

**Think**  
ThinkInLAMP.com

**PHPCON**  
JUL11-12  
SHANGHAI CHINA

2015 PHPCON SHANGHAI





# PHP在移动电商领域的实践



有赞，移动电商服务平台

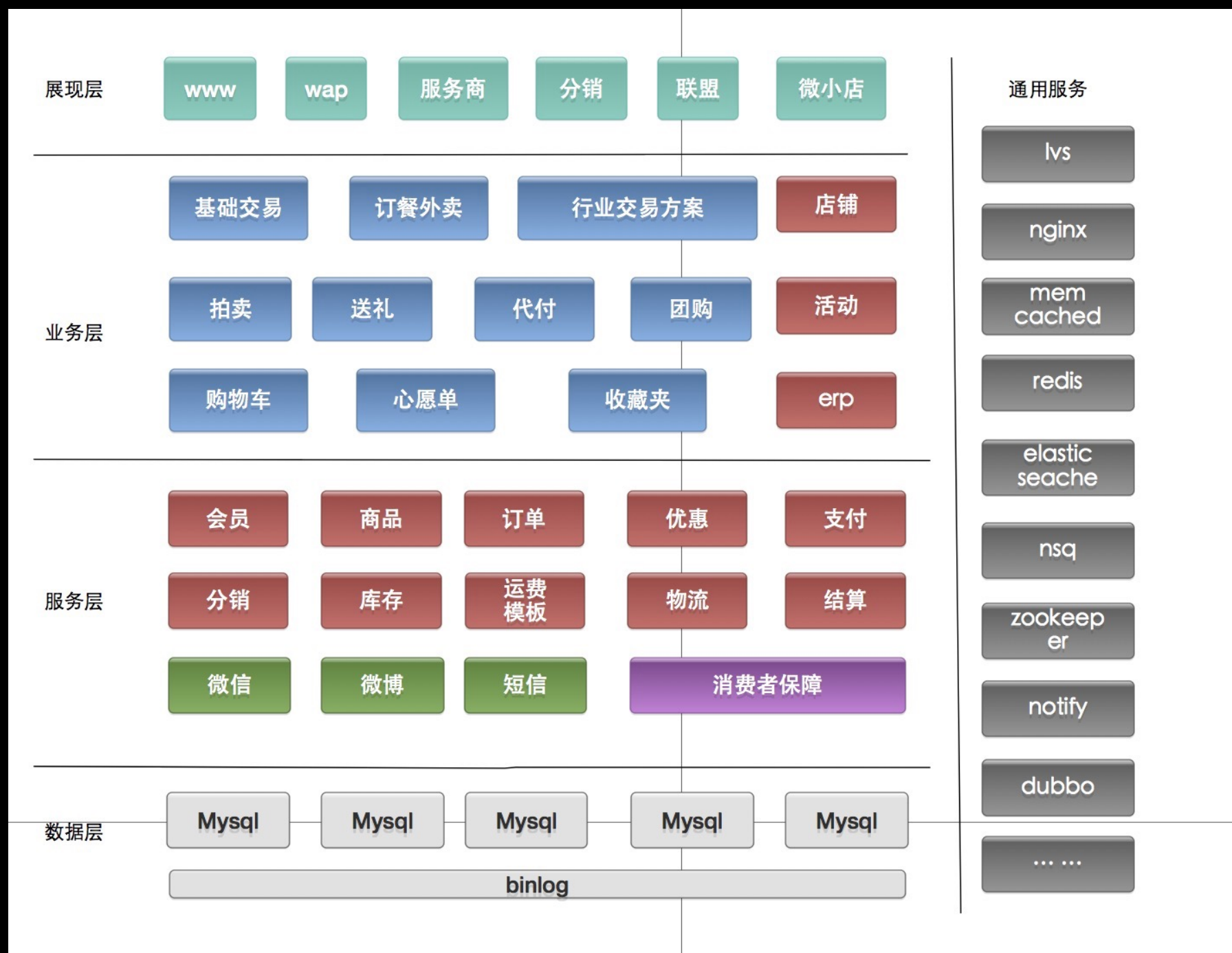
# 最初的设想

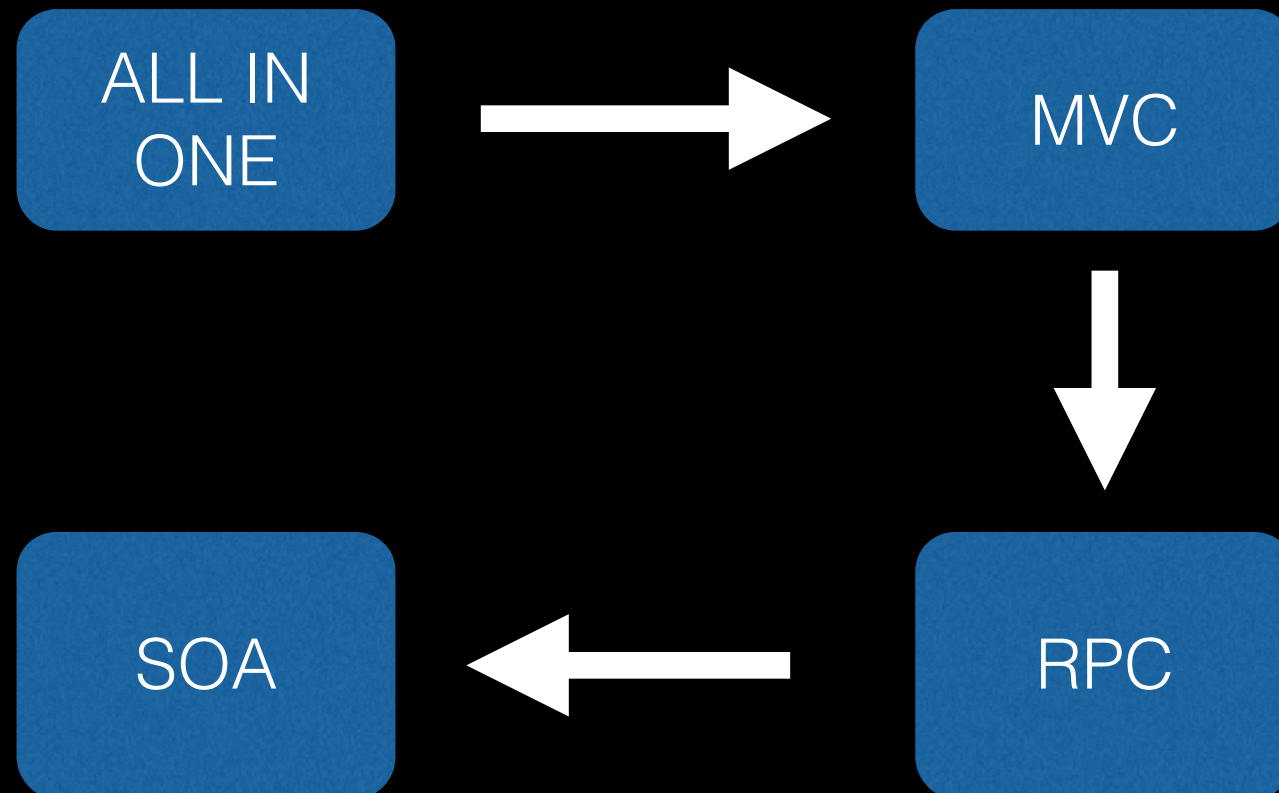
- SCRM
- 技术栈：YAF，虚拟机，Linux，VM上建数据库，没有备份，没有HA，自己处理静态文件
- 人才储备：BOSS+主程（副BOSS）+一枚实习生

# 现状

- 公司200人
- 技术120+
- 产品线  $< 5$
- 专业卖家 200w+(两年半的积累)
- 个人卖家 300w+(4个月)
- 月度覆盖消费者 2亿+
- 日处理消息数量  $> 1$ 亿









# 第一个阶段

- 数据计算的问题
- UV,PV，访问深度，粉丝增减，活跃度，消费频次,优惠券统计，投票调查，等等
- 爬虫问题
- 延时任务问题

# 解决办法

- 更换基础服务提供商
- 使用“云数据库”
- 使用外部云存储和CDN（兼容性：先上传到自己服务，再异步同步到云存储）
- 使用Python替代PHP做后台服务
- 引入Mongodb存储用户信息
- Scrapy构建自动爬虫系统

