WOTA

51CTO

World Of Tech 2017

全球架构与运维技术峰会

2017年4月14日-15日　北京富力万丽酒店

ARCHITECTURE

WOTA

51CTO

出品人及主持人：

赖春波 滴滴出行平台技术部 高级技术总监

高可用架构

WOTA

51CTO

高性能视频播放调度系统

•邓铮 一下科技高级研发总监

WOTA

51CTO

邓铮

一下科技

高级研发总监

分享主题：

海量播放请求下的精准视频播放调度系统

公司介绍

- 一下科技 国内领先的移动视频服务商
- 旗下产品
  - 秒拍 (2013.9)
  - 小咖秀 (2015.5)
  - 一直播 (2016.5)
- 每日短视频播放次数超过20亿次

短视频 vs 长视频

视频时长较短, 一般为几分钟	视频时长一般为20分钟以上乃至数小时
视频来源广泛,以UGC内容为主,比较鲜活	视频来源以版权内容为主
视频更新量很大,每日数十万条	视频更新很少,每日数十条乃至数百不等
视频平均播放量较小,数次至数十次不等	视频平均播放量在数千到数万级别

短视频播放面临的挑战

- 时长短
  - 首播延迟时间敏感
- 来源广
  - 上传来源地区广泛, 需要快速分发
- 更新量大 平均播放次数少
  - 内容普遍比较冷,快速启动很重要

如何解决

- 上传
  - 就近寻源(北京/宁波/广州/天津)
  - 传输压缩
- 播放
  - CDN分发
  - 多级节点预推送

CDN如何选择

- CDN厂商很多, 节点也很多
- 如何判断该采用哪一家的哪个节点呢?

调度系统

- 将用户的请求转发给合适的后端服务的系统
- 视频调度
  - 输入:用户的IP和请求内容
  - 逻辑处理:对可分发的CDN节点打分排序
  - 输出:转发请求到对应节点

