
运维世界大会·深圳站

# Codis 过去和未来

黄东旭 PingCAP

# 关于我

- PingCAP Cofounder & CTO
- MSRA / Netease / PingCAP
- 基础软件工程师 / 架构师 / 开源狂热分子
  - Codis
  - TiDB
  - TiKV
- Weibo: @Dongxu_Huang
- Email: huang@pingcap.com

运维世界 Ops World

# Codis

A Redis Cluster Solution

# 2014 年...

- 重度 Redis 用户
- 大多数业务的 cache 命中率要求 90% 以上，否则 DB 就不行了
- 数据量越来越大
- 独立的 Redis 实例越来越多，业务苦不堪言
- Redis 一故障，基本业务就不可用了
  - 恢复时间
  - 数据一致性
- 引入 Twemproxy

# Redis 为什么那么快

- 绝大部分请求是纯粹的内存操作(非常快速)
- 采用单线程,避免了不必要的上下文切换和竞争条件
- 非阻塞IO
- 内部实现采用epoll，采用了epoll+自己实现的简单的事件框架。epoll中的读、写、关闭、连接都转化成了事件，然后利用epoll的多路复用特性，绝不在io上浪费一点时间

# Twemproxy

- 最早/使用最广泛的解决方案
- Proxy based
- 静态的拓扑
- 运维基本靠手
- 最大痛点:无法平滑的扩/缩容
  - 甚至修改个配置都需要重启服务。。。
- 但是业务正在迅速的扩张

# 一次 Twemproxy 扩容引发事故后。。。

- 下定决心彻底解决的 Redis 的扩展性问题
- 3 个人
- 1 个月

# Twemproxy 的问题和优点

问题：

- 扩容缩容需要停服务
- 没有完善的监控和运维工具
- 不支持在线的数据迁移
- proxy 单线程，无法利用好 CPU，只能靠多进程

优点：

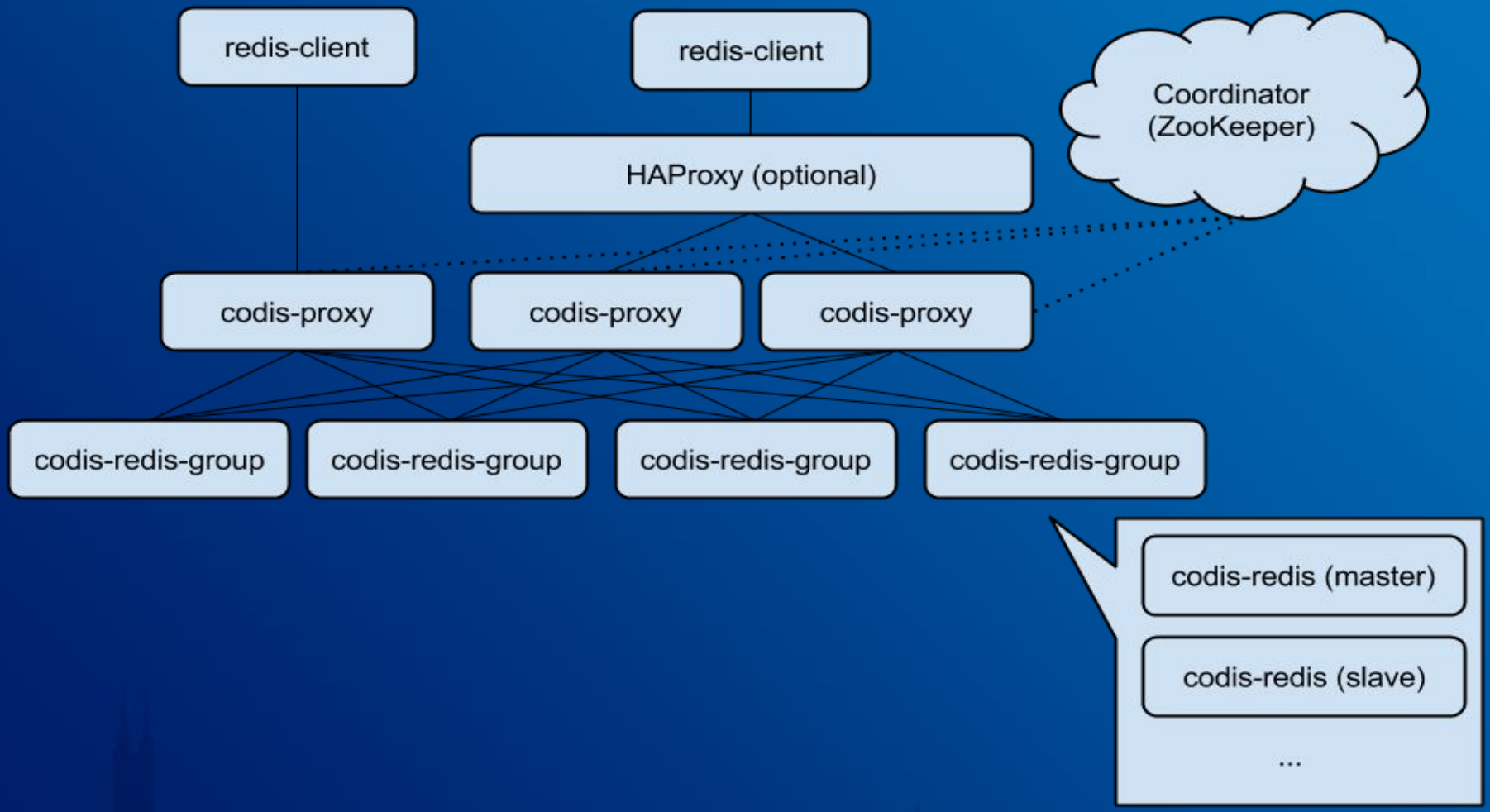
- state-less 的 proxy 模型，对业务侵入小，proxy 层能水平扩展
- 架构比较简单，升级 proxy 对 redis 的数据没影响

