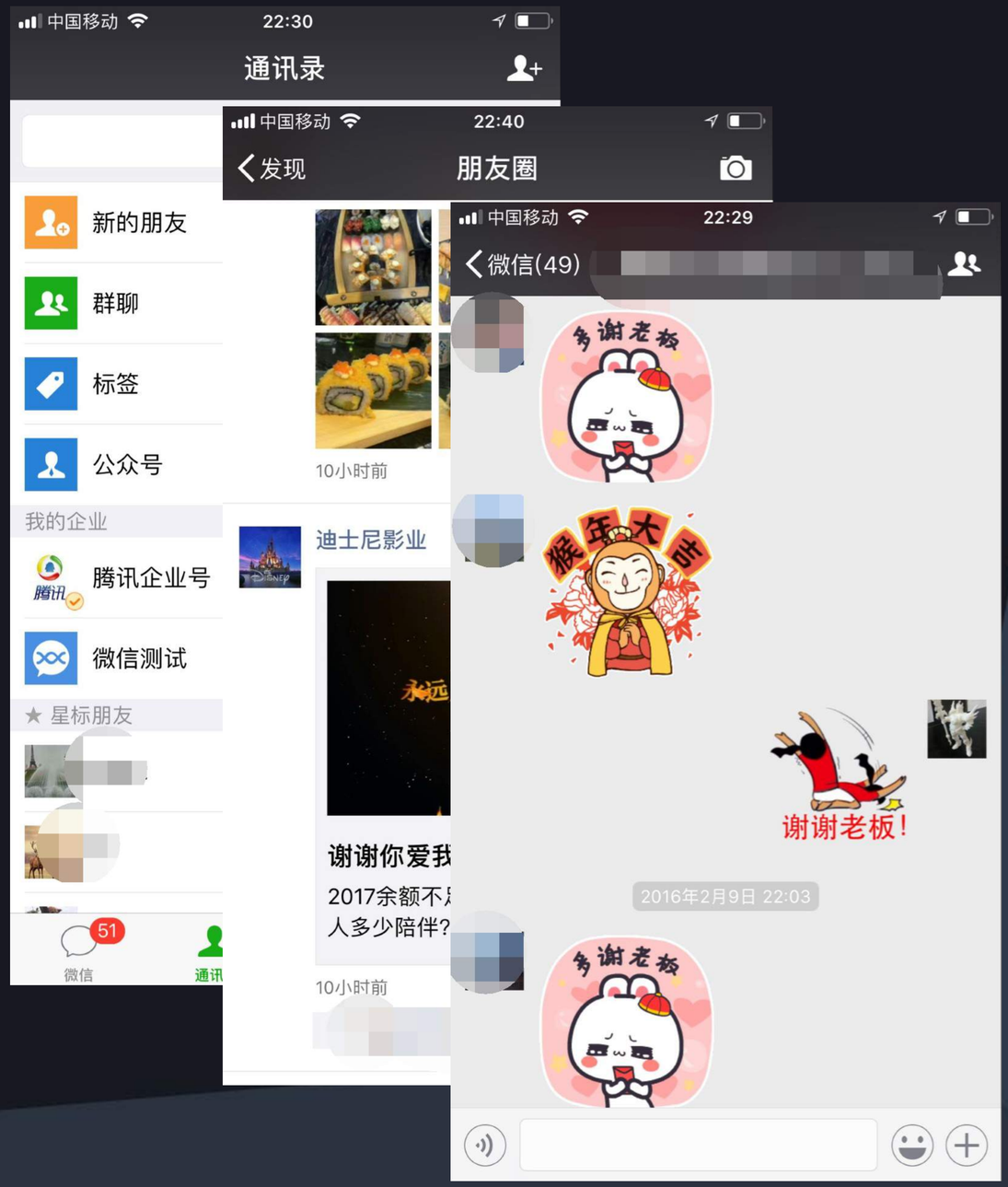
- PaxosStore是什么？

  一个在跨园区数据中心间同步复制，提供灵活的数据模式和访问接口，支持亿行大表，并具备快速伸缩能力，低延迟低成本，强一致和高可用的分布式存储系统。

- PaxosStore特点：
  - 无租约Paxos工程化，多主多写，高可用；
  - 针对业务特性优化，合并整体优化成本15+%；
  - 同一容灾、迁移框架下，支持多种插件化存储引擎、亿行大表；
  - 快速伸缩能力，基于反馈的自适应迁移系统。

项目介绍

- PaxosStore部署情况：
  - WXG内部广泛部署、数千台机器
  - 数万亿/天的读写量、峰值1亿+/秒
  - PB级的结构化数据、全球多个数据中心
- 微信业务支持：
  - 账号/消息/朋友圈/通讯录/...



为什么需要PaxosStore？

- 日益庞大的存储集群
  - 海量存储机器，PB级数据量
  - 月均上百台单机故障
- 旧架构缺陷：数据异步同步
  - 单机故障导致部分Key不可用
  - 单机数据丢失，导致未同步数据丢失
- 需求列表
  - 提高服务的可用性
  - 提高数据的安全性

