

# 在线业务快速扩容

用户标签服务



主讲人：黄俊炜



# CATALOG 目录

01 背景介绍

02 方案分享

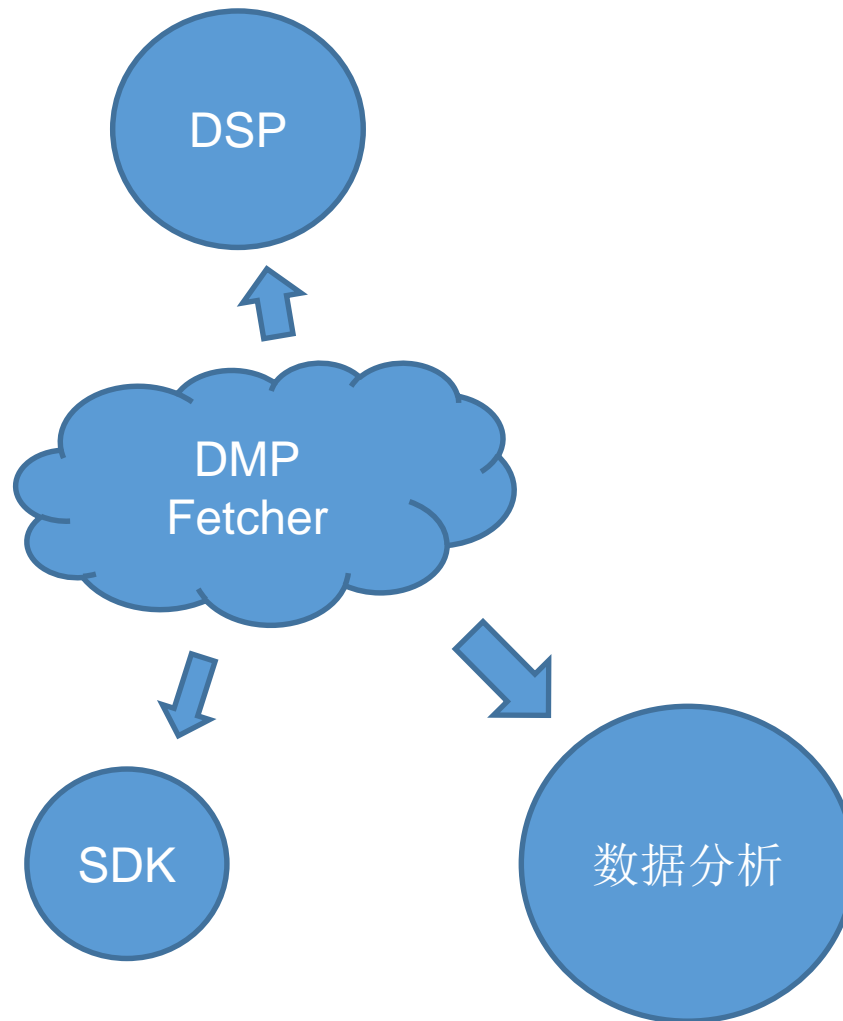
03 技术实践

04 经验心得

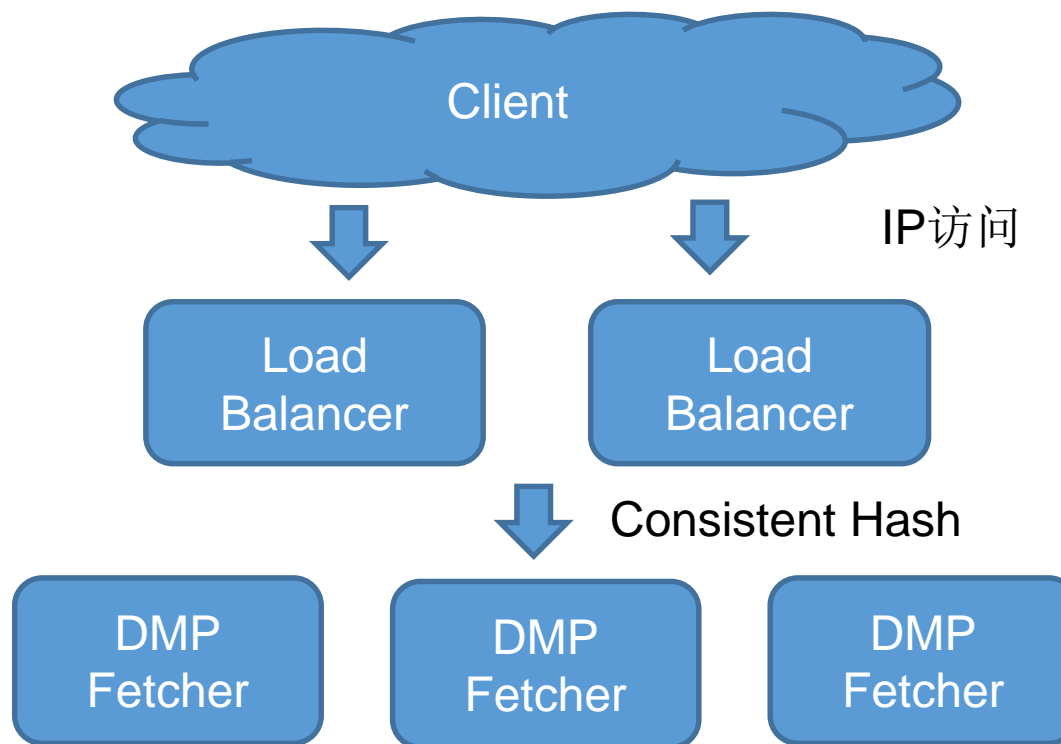
01

# 背景介绍

- 简称DMP Fetcher
- 查询用户的标签
