



高可用通讯架构

王春来@陌陌

WANG.CHUNLAI@IMMOMO.COM

应用场景



- ✱ 即时通讯聊天
- ✱ 游戏服务器
- ✱ 其他长连接消息推送

陌陌通讯数据



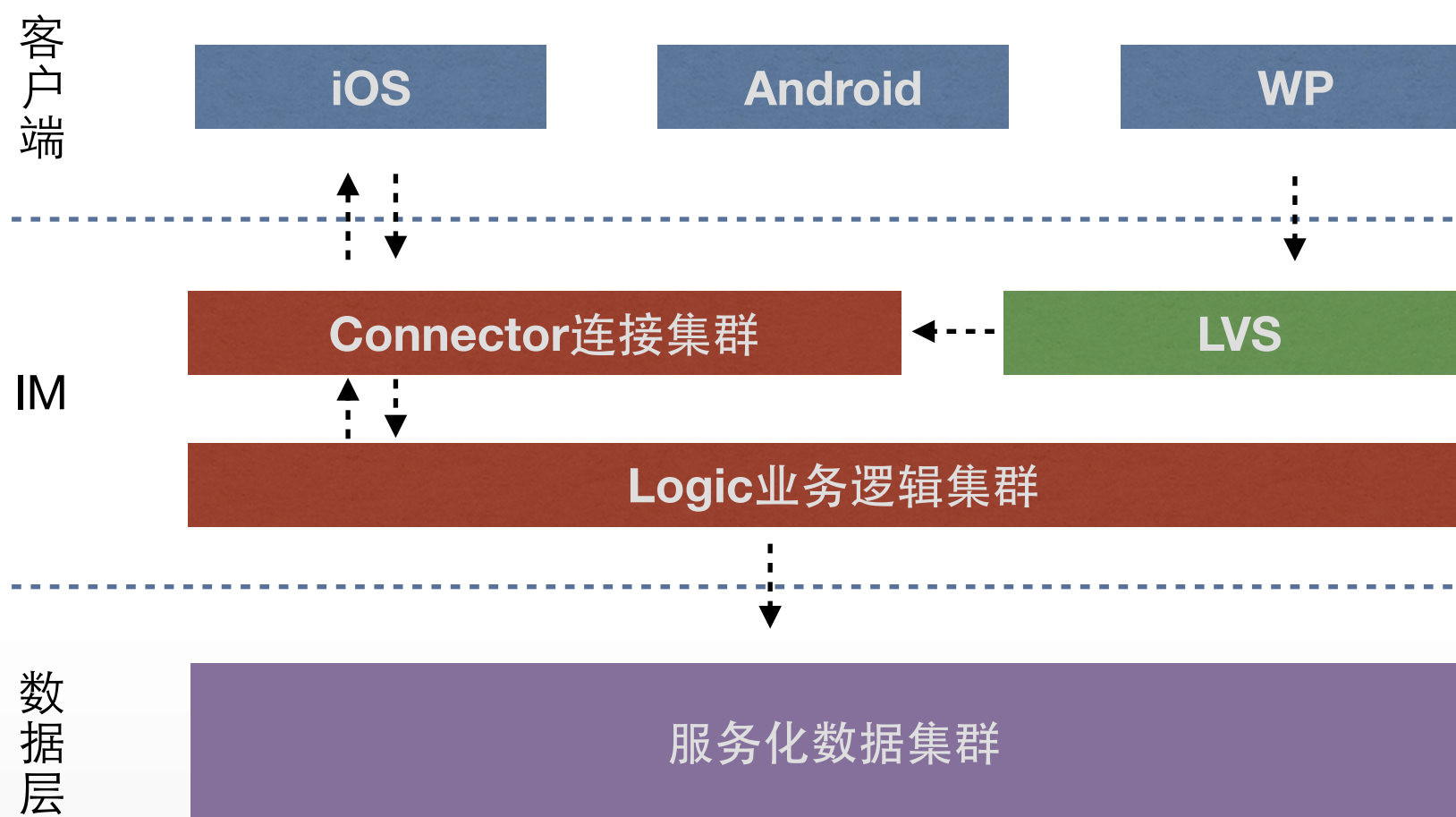
- * 2011.8 发布1.0版本
- * 2012.2 100万用户 1台 openfire+xmlpp
- * 2012.8 1000万用户 10台 自研通讯架构和协议
- * 2014.2 1亿用户 50台 服务化
- * 2014.8 1.7亿用户 70台

