

高并发实时弹幕的实战之路

一种**互动**的体验

直播弹幕的挑战

- ◆ 高稳定性

- 长链接一定要稳定，不能聊着聊着断开了

- ◆ 高可用

- 一台机器挂了，我还可以连另一台

- ◆ 低延迟

- 一个人发的一句话，另一个人立马收到

实时推送一直播弹幕



话题的力妹23333

萝力 辅导员 UL 24 ZZ依J张:

起不来就多睡一会。-

RubyWQT: 就赖床

Fight 蒜帝 发射飞波 X 1

UL 1 毒岛: 椰椰?

房管 萝力 砂糖战士 UL 35 嘻嘻

frank: 就摸鱼呗, 还能怎样(▽▽)

UL 11 红焗毛线: 素颜更美丽

超级无敌小污女: 拿豆腐撞头

房管 萝力 拔凉 UL 20 妹的诗: 这是

力妹的惯例

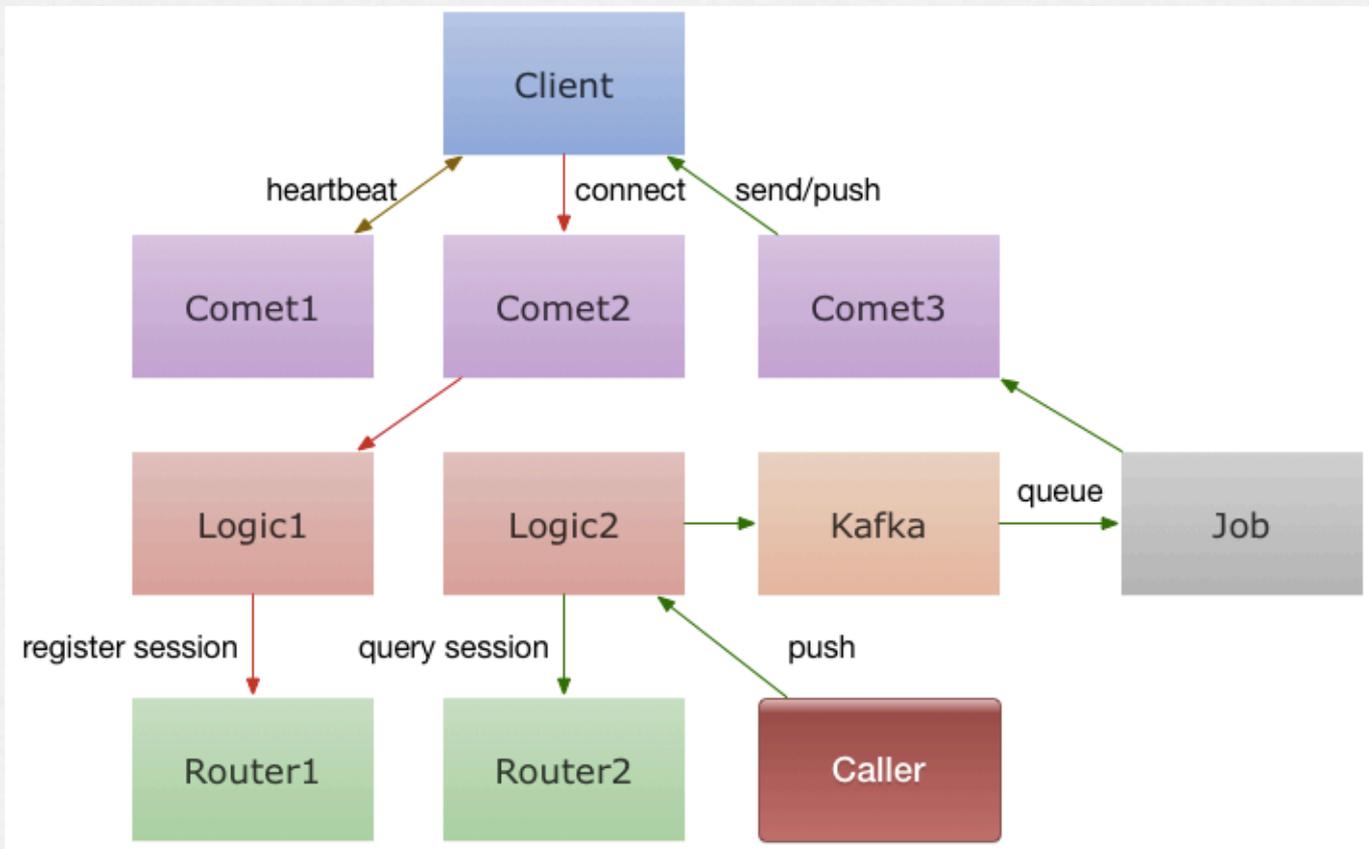
秋天该很好 赠送团扇 X 1

神王雷 老爷 进入直播间

UL 18 冬冬雅: 哦~懒床

0 / 30

直播弹幕—服务构架(GoIM)