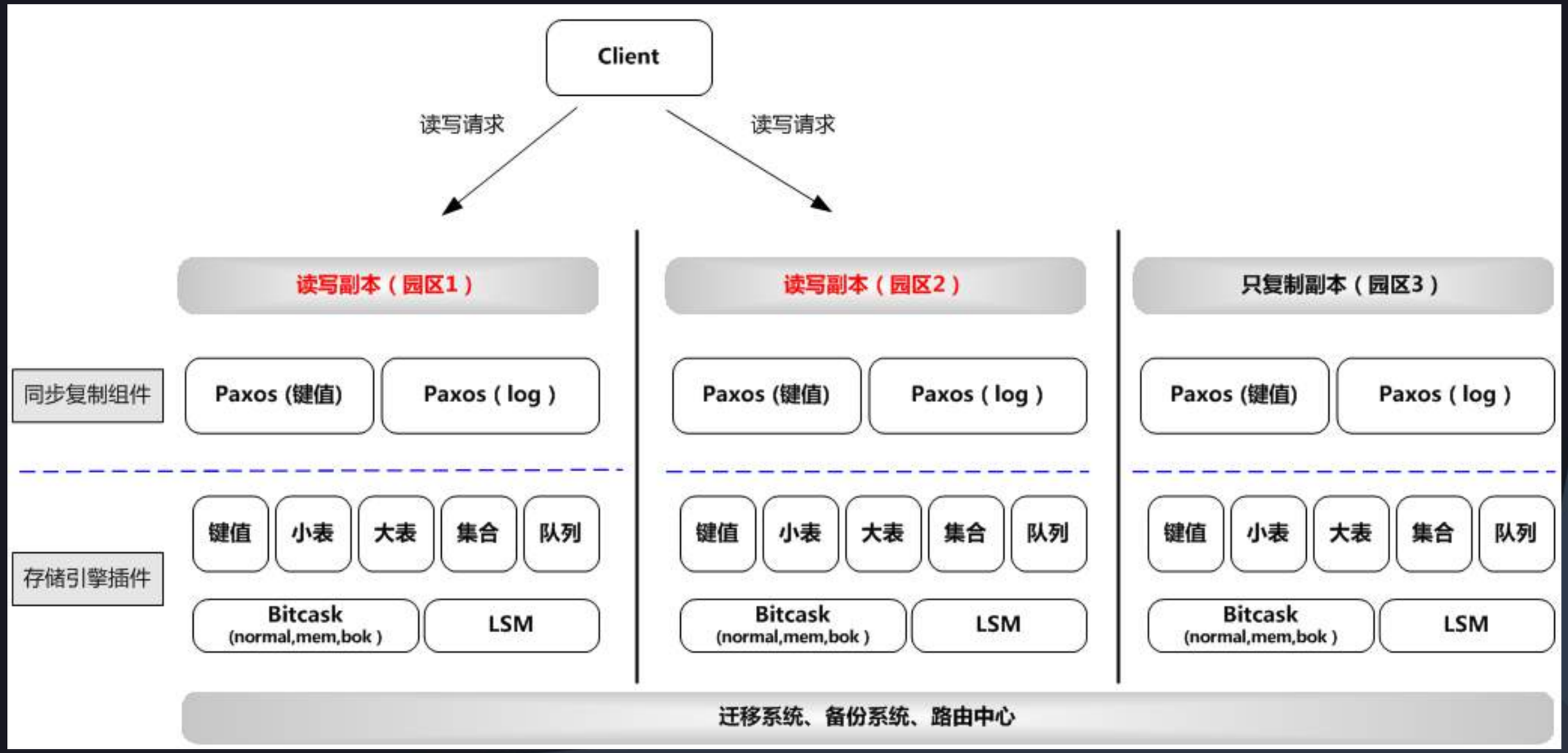
月份	4月	5月	6月
台数	100	176	200

机器规模7000台，每月频繁的单机故障

TABLE OF
CONTENTS 大纲

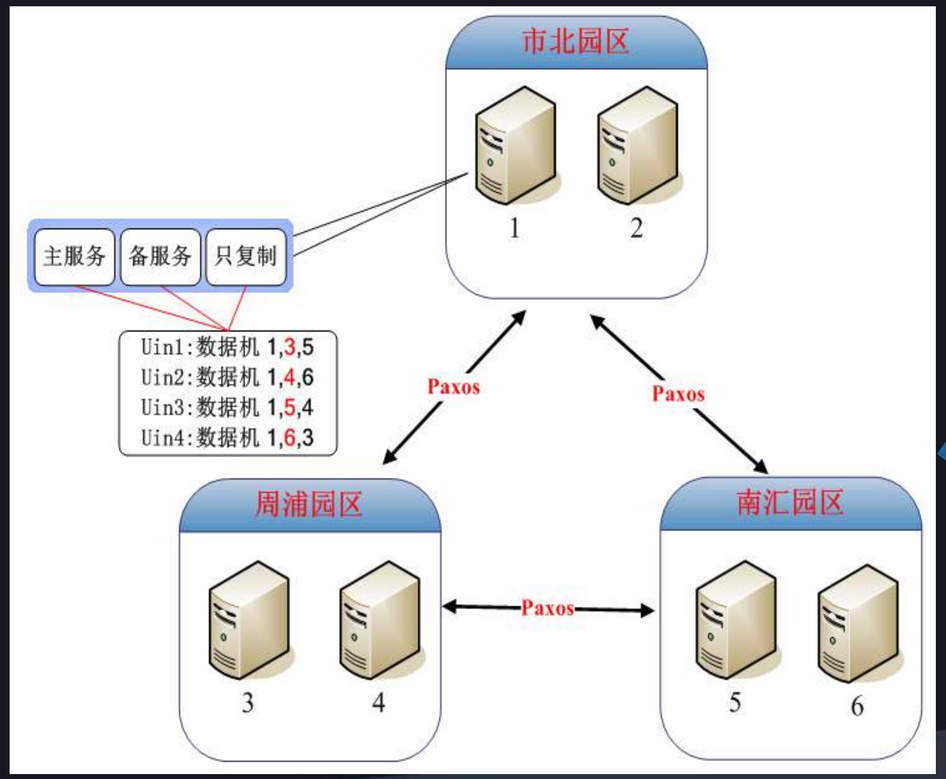
- 项目介绍
- PaxosStore架构设计
- 具体案例分析
- 项目的挑战和发展