调度系统的特点

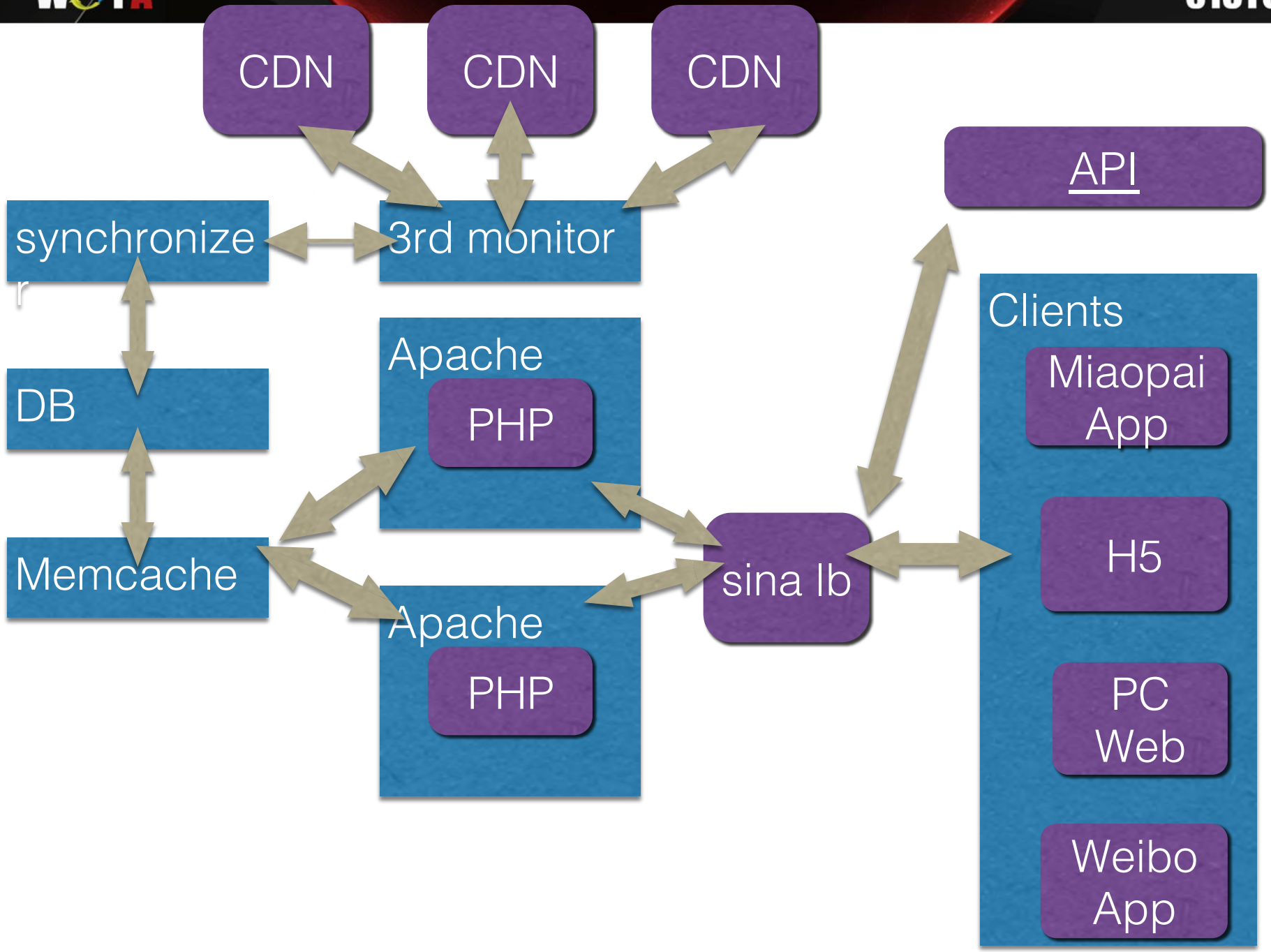
负载均衡 故障隔离 健康检查 日志记录 权限分配	高可用 高吞吐量 高性能

秒拍调度系统的发展

- GSLB v1
  - 基于第三方评分的地域调度系统
- GSLB v2
  - 基于C段IP的精细调度系统

GSLB v1

- 基于第三方监测结果
- 统计来源请求的地域与运营商
- 对相关结果进行打分
- 调度到相应节点


CDN
CDN
CDN
API
synchronize
r
3rd monitor
Clients
Apache
PHP
DB
Miaopai
App
H5
Memcache
sina lb
Apache
PHP
PC
Web
Weibo
App

评分原理

- 监测目标
  - 全国各省市不同运营商的节点
- 监测方式
  - 第三方监测机构定时去测试播放
- 评分体系
  - 针对城市+运营商级别做排序
- 判定原理
  - 用户IP->城市及运营商->根据评分选定节点

系统优点

- 整体结构简单
- 易维护
- 水平扩展性强
- 性能可以满足要求

系统缺点

- 测试点数有限 千量级
- 测试间隔较长 不能反映及时情况
- 系统在高并发有瓶颈
  - IP反查较慢
  - Apache+PHP单次请求时间长
  - 受限实体环境难于及时扩展

GSLB v2

- 核心思想
  - 精细化调度
    - 调度粒度细化
    - 积累测试数据
    - 近实时反馈
  - 提高吞吐量
    - 云端迁移
    - 引入OpenResty
    - IP快速定位