# Redis Cluster

- 官方出品
- 去中心化设计，类 raft 的算法
- 客户端需要修改 (smart client)
- 整个系统高度耦合，升级比较困难

# Codis

- 完全兼容 Twemproxy
- 3 个人就写了俩礼拜。。。就上线了



# Codis 整体设计

- Pre-sharding
  - Slot => [0, 1023]
- Zookeeper
- Proxy 无状态
- 平滑扩容/缩容
- 扩容对用户透明

# 设计考量 1/3

- 分布式系统是复杂的
- 开发人员不足
- 尽量拆分，简化每个模块，同时易于升级
- 每个组件只负责自己的事情
- Redis 只作为存储引擎
- Proxy 状态

# 设计考量 2/3

- Redis 是否挂掉的判定放到外部，因为分布式系统存活的判定是复杂的
- 提供 API 让外部调用，当 Redis master 挂掉的时候，提升 slave 为 master
- 我们不喜欢读写分离
- 基础软件开发者的自我修养。。。。

# 设计考量 3/3

- graph everything
  - slot status
  - proxy status
  - group status
  - lock
  - action

# proxy vs smart client

**proxy**:

更好的监控，控制

后端信息不暴露，易于升级

**smart client**:

更好的性能

更低的延迟，升级比较麻烦

# 无状态 Proxy

1. 路由表统一存储在 Coordinator 中
2. 连接任意一个 proxy 发起请求的效果是一样的
3. 负载均衡变得非常简单，proxy 可以平滑的水平扩展

# 路由信息一致性保证

在 cluster-admin 发起集群状态变更时，所有的 proxy 必须达到信息的一致后，才能重新对外提供服务

- cluster-admin 和 proxy 之间的二阶段提交

Codis 的设计背景是要求强一致的

- 宁可放弃部分可用性

# Codis 2.0+

- Codis-HA
  - 为什么不做到自动？
- Performance
  - Pipeline
- 读写分离
- Stronger dashboard

# Result

1. 平滑水平扩展
2. 可插拔的存储引擎（各个模块之间的纽带是 Redis Protocol）
3. 由于不同组件之间解耦, 都可以独立升级
    a. proxy
    b. cluster admin
    c. storage engine

运维世界 Ops World

# 我对性能的一些看法

- 延迟？吞吐？
- Benchmark 的陷阱
- 需求和理想
- 性能是一切吗？

# 基础软件的未来

- 随着单个计算设备物理成本的降低，集群化是必然的趋势
  - Scale-up vs Scale-out
- 最后必然的结果：Cloud-native
- Infrastructure 真的会变成水电一样自然的东西

运维世界 Ops World

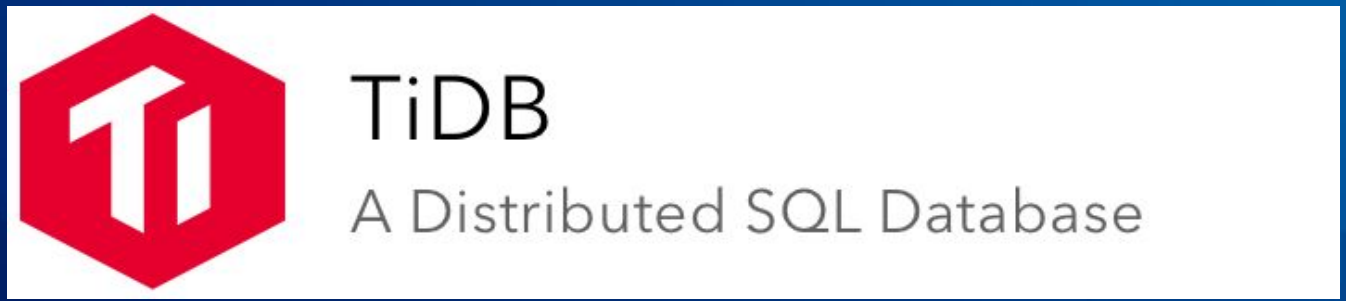
# 基础软件的未来

- Application
  - Container
  - Scheduler
- Cahce
- DB
  - **OLTP**
  - OLAP
- DFS / Distributed Object Storeage

**GIFEE (Google  Infrastructure For Everyone Else)**

# 最后做个小广告 :)

- Google Spanner / F1 的开源实现
  - A distributed SQL database that is widely used in Google
  - The only OLTP database that works at scale
- 高度兼容 MySQL
  - 事务
  - 复杂查询
  - 主从复制
  - 运维工具



运维世界 Ops World

# Thanks

PingCAP - OpsWorld 2016
深圳