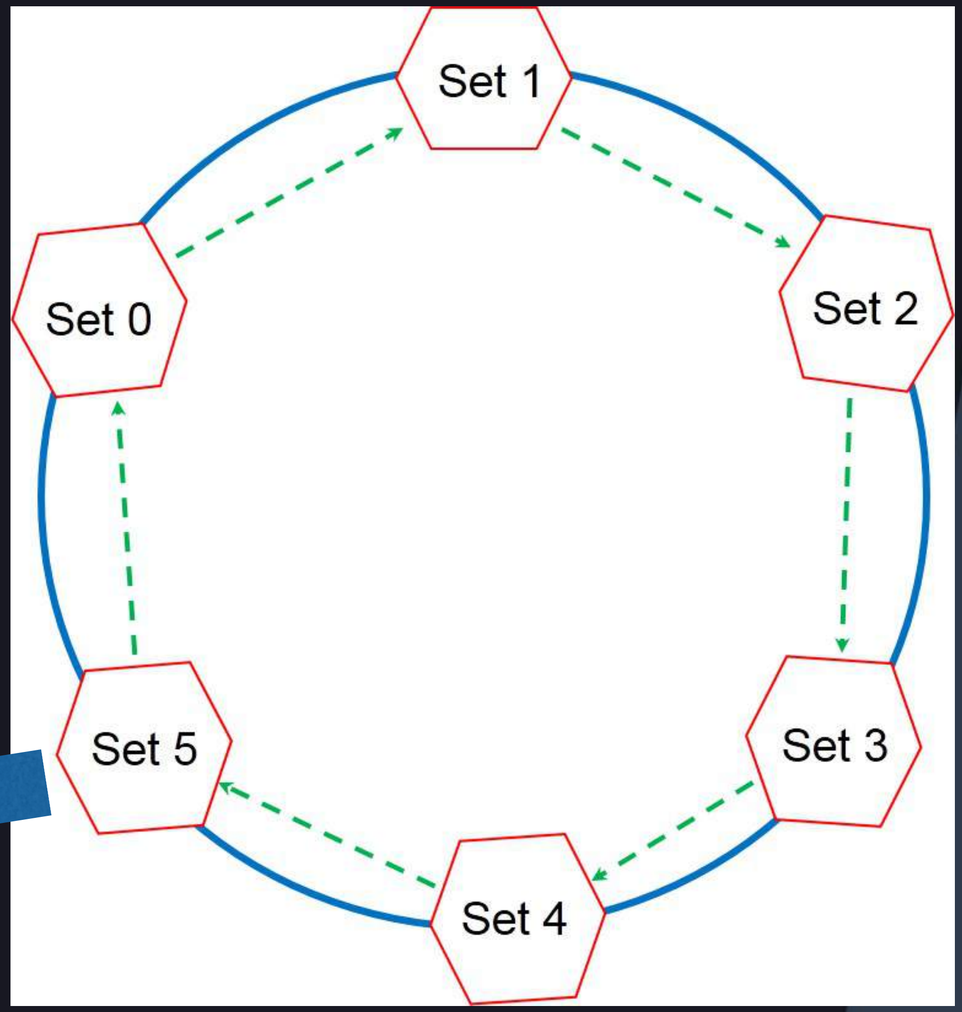
PaxosStore整体架构



数据分布方式

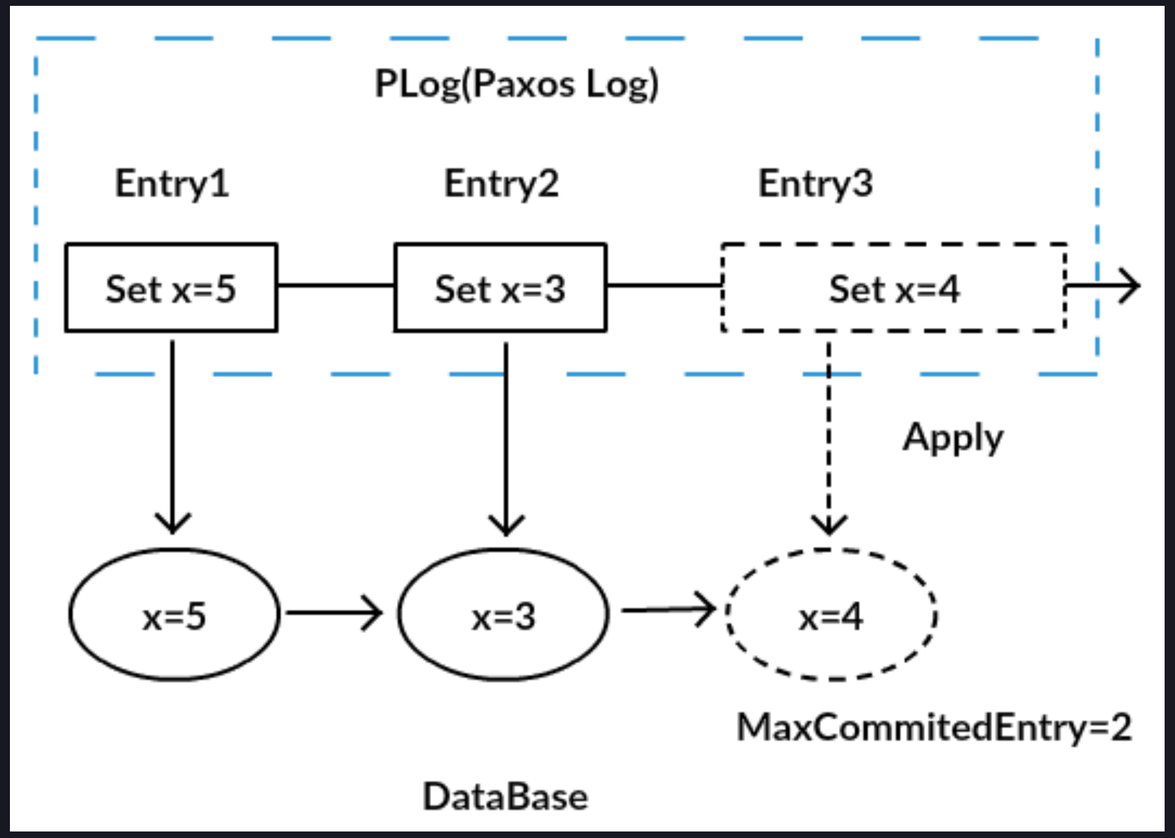
- 按Set隔离，一致性哈希；
- Set内三园区容灾；
- 读操作就近访问。

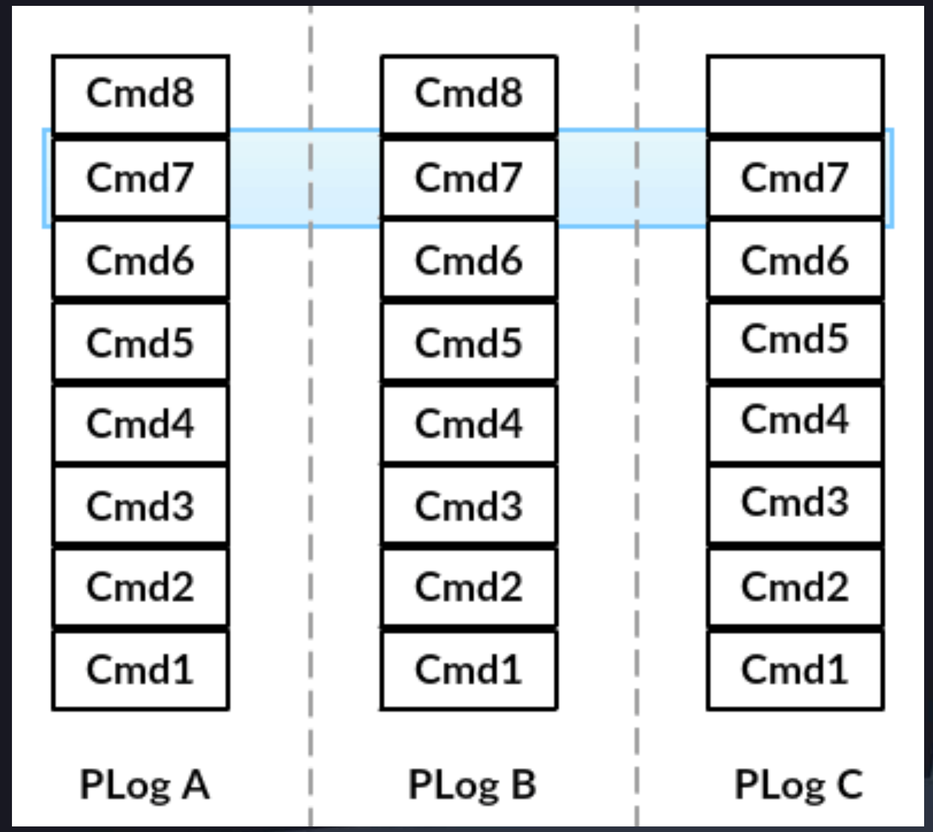




PaxosLog（PLog）介绍

- Paxos协议保证PLog中每个Entry的一致性