质量评测

- 建立基于客户端的反馈机制
- CDN厂商+节点质量报告
- 首播时间
- 卡顿率
- 播放成功率
- 播放完成比例
- …

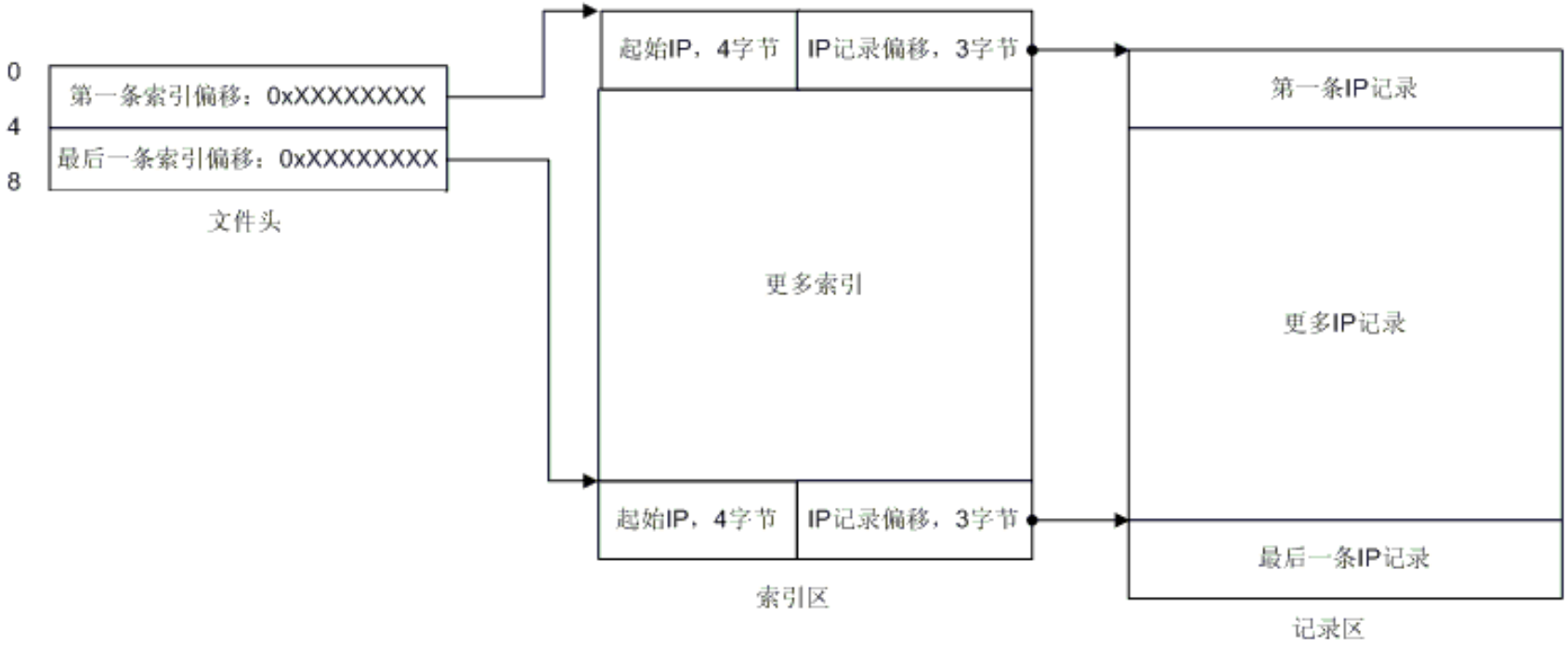
调度的精细化

- 传统基于IP调度的问题
  - 取决于IP库的精准度
  - 经常有IP判断不准的问题
  - 小运营商出口问题

IP库现状

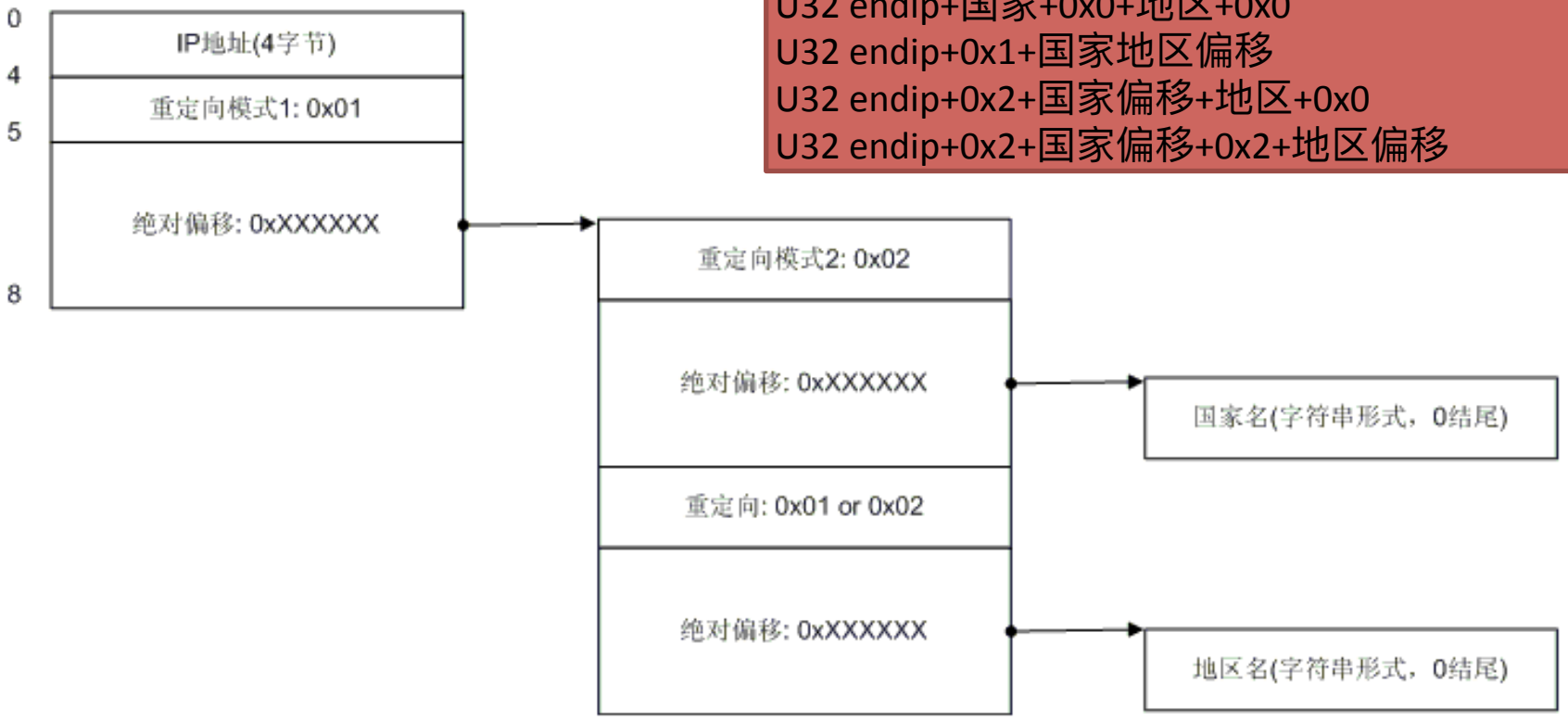
- 官方数据IANA(Internet Assigned Numbers Authority)
- 渠道收集
- 网友上报
- 运营商数据
- 非结构化(黄浦区淮海路与北京路交界益网通网吧)

纯真IP格式


0
4
8
第一条索引偏移：0xXXXXXXXX
最后一条索引偏移：0xXXXXXXXX
文件头
起始IP，4字节
IP记录偏移，3字节
更多索引
起始IP，4字节
IP记录偏移，3字节
索引区
第一条IP记录
更多IP记录
最后一条IP记录
记录区

纯真IP格式(续)

U32 endip + 文字描述
U32 endip+国家+0x0+地区+0x0
U32 endip+0x1+国家地区偏移
U32 endip+0x2+国家偏移+地区+0x0
U32 endip+0x2+国家偏移+0x2+地区偏移



纯真的特点

- 核心算法
  - 索引 + 二分查找