- 只读服务
- 主要的服务对象
  - DSP广告
  - SDK广告
  - 数据分析
- 目标
  - 低延迟
  - 方便扩展
  - 节省成本



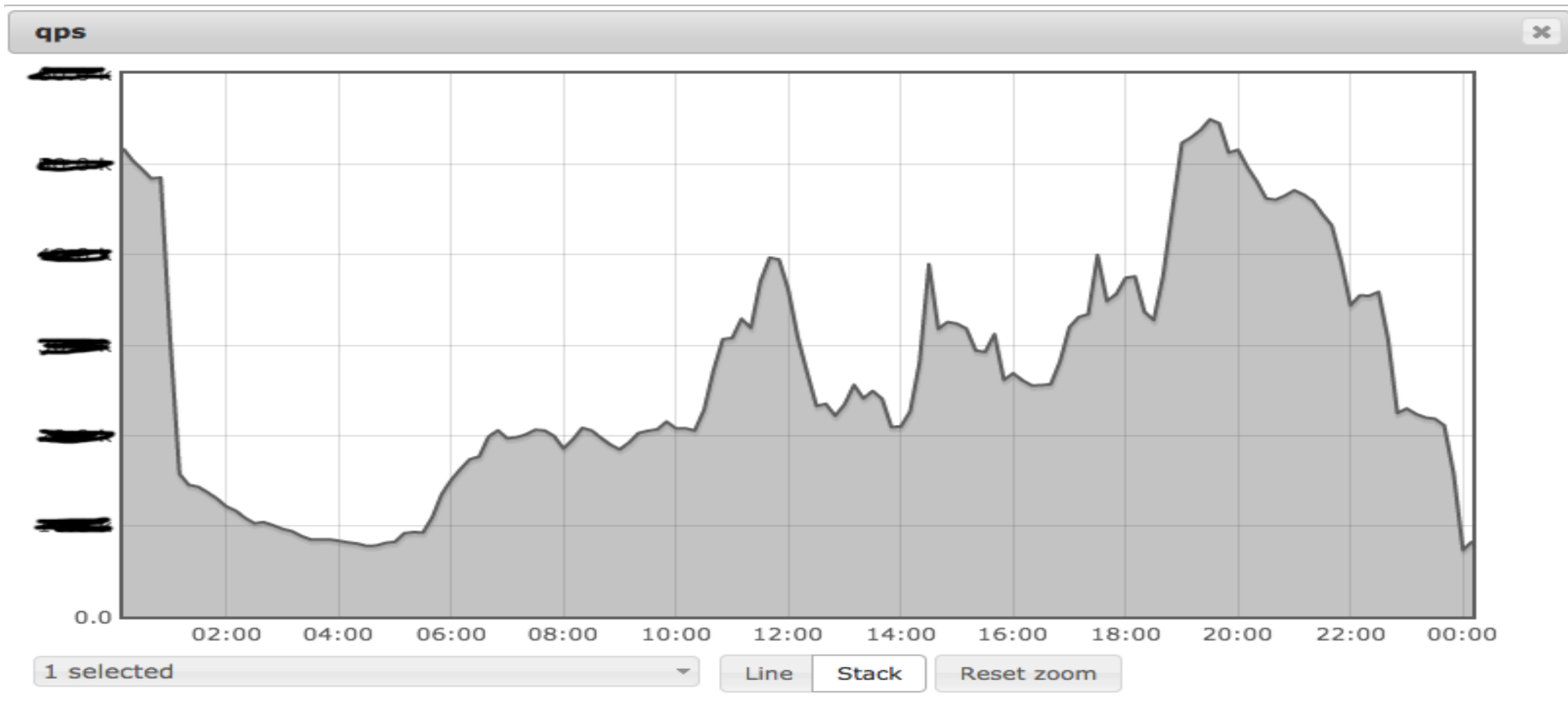
- Golang开发
- 初始设计
- TCP Binary协议
- HAProxy做负载均衡
- Consistent Hash
- 扩展策略
- 水平扩展