直播弹幕一模块

- ◆ Comet
 - 维持客户端长链接、心跳.
- ◆ Logic
 - 逻辑处理, 如账号验证、业务判断
- ◆ Router
 - 在线信息(session)存储
- ◆ Job
 - 消息分发
- ◆ Kafka - 第三方服务
 - 队列服务

高稳定性—内存优化

- ◆ 一个消息一定只有一块内存
 - Job聚合消息, Comet指针引用
- ◆ 一个用户的内存尽量放到栈上
 - 内存创建是放在对应的用户Goroutine里的
- ◆ 内存是自己控制的
 - 查看模块各个分配内存的地方, 使用内存池

高稳定性—模块优化

- ◆ 消息分发一定是并行的互不干扰的
 - Job与每个Comet链接都是独立的Goroutine
- ◆ 并发数一定是可控的
 - 每个需要异步处理开启的Goroutine都是预先创建好固定个数
- ◆ 全局锁一定是被打散的
 - Socket链接池管理、用户在线数据管理都是多把锁

2015年末压测数据

CPU	内存	数量
E5-2630	DDR3 32GB	2

- * 不同UID同房间在线人数：500,000(每台250,000)
- * 持续推送数量：50条/秒
- * 推送到达：2,440万/秒
- * 推送内容：{"test":1}
- * 推送类型：单房间推送
- * 资源使用：1400%~2340%（刚满）
- * 内存使用：4.22 GB
- * 流量：3.19 Gbit/s

2016年三月压测数据

CPU	内存	数量
E5-2630	DDR3 32GB	1

- * 不同UID同房间在线人数：1,000,000
- * 持续推送数量：40条/秒
- * 推送到达：3,590万/秒
- * 推送内容：{"test":1}
- * 推送类型：单房间推送
- * 资源使用：2000%~2300%（刚满）
- * 内存使用：14 GB
- * 流量：4.39 Gbit/s