- 优点
  - 占用内存小
  - 文件体积小
- 缺点
  - 数据会越来越增长臃肿化
  - 非结构化数据

解决方向

- 结构化存储
- 索引大小固定非增长
- 减小查找时间
- IP最好直接对应数据(城市/运营商)

核心算法

- 一个C段只有256个ip，A.B.C.0~A.B.C.255
- 一般一个C段ip的地理位置，运营商信息都会一致
- 描述C段的所有IP，只有256*256*256=16777216个
- 如果一个ip对应信息是一个字节，需要储存空间16M，对应信息是两个字节，需要储存空间32M，每个C段ip对应一个编码（IPC码）
- 查询只需要根据偏移直接定位(A*256*256+B*256+C)*2
- 信息的前半段描述地区，后半段描述运营商

如何高效表示信息

- XXXX XXXX XXXX XXXX
- X 国内/国外，国内0，国外1，国外精度到国家
- XX 大区，4大区，华北，华中，华南，西部
- X XX 省，区内8省
- XX 省内区域，如粤东，粤西，粤北，珠三角
- XXX 区内8市
- X X 市内4县区
- XXX isp区分

校验方式

- Ipc & 0xF000 是否国外ip
- Ipc & 0xFC00 得出ip省份
- Ipc & 0xFFE0 得出ip城市
- Ipc & 0x7 得出运营商
- Ipc - Ipc2 判断两ip的距离