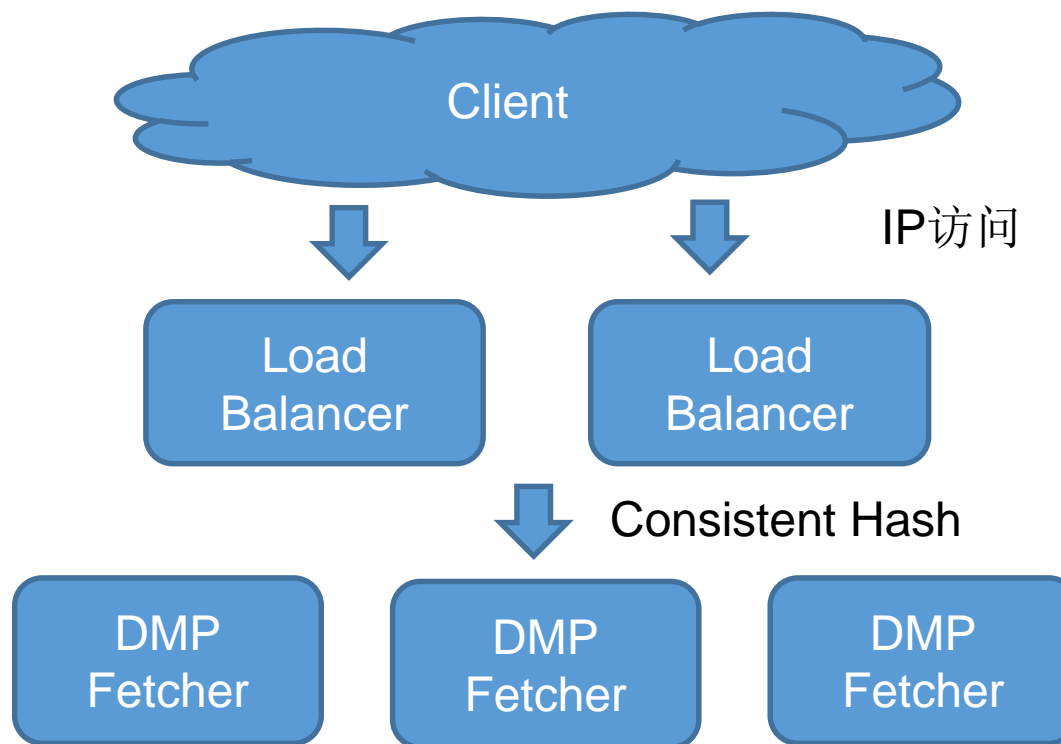


# 背景介绍

■ 流量趋势



- 负载均衡器
  - 常成为性能瓶颈
  - 扩展困难
- 扩展Fetcher
  - 负载均衡器Reload缓慢
- Ansible
  - 手工维护动态节点
- 网络链路长，传输消耗大