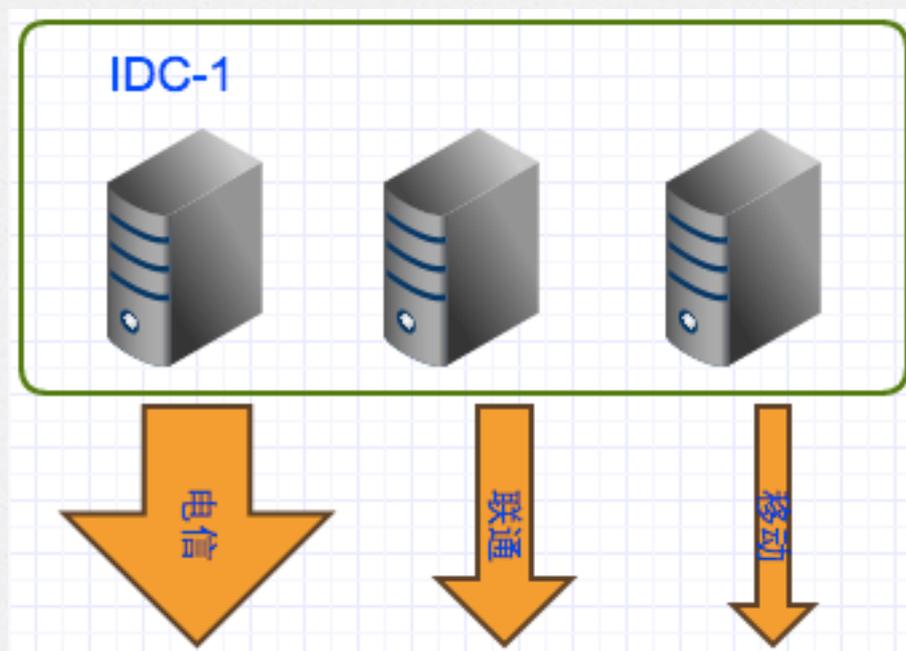
模块优化结果

直播弹幕优化的最终瓶颈只有流量

高稳定性—网络优化

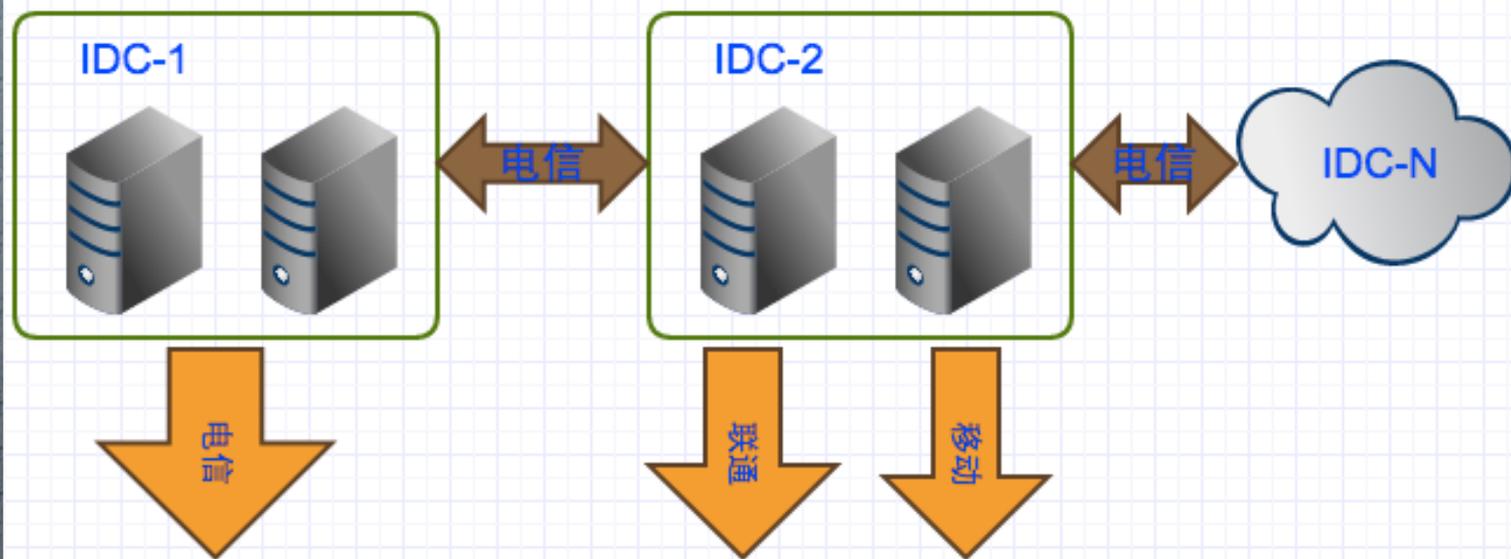
◆ 早期部署结构

- 单线IDC流量不足
- 单点问题
- 接入率低



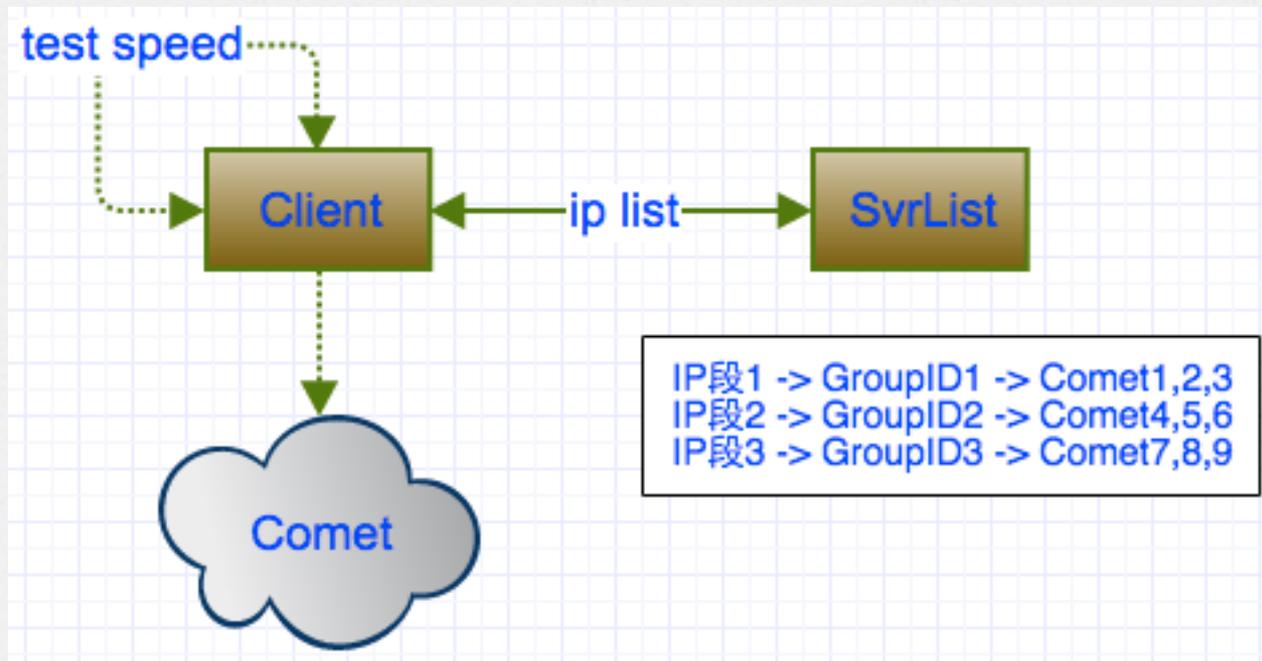
高稳定性—网络优化

◆ 现在部署结构

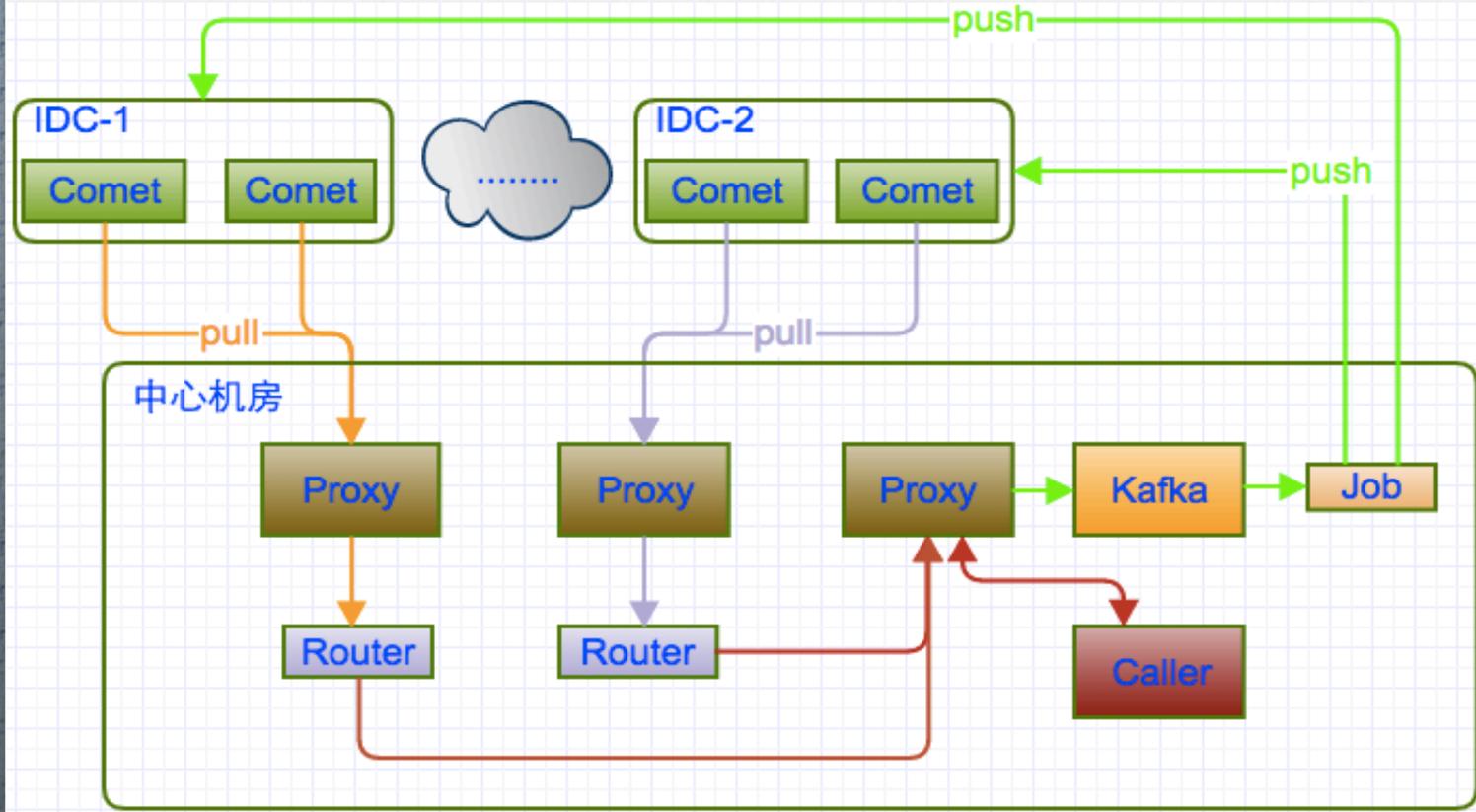


高可用性

- ◆ 即将部署结构，如下图：



高可用性



故障监控

- ◆ 模拟Client监控每秒消息到达
- ◆ 线上开启Pprof,随时可以抓图分析进程状况
- ◆ 白名单功能: 指定人打印服务端日志
- ◆ 服务器负载监控, 短信报警

多IDC方案—新的里程碑，路还很长

- ◆ 不断的优化、调节用户最优接入节点
- ◆ IDC的服务质量监控：掉线率
- ◆ 自动切走“失联”服务器
- ◆ 消息的100%到达：丢包率
- ◆ 流量控制

谢谢