数据积累

- 数据缺失时主动探测
- 探测原则
  - 同区域同ISP优先
  - CDN厂商节点分散化探测
- 数据已有则进行更新得分

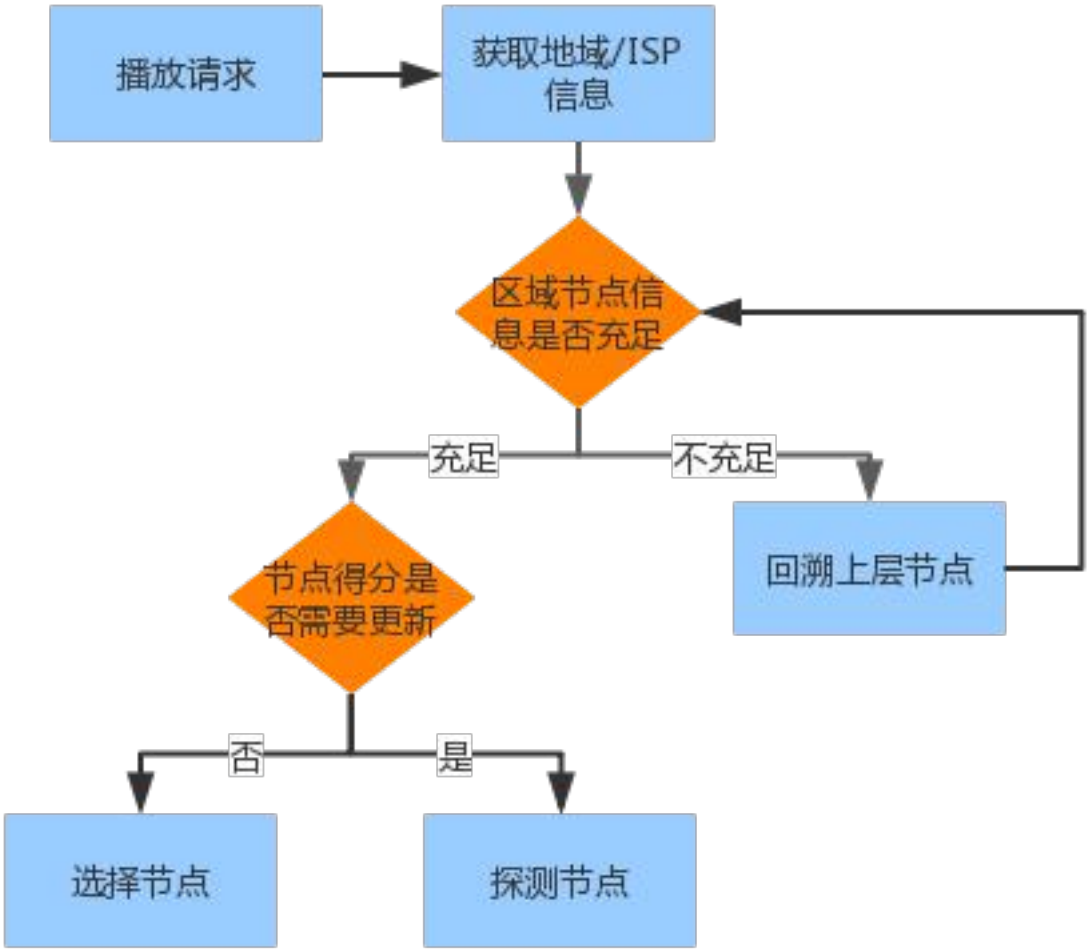
评分原则

- 基础得分:首播时间
- 播放卡顿罚分
- 播放失败罚分
- 得分计入时间