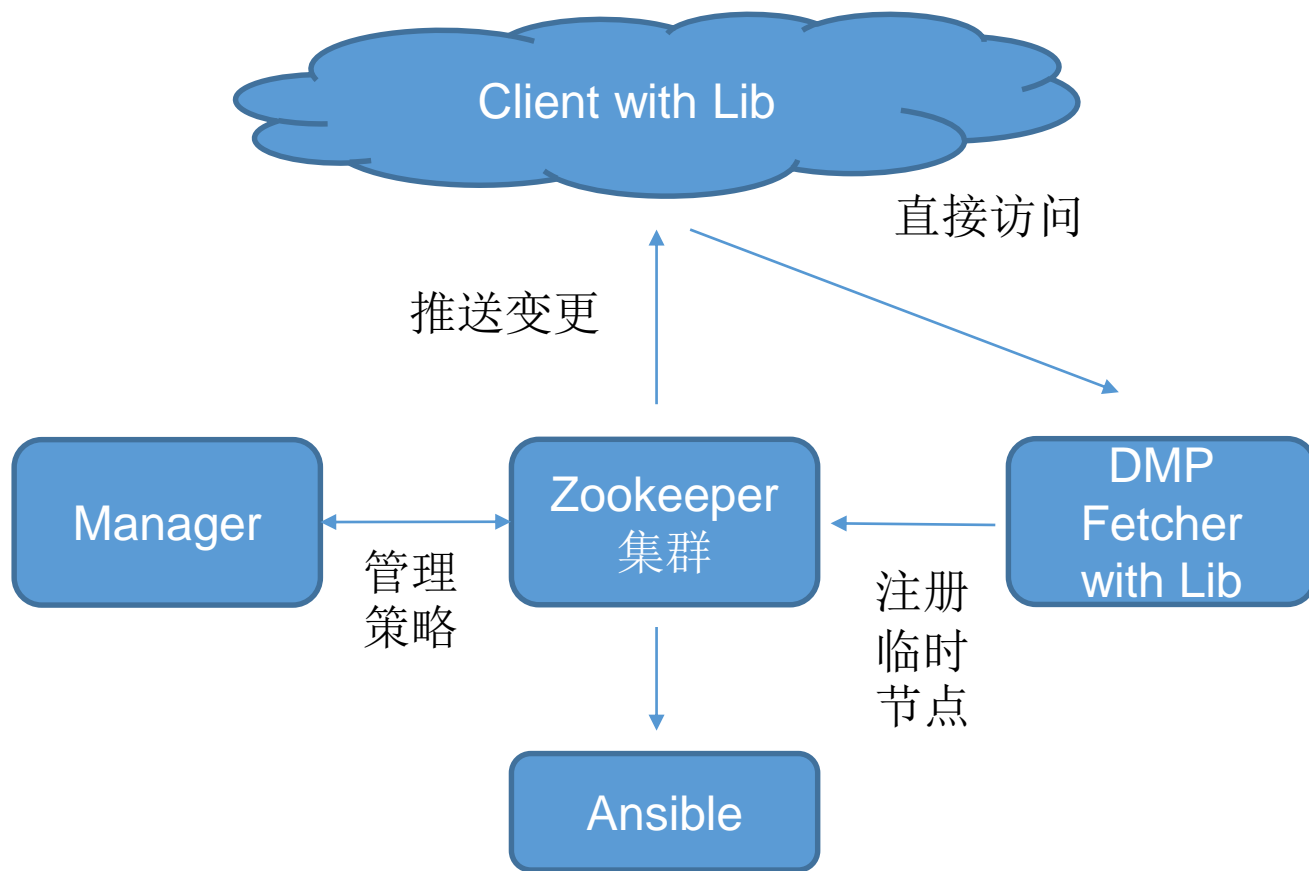


02

## 快速扩容方案



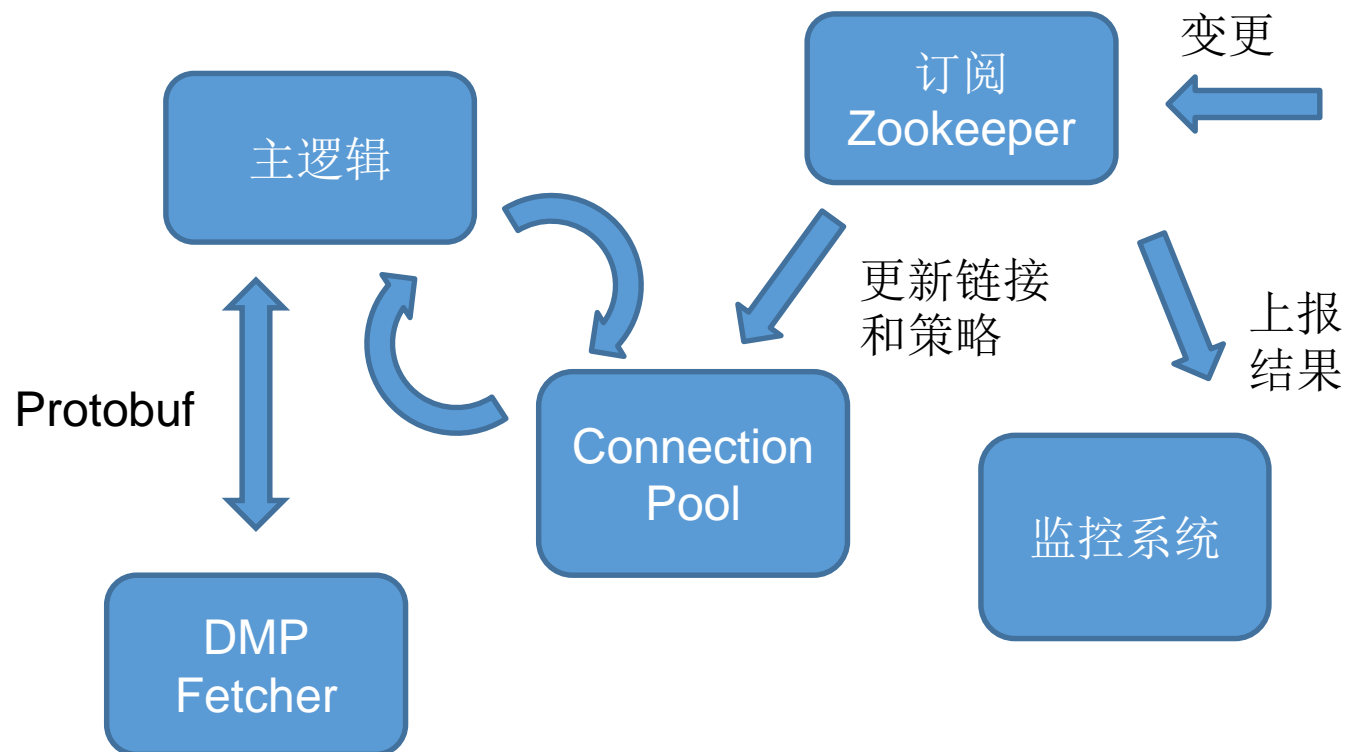
- 服务发现
  - Zookeeper集群
    - 请求分发策略
- 废弃负载均衡器
- Manager
  - 管理策略
  - 定制扩容条件
- 提供Lib
  - Fetcher启动时注册zookeeper临时节点
  - Client订阅zookeeper推送的变更
- Ansible
  - Dynamic Inventory