陌陌通讯数据



- ✱ 同时在线连接数1200万+
- ✱ 单台服务器正式环境压测100万长连接
- ✱ 每天收发消息量70亿

通讯总体架构

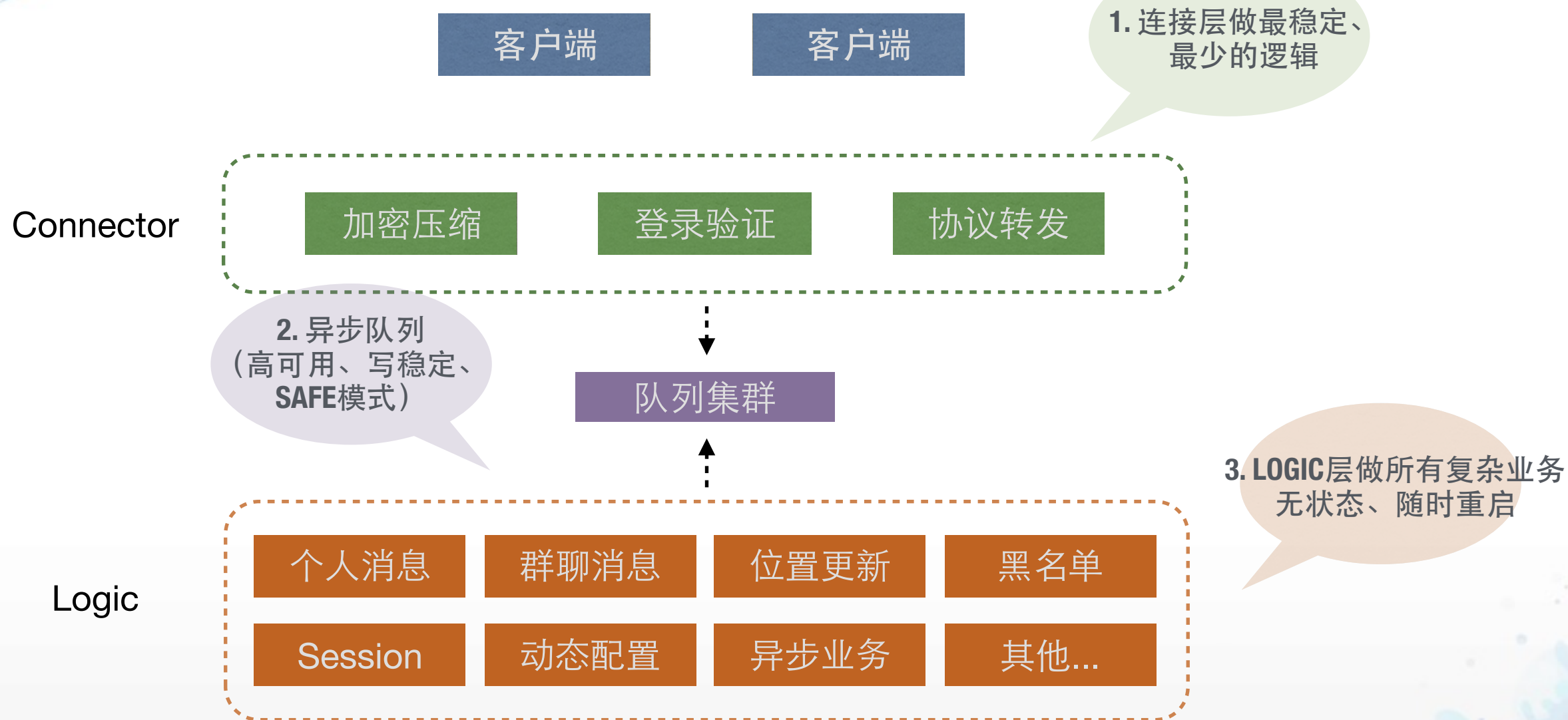




工作的目标就是解决问题



业务更新不断线



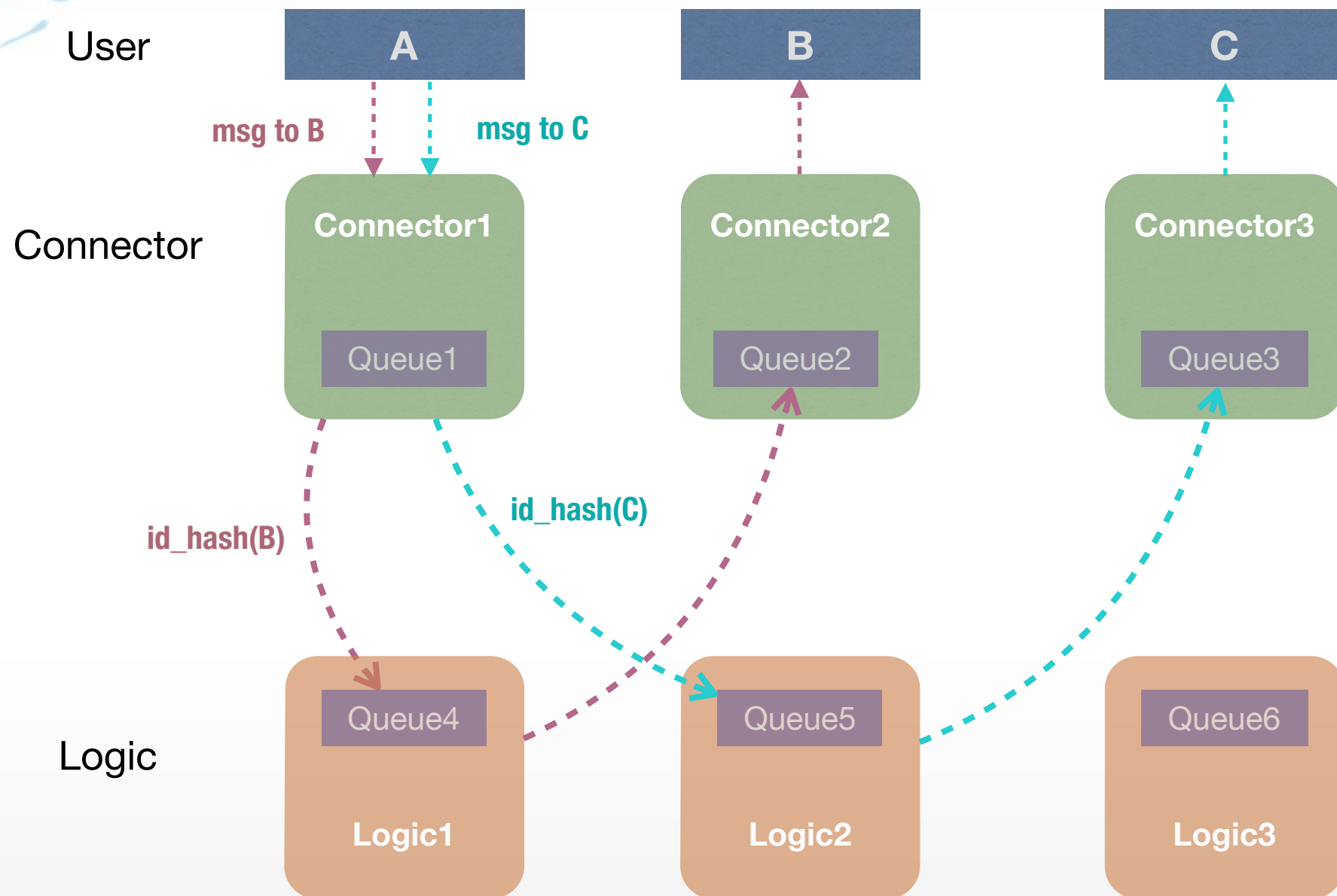


通讯服务器只做需要长连接场景的服务，
其他业务都用http请求搞定: nginx+php

- 部署简单、快速
- 减轻通讯服务器压力



性能优化



长连接动态调整

- 压力均衡、资源浪费、整体承载能力下降
- GC问题

查询目标用户SESSION

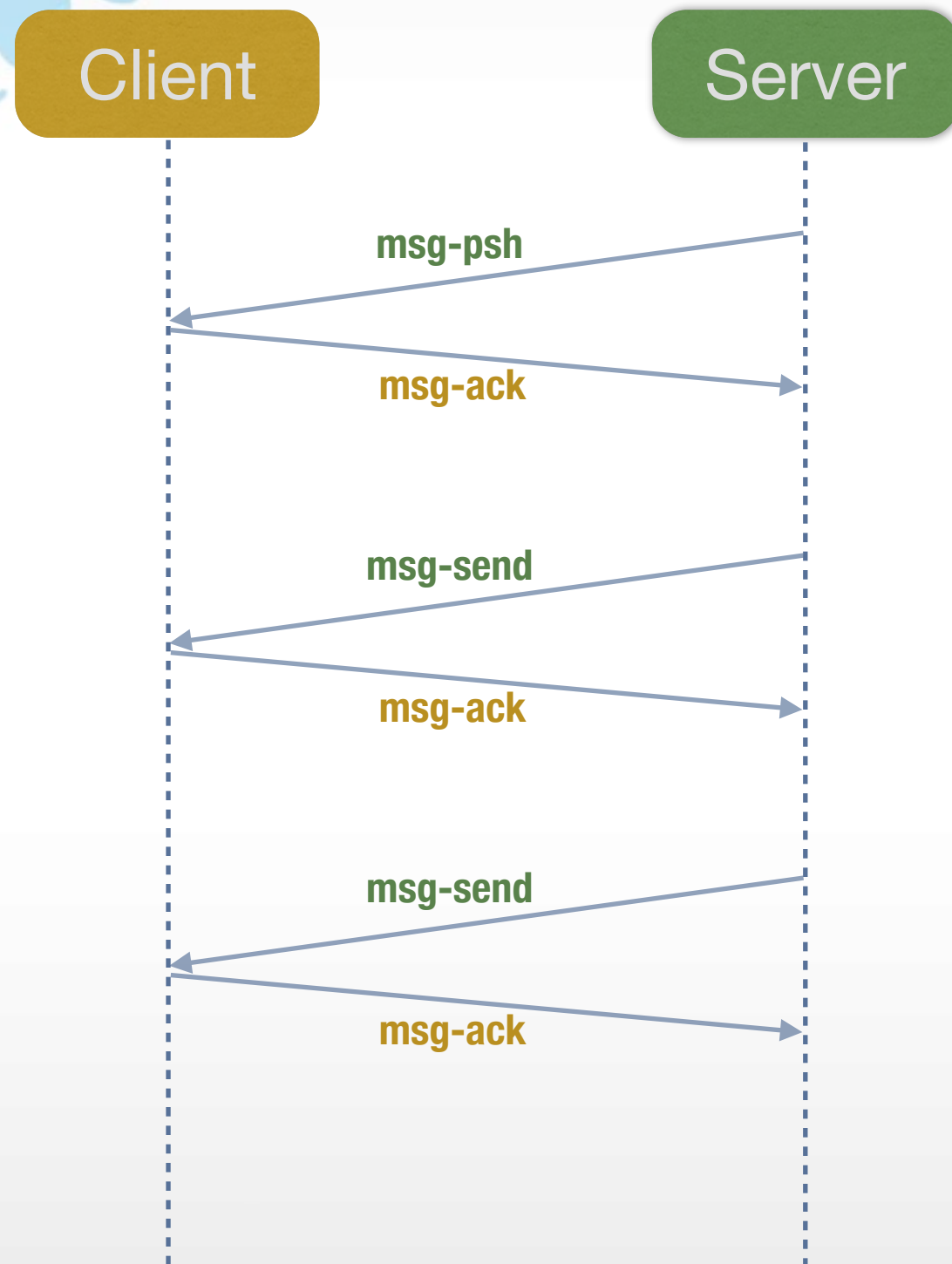