- 随时间衰减更新最终得分

节点选择

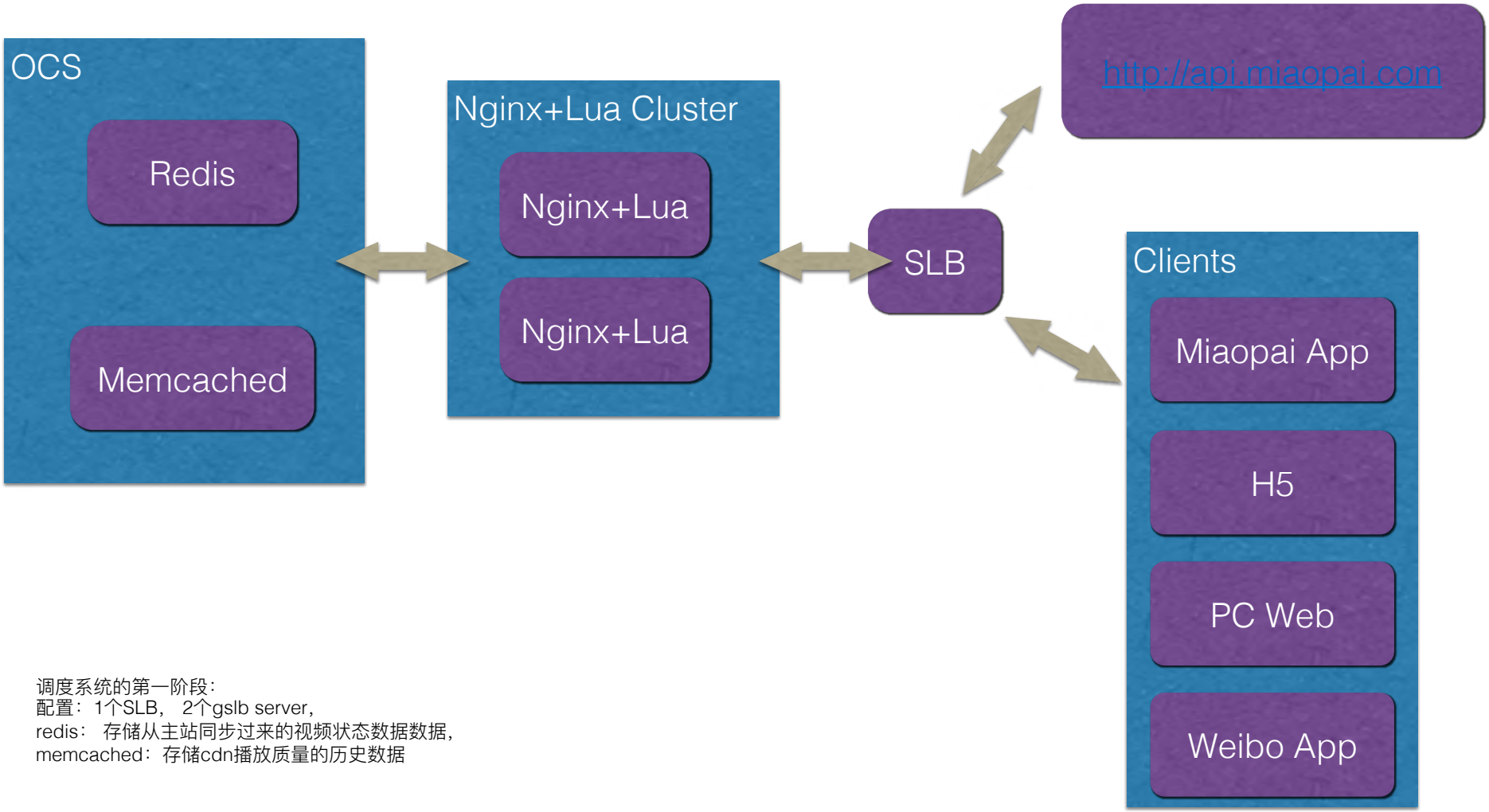
- 确定比较阈值
- 基于IPC码获取同区域内不同节点得分
- 如区域内节点数据满足阈值要求
  - 如节点得分需更新
    - 探测需更新节点
  - 否则选取节点
- 否则向上回溯节点