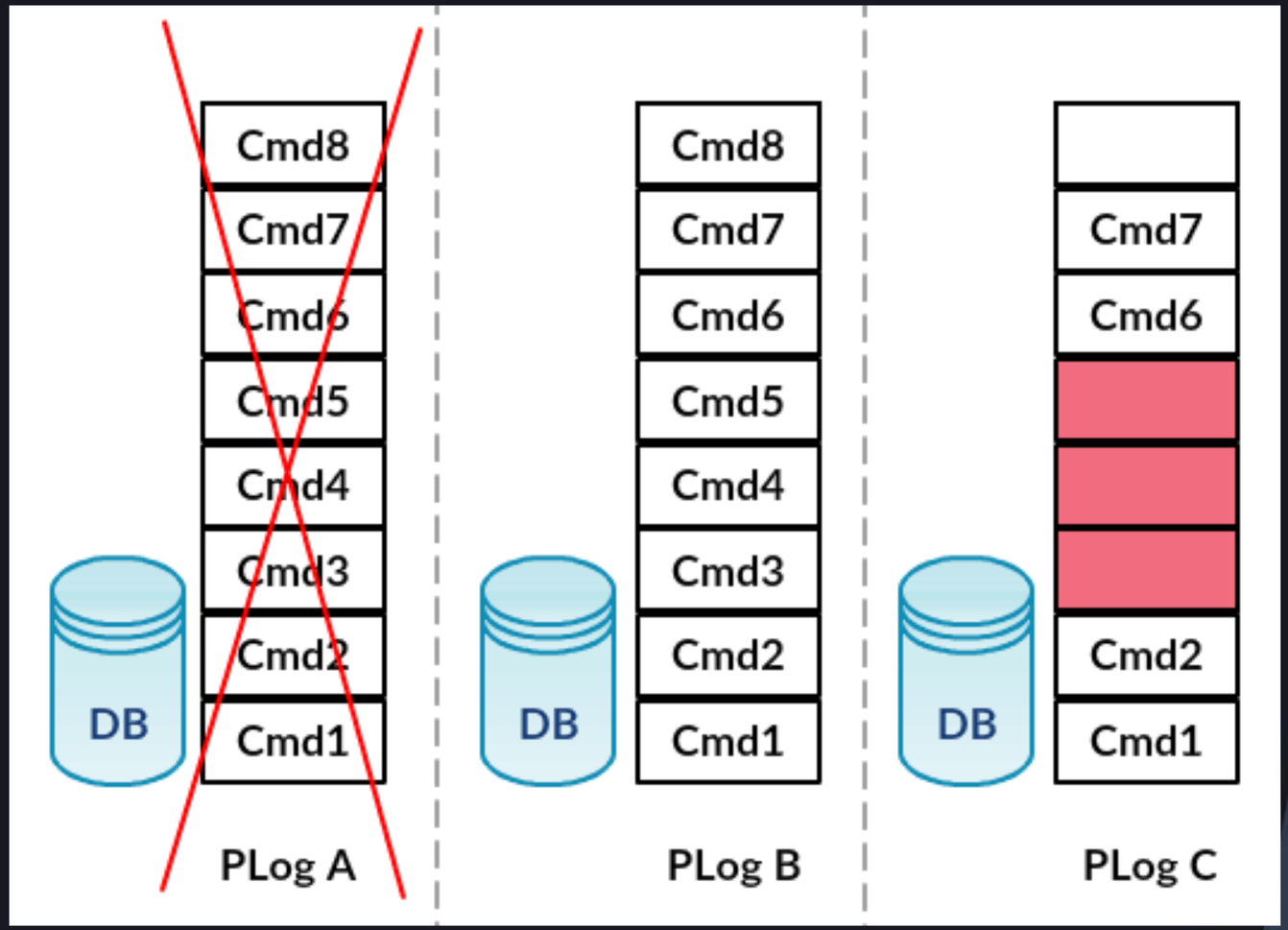




ABC三台机上对应的三条PLog

三机PLog架构设计要点

- 三份数据副本
  - *更加安全可靠*
  - *NoRaid代替Raid10降低成本*
- 多主多写
  - *AB机自动跳转*
- PLog允许空洞
  - *可用性更高*
- PLog细粒度化
  - *用户数据相互隔离*
- 快消型PLog
  - *消除磁盘瓶颈*