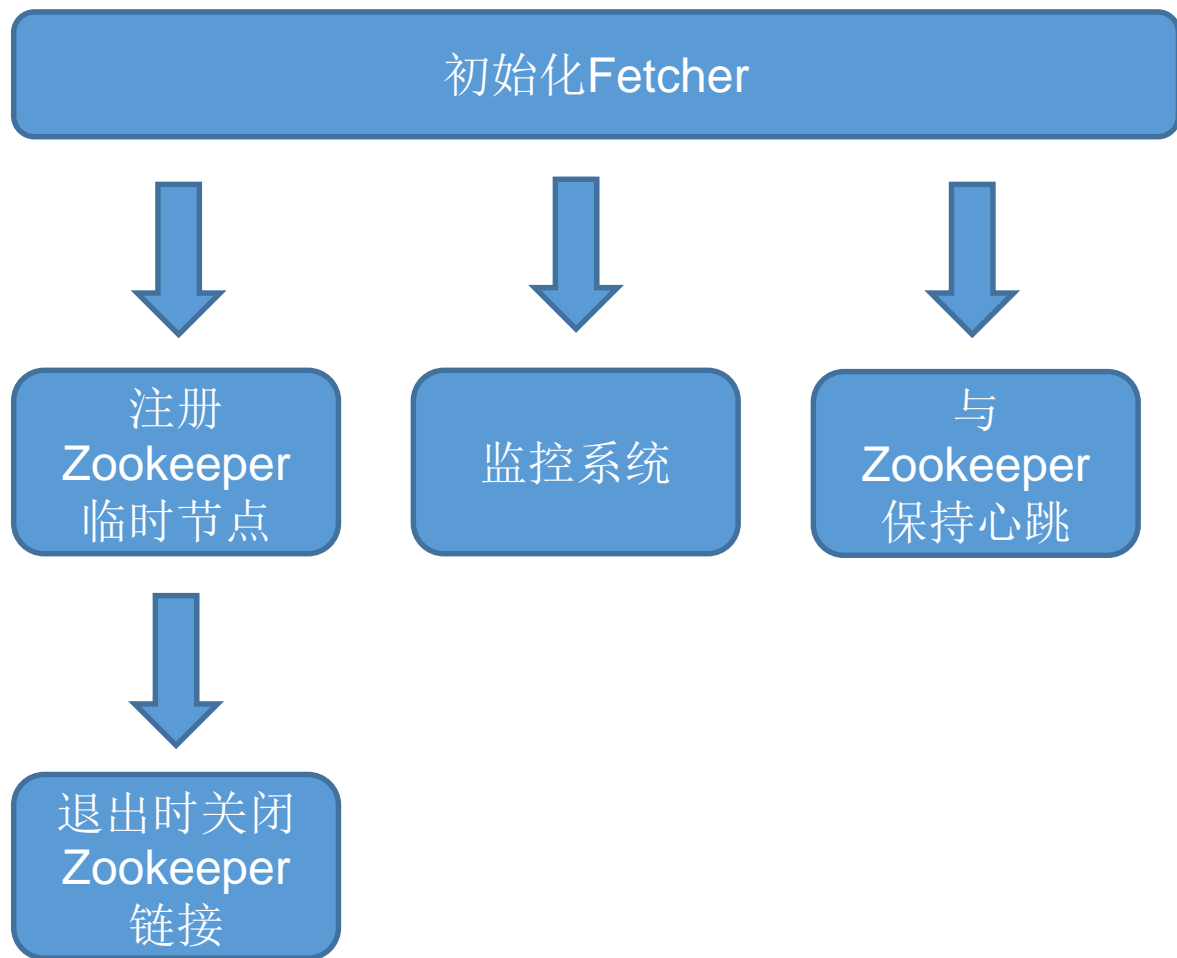
03

技术实践

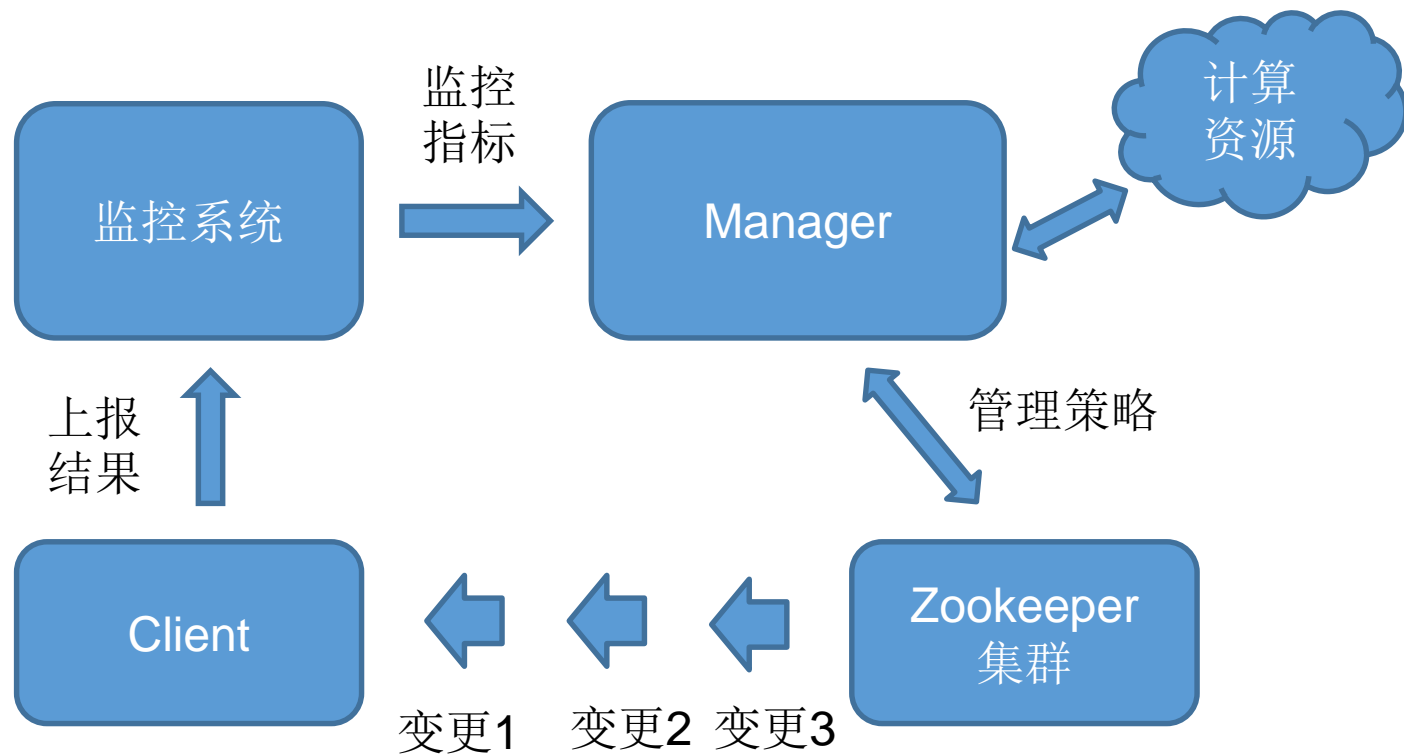
- 主逻辑
  - TCP Binary
  - Protobuf
  - 失败重试
- Connection Pool
  - Go channel
  - 根据策略返回链接
- 订阅Zookeeper
  - 维护链接池正确性



- 初始化
  - 协议框架
  - 建立数据库链接
- 临时节点
  - 注册临时节点
  - 退出时删除
  - 登记访问方式和处理能力