节点选择流程

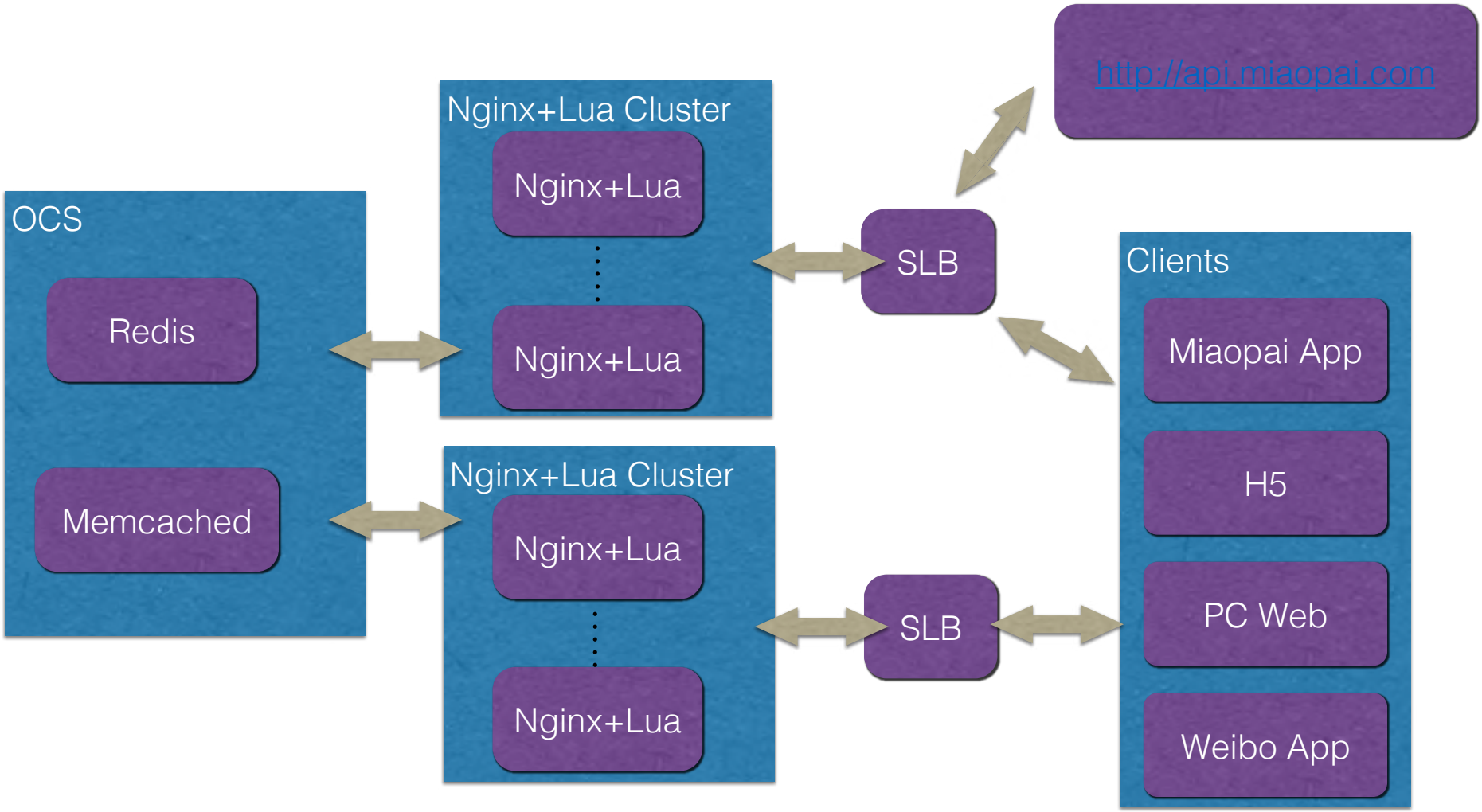

播放请求
获取地域/ISP信息
区域节点信息是否充足
充足
不充足
回溯上层节点
节点得分是否需要更新
否
是
选择节点
探测节点

系统吞吐量优化

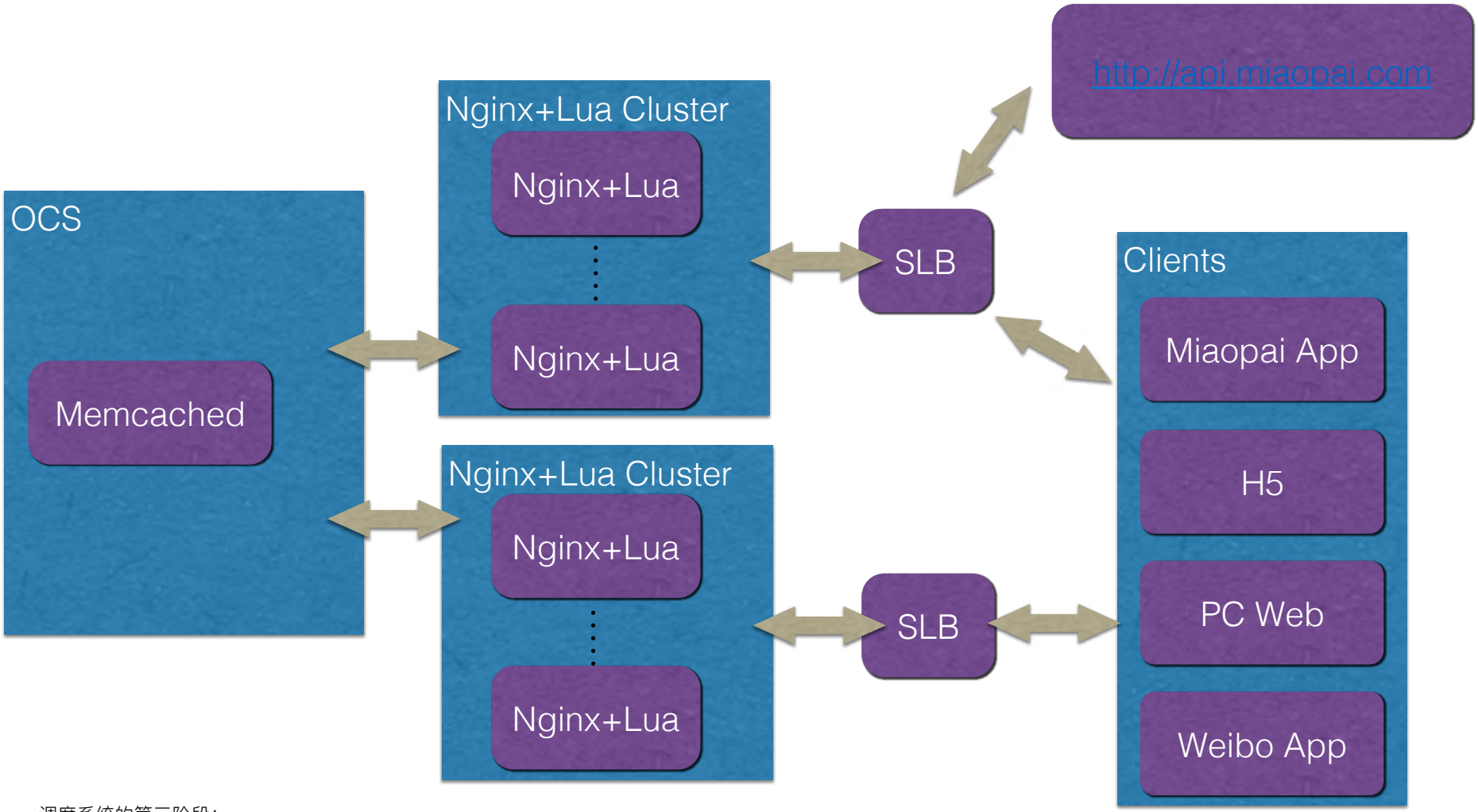
- 数据源
  - Memcache
  - Redis
- 纯异步通信的选择Lua
- OpenResty开发方便



调度系统的第一阶段：
配置：1个SLB， 2个gslb server，
redis： 存储从主站同步过来的视频状态数据数据，
memcached：存储cdn播放质量的历史数据



调度系统的第二阶段：
播放量成倍增长，对server进行了横向扩展
配置：多个SLB， 多个gslb server
redis： 存储从主站同步过来的视频状态数据数据，
memcached：存储cdn播放质量的历史数据



调度系统的第三阶段：
由于每个请求都需要对redis进行get操作获取channel的状态数据，redis性能出现瓶颈，于是替换掉了redis，把redis的存储变为memcached
配置：多个SLB， 多个gslb server，
memcached：存储cdn播放质量的历史数据，存储从主站同步过来的视频状态数据数据

由于openresty不支持mc的sasl验证协议，所以没有对mc进行横向扩展

后续展望

- 异地多活部署
- 混合云改造
- P2P调度融入
- 自建CDN节点
- 灾备建设
- 监控统计完善


WOTA
51CTO
Thank you !