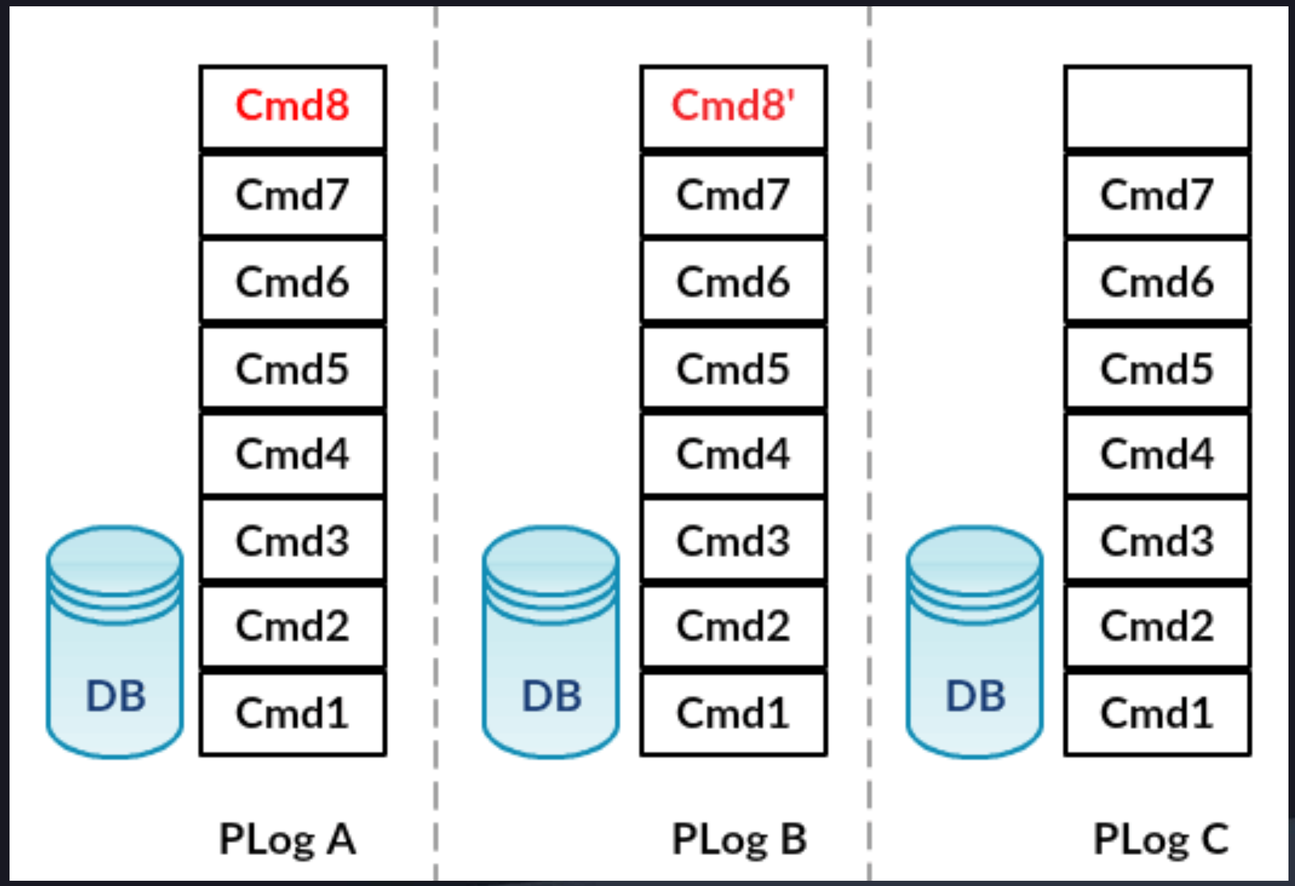
三机PLog架构

三份数据副本下的性能优化

- Paxos协议优化
  - 预授权（Pre-Preparing）
- 常规优化
  - 网络IO按IP聚合投递
  - 磁盘IO合并
- 针对性优化
  - 只复制副本，延迟批量Commit