- 上报QPS、活跃worker数
- 与Zookeeper保持心跳



- Manager掌控扩容的过程
  - 管理计算资源
  - 管理请求分发策略
- Manager计算策略
  - 读监控指标
    - QPS
    - 平均响应时间
  - 读Zookeeper
    - Fetcher处理能力
- Zookeeper按顺序推送变更到Client

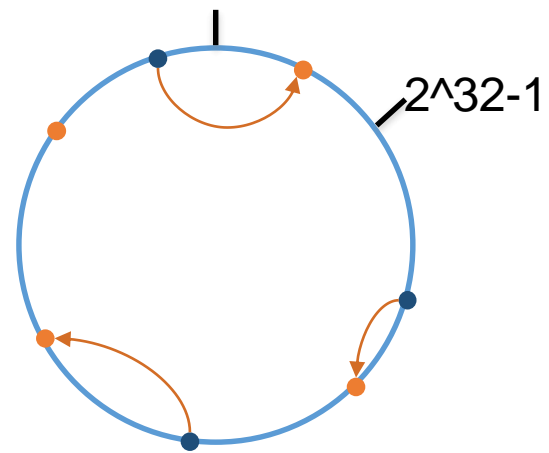
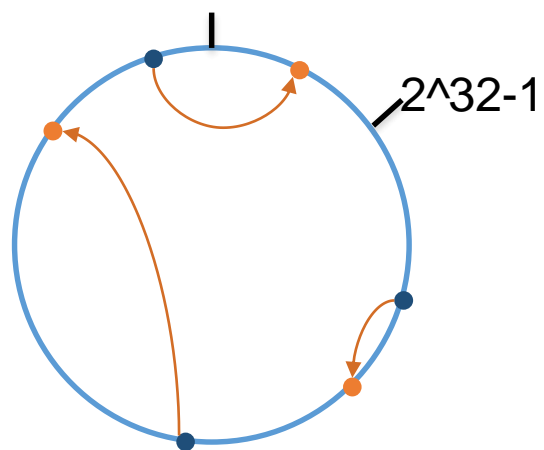