JVM缓存提升性能

需要有消息总线更新JVM缓存

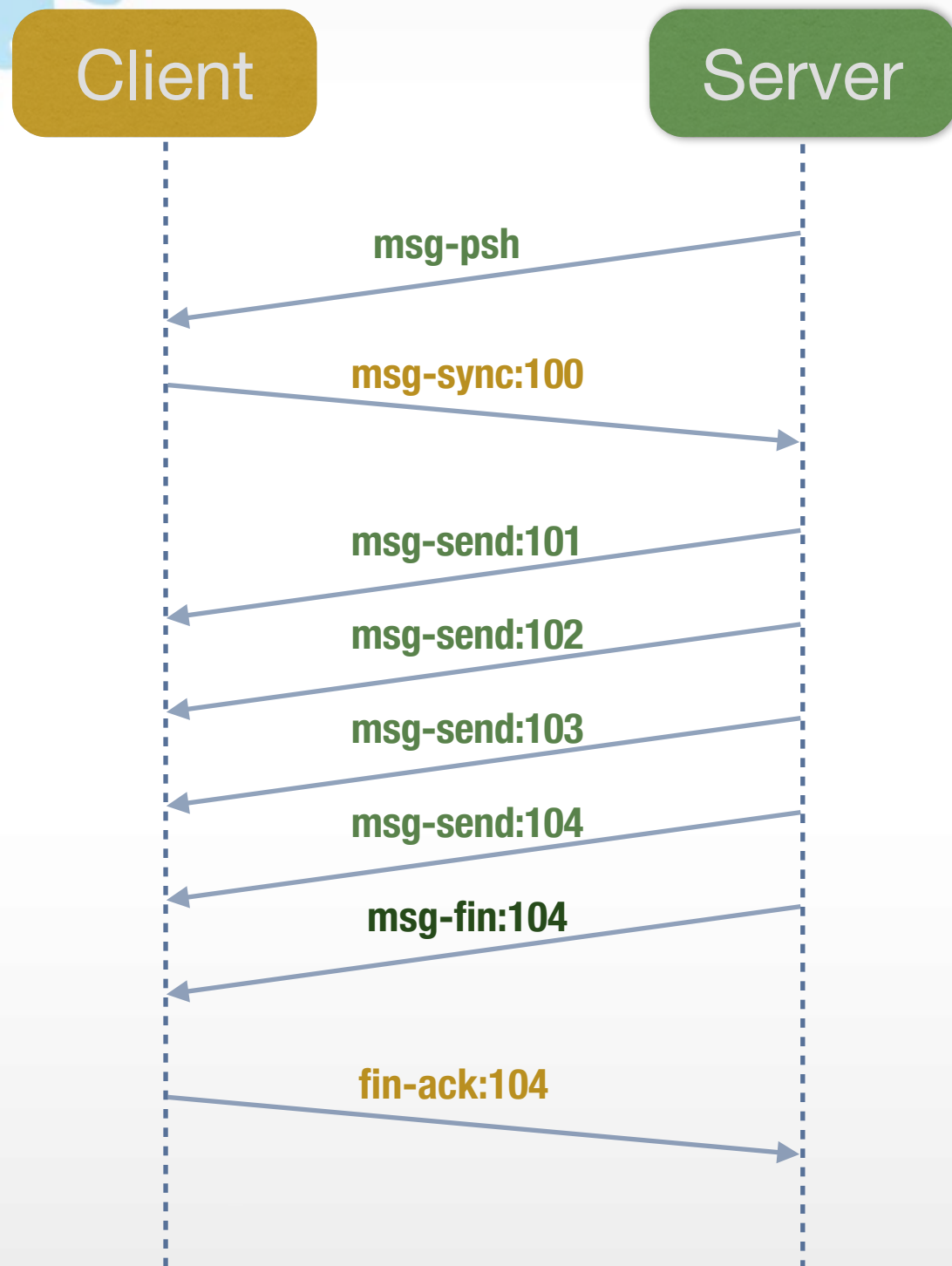
利用HASH分布的弱状态性提升缓存命中率，95%



解决弱网络和丢消息问题



利用msg-ack解决丢消息问题,
但是效率太低



减少C/S交互次数

- $n + 4 < 2n$ ($n > 4$)

基于消息版本号

- 保证消息顺序
- 兼容群消息模式



90%故障来自于变更



灰度发布

- 执行力
- 自动化
- 开关设计

2014-9-7 10:24:35 -- 10.83.65.160										
cmd	cmdtime	htime	cmdq	work0	workq0	error	cmdc_p	cmdc_t	redis_p	redis_t