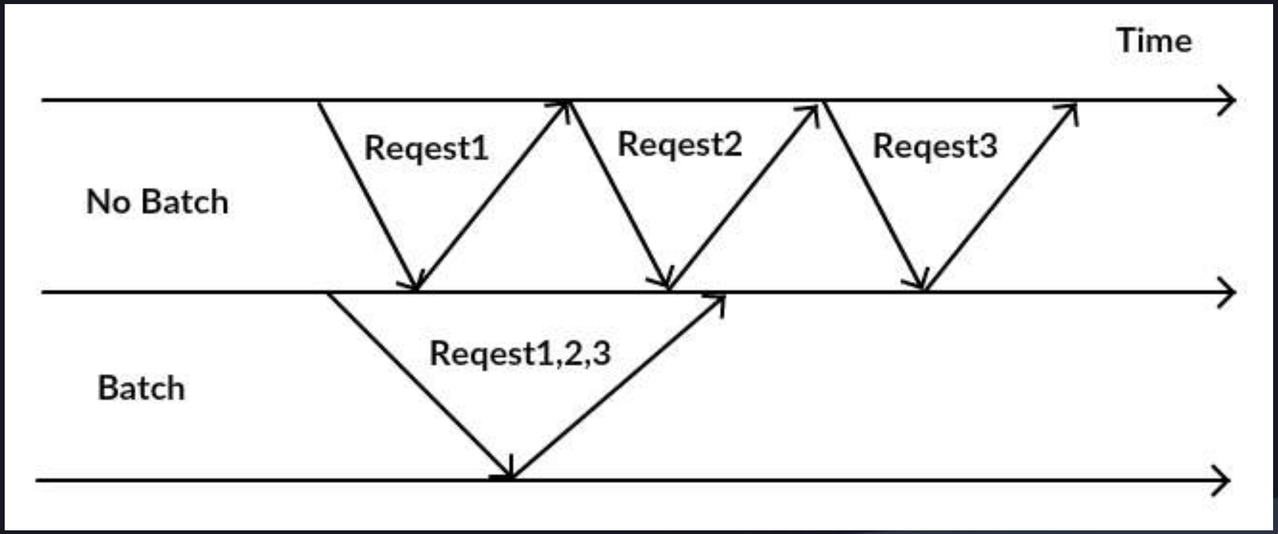
多主多写下的活锁问题

- 活锁问题是怎么产生的？

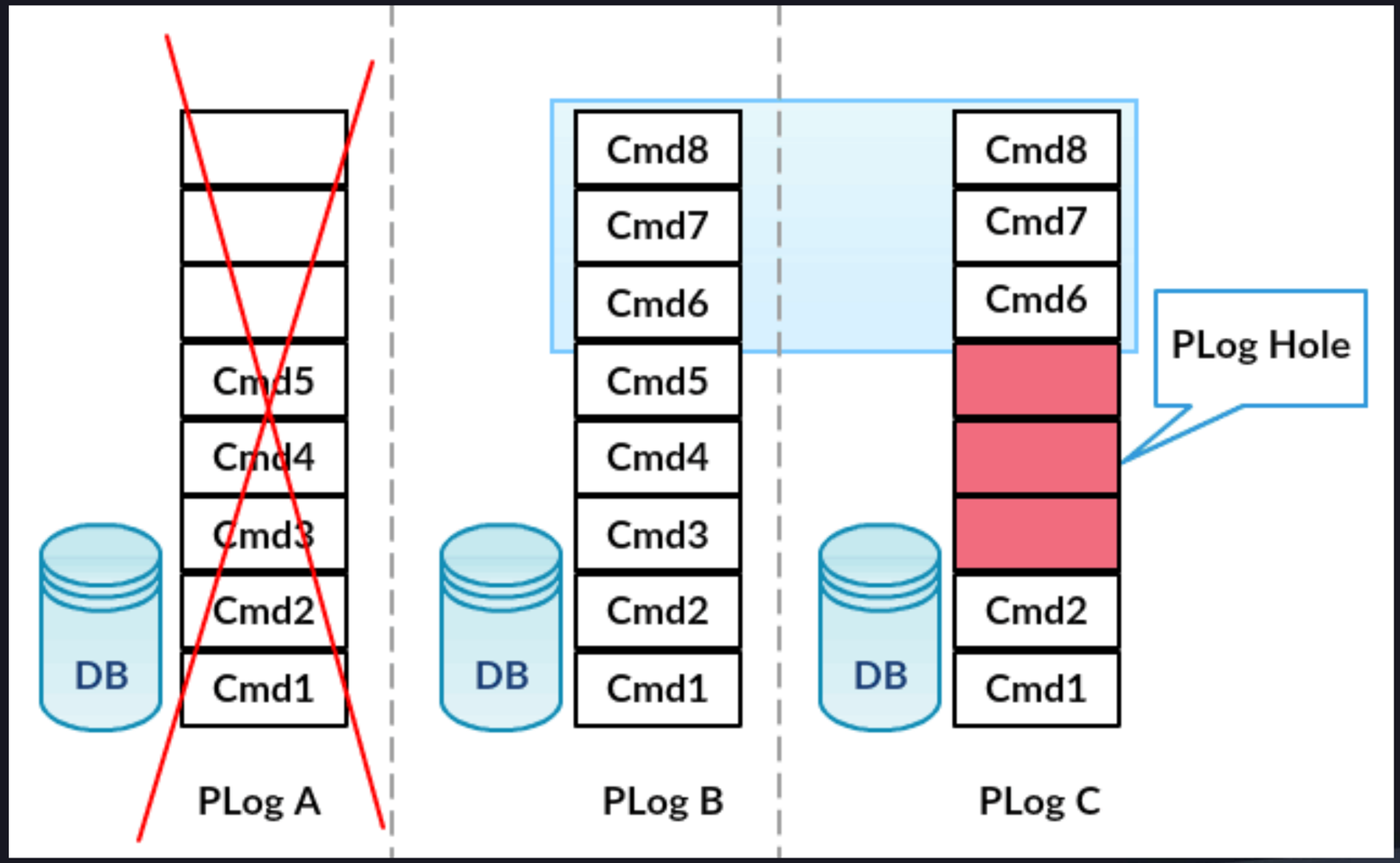


多主多写下的活锁问题

- 活锁问题如何解决？
  - Client路由先A后B
  - 冲突时进行避让
  - 相同PLog的请求合并