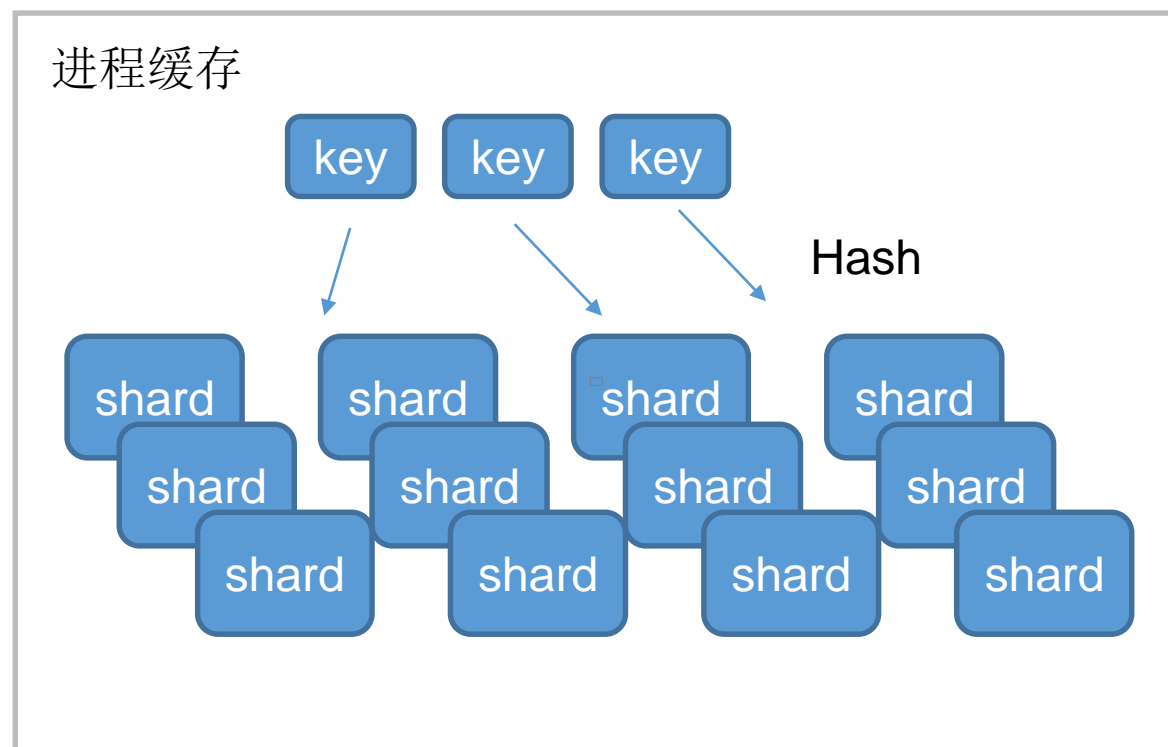
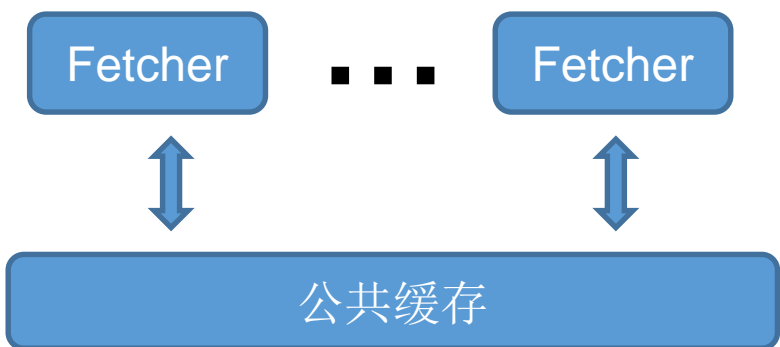
04