5416	0	0	198	7/200	92/200	0	1696	0	12593	0
5416	0	0	198	7/200	98/200	0	1696	0	12593	0
5333	0	16	169	2/200	99/200	0	1824	0	13224	0
5530	0	10	157	3/200	97/200	0	1710	1	12743	1
5260	0	11	181	5/200	99/200	0	1880	1	13542	1
5562	0	13	161	3/200	82/200	0	2009	0	14455	0
5324	0	9	156	7/200	89/200	0	1842	0	13311	0
5524	0	12	182	3/200	91/200	0	1700	0	12639	1
5466	0	9	168	2/200	83/200	0	2025	0	14647	0
5541	0	23	176	3/200	93/200	0	1973	0	14051	0

监控

- 命令行方式 -> 监控系统
- 嵌入到基础框架

指标名称	当前数值	历史统计曲线图
1 每秒请求量(cmd)		
2 超时量(htime)		
3 队列堆积量(cmdq)		
4 错误量(error)		
--		
5 每秒请求量sum(cmd_sum)		
6 超时量sum(htime_sum)		
7 队列堆积量sum(cmdq_sum)		
8 错误量sum(error_sum)		
--		
9 总用户(tot-ses)		
10 物理连接阶段登陆用户(iop-ses)		
11 逻辑连接阶段待登陆用户(p-ses)		
12 登陆用户(session)		
13 苹果客户端(ios)		
14 英文版苹果客户端(eios)		
15 安卓客户端(android)		
16 微软客户端(wp)		

必要的监控点



1. 每个执行单元的超时数量以及详细日志
2. 队列堆积数量、线程占用数量
3. io
4. 缓存命中率
5. gc
6. 数据包处理数量

手机实时运维



随时查看运营数据、服务器指标数据

出现异常情况临时修复，为拿到开发环境争取时间



减少故障影响面



1. 业务线程隔离互不影响，任何服务挂掉不影响聊天
2. 对其他系统的服务都尽量设计成弱依赖
3. 支持降级，各个业务的动态开关
4. 定期的故障演练，检验系统优化方案，加快故障修复时间



其实通讯服务器没想象中复杂，解决好上述问题
支撑千万级用户就不是问题

谢谢!

@Chunlai