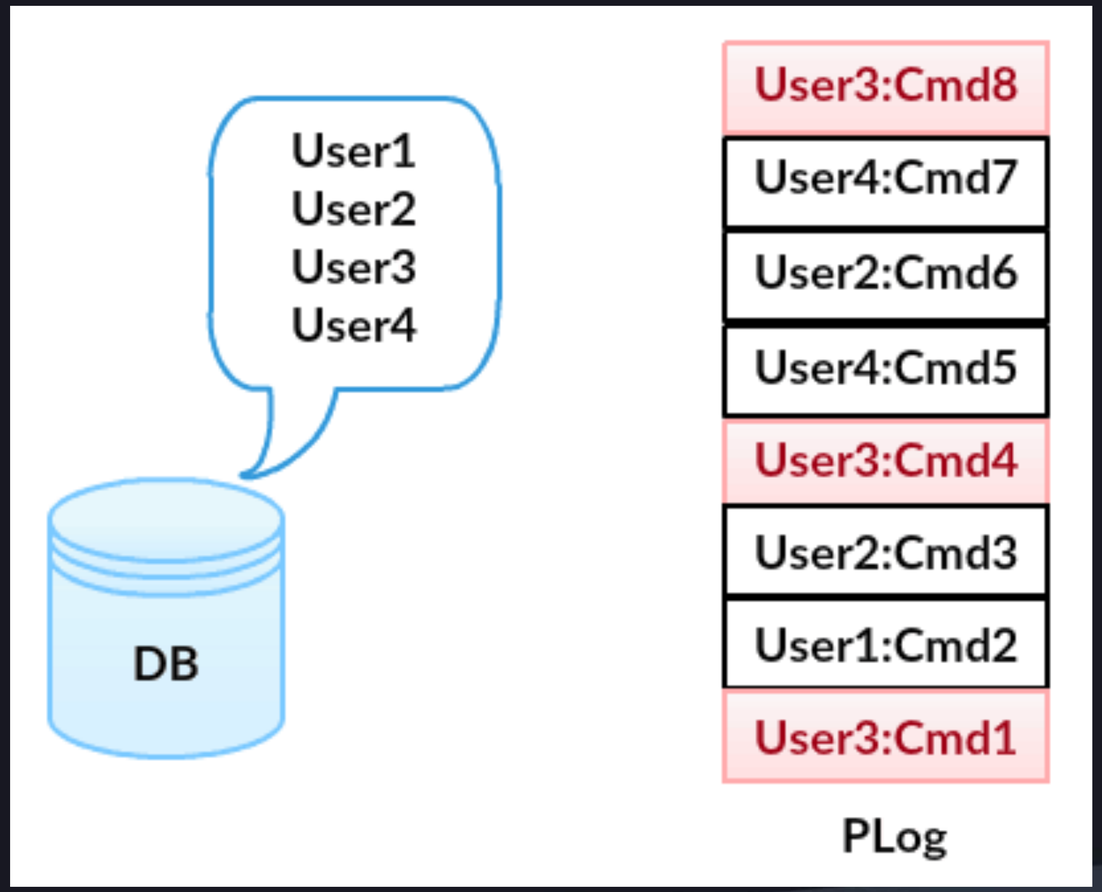


为什么PLog（变更日志）允许空洞的情况下可用性更高？



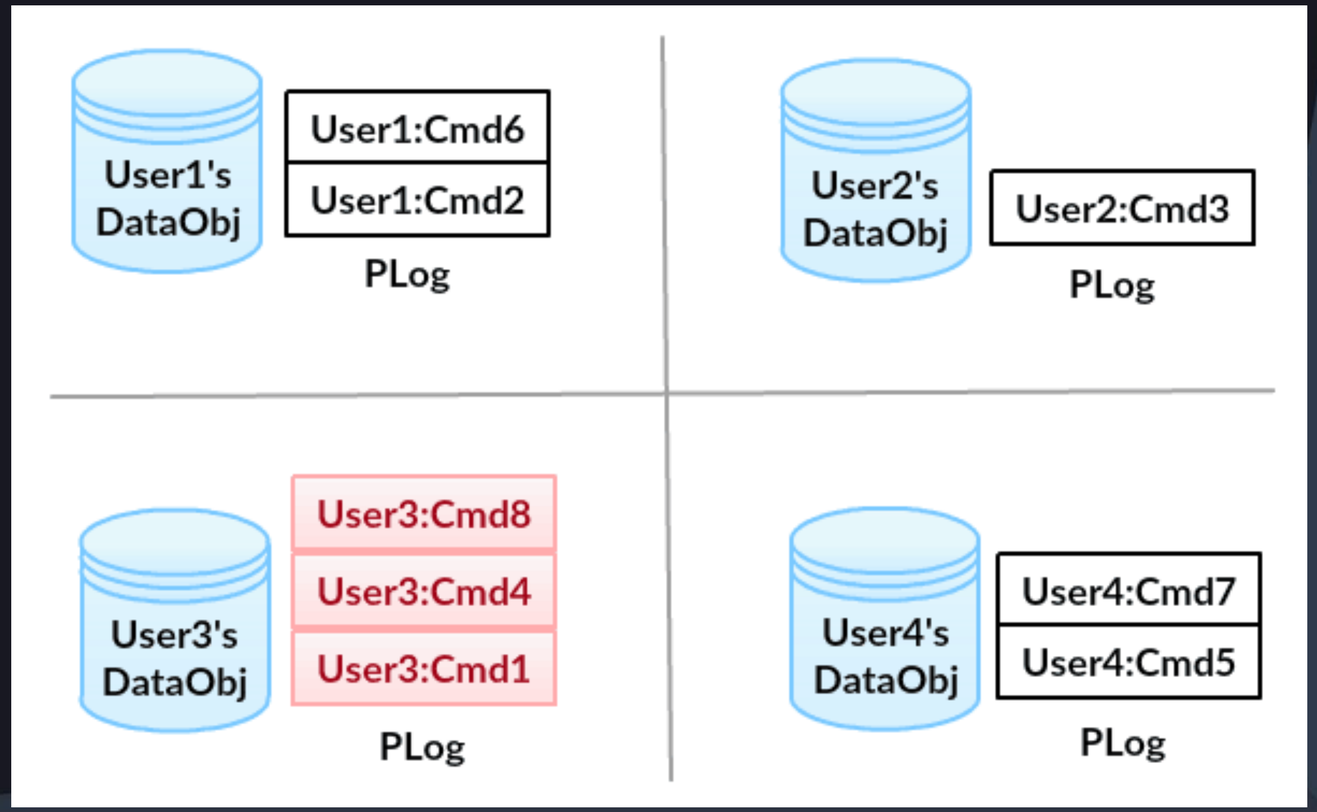
PLog（变更日志）与数据对象的对应关系

- 整个数据库对应一个PLog
  - 访问单个用户数据
    - 落后时Catchup所有用户变更
- 单个用户数据发生异常
  - 影响对应PLog的所有用户