经验心得

- 单调性
  - 一个Key的被分配到固定的节点
  - 充分利用进程内缓存
- 平衡性
  - 节点增减，受影响的数据少
  - 减少进程缓存失效的影响



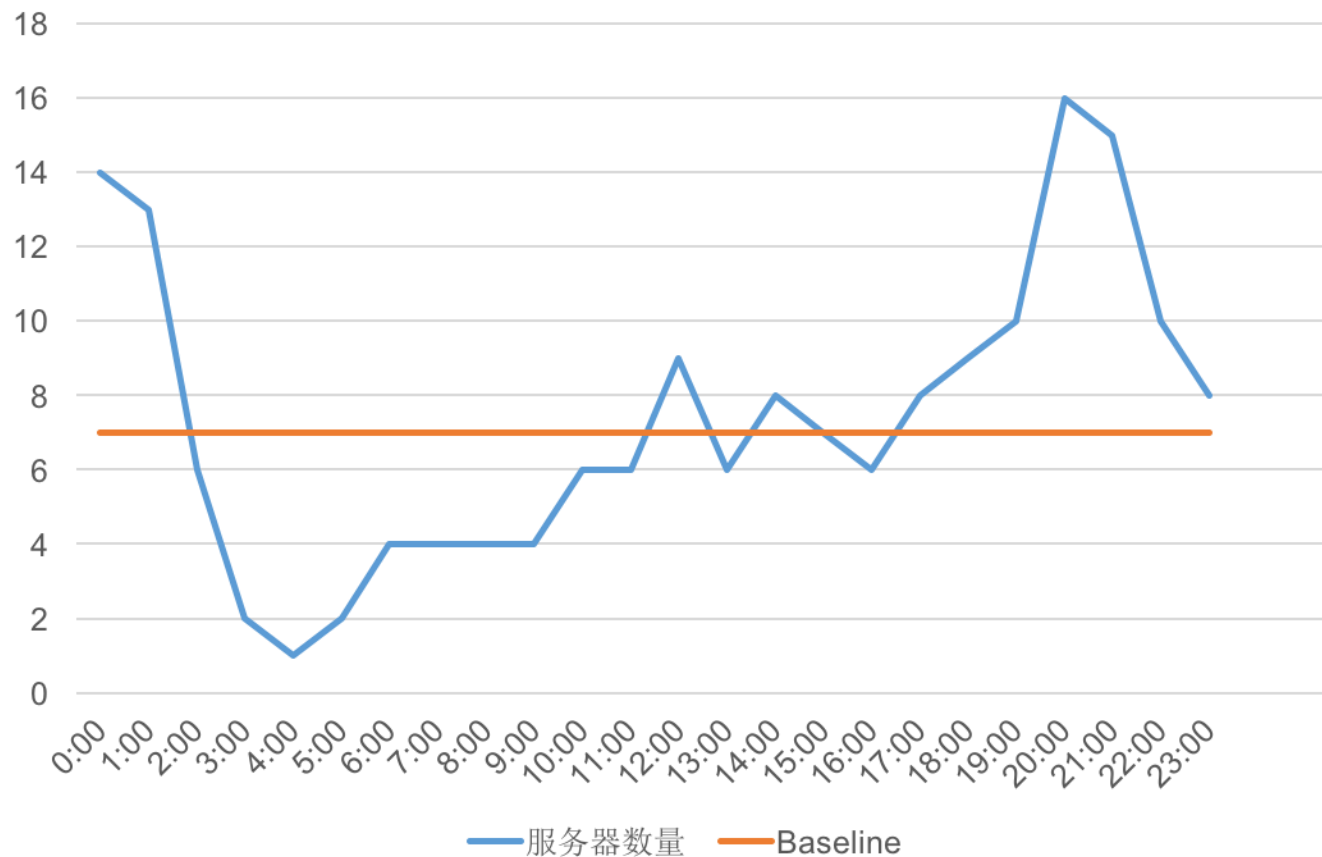
- Fetcher缓存
  - 进程内缓存
    - 分片
    - Array
  - 公共缓存
    - 避免缓存穿透



- 早扩容
  - 确保在系统负荷到达处理能力极限前扩容
  - 避免雪崩
  - 为服务器、Fetcher启动预留时间
  - 降低启动失败的损失
- 迟缩容
  - 避免错误判断短暂波谷而过快缩容
  - 防止缩减过多计算能力
  - 避免过快缩容后再次扩容导致变动过于频繁
  - 定期进行Benchmark，调整扩缩策略

- Health Check改进
  - 启动新服务器或Fetcher实例，执行Health Check脚本，保证查询速度
  - 进行适当的Warm up
- Shutdown改进
  - 针对服务器以及Fetcher实例的特点，定制 Graceful Shutdown脚本
- 综合历史扩缩记录
  - 购买适当的预留实例

服务器数量趋势图





- 内网DNS配置TXT记录，标记Zookeeper集群的地址。减少Client的配置。

```
;; ANSWER SECTION:
zookeepers.y.cn.      60      IN      TXT     "127.0.0.1:2181"
zookeepers.y.cn.      60      IN      TXT     "127.0.0.1:2182"
zookeepers.y.cn.      60      IN      TXT     "127.0.0.1:2183"
```

- Zookeeper集群的tickTime适当降低，让Client能尽快获知Fetcher实例异常



Q&A



有米科技 | 全球领先的综合性移动互联网企业



# THANKS



有米科技 | 全球领先的综合性移动互联网企业