PLog（变更日志）与数据对象的对应关系

- PLog细粒度化
  - 用户数据相互隔离
  - 数据落后、丢失时，独立恢复
- PLog细粒度下的恢复策略
  - 访问时优先恢复
  - 后台线程全量恢复

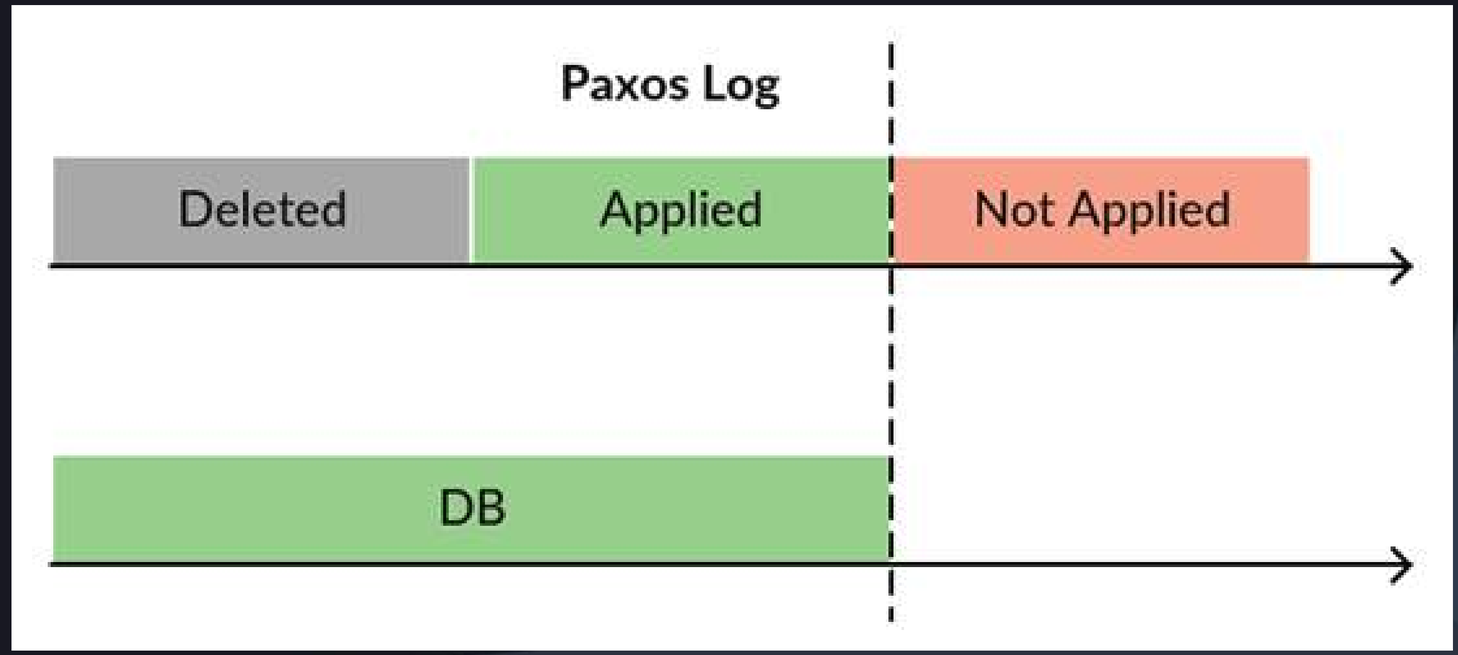


基于PLog（变更日志）的数据恢复

- 如何应对PLog无限增长的存储空间？

- PLog日志存在的意义
  - 协议交互需要
  - Commit到DB数据库

- 删除PLog日志
  - 安全性：Commit下可删
  - 过期删除



快消型PLog

- PLog存多久合适？
  - 存储成本 VS 数据全量恢复性能

- 增量型数据
  - 以严格递增序列号为Key的列表结构
  - 以时间戳为Key的集合结构

数据恢复方式总结

- 基于PLog日志
- 基于增量数据
- 基于全量数据

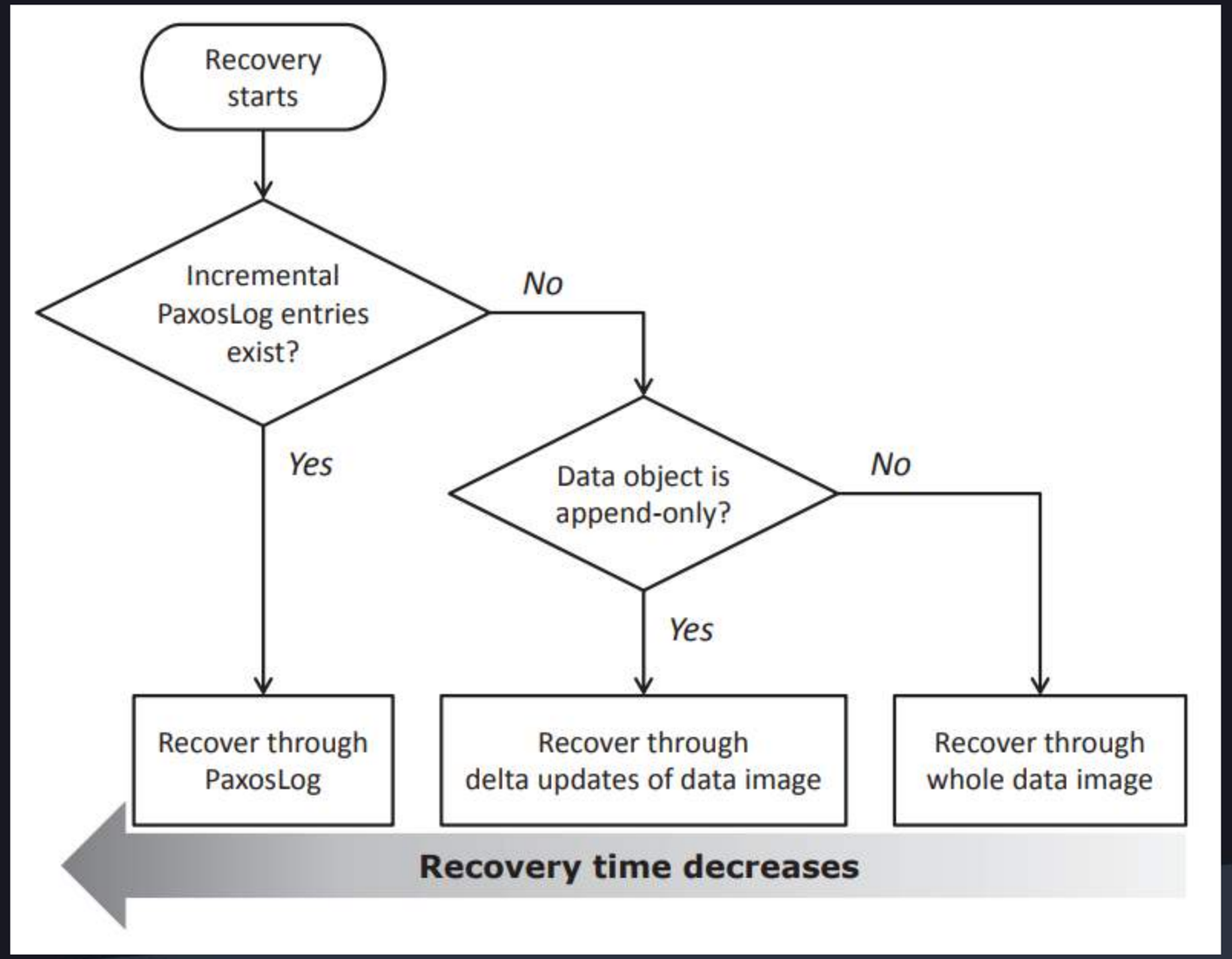
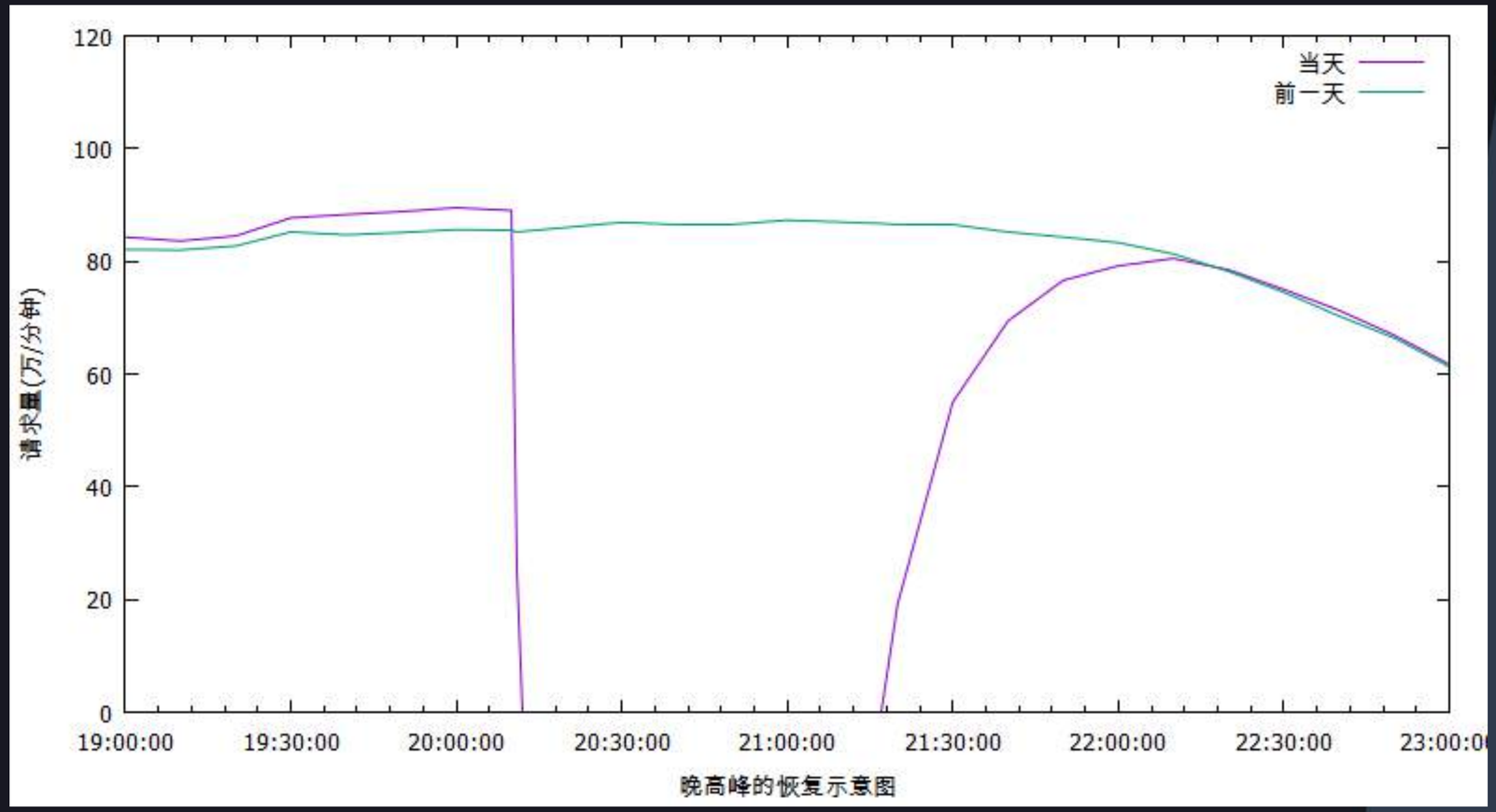


TABLE OF
CONTENTS 大纲

- 项目介绍
- PaxosStore架构设计
- 具体案例分析
- 项目的挑战和发展

朋友圈索引存储

- 背景
  - 单机TB级数据
  - 热数据效应明显
- 实施效果
  - 机器故障，平滑切换
  - 回归集群，快速恢复



消息存储系统

- 背景
  - 1000+台机器，PB级数据
  - 消息写量：千亿级/天
  - 存储转发机制
- 实施效果
  - NoRaid代替Raid10，收益明显
  - DB支持增量恢复，PLog日志快消

TABLE OF
CONTENTS 大纲

- 项目介绍
- PaxosStore架构设计
- 具体案例分析
- 项目的挑战和发展

项目的挑战和发展

- 灵活伸缩
  - 数据分布支持按范围划分
- 计算存储分离
  - 分布式文件系统
  - 分布式表格系统

THANK YOU

如有需求，欢迎至 [讲师交流会议室] 与我们的讲师进一步交流

ArchSummit
全球架构师峰会 2017