# 第二个阶段

- Web机器开始增多，FPM进程数增多
- 开始多个子项目，子项目之间的共享业务逻辑
- 云数据库扛不住
- 单表数据过大（最多超过3亿条，150G)
- 更多的报表和实时计算的需求
- 监控和量化

# 解决方法

- YAR
- 从云数据库迁移到物理机（64G 32core SAS盘）
- 一主多从（mysqlnd->atlas）
- 全面转向CDN，静态文件全部存储在云存储
- 部署更大规模的Hadoop+storm集群

# 更细致的优化

- 给Nginx和FPM之间设置超时时间（推荐10s以内）
- PHP和MySQL之间设置连接超时时间
- 使用Tengine 替换原生Nginx
- 使用最新版supervisor来管理PHP后台进程服务
- 适量的FPM进程数（FPM 只能解决Hold的连接数，最终的性能取决于每个Request的响应时间）
- 使用tsar 来给数据库和应用进行诊断

Time	-----nginx-----								
Time	accept	handle	reqs	active	read	write	wait	qps	rt
07/07/15-23:04:37	453.00	453.00	807.00	12.0K	0.00	100.00	11.9K	807.00	153.83
07/07/15-23:04:38	507.00	507.00	788.00	12.0K	0.00	120.00	11.9K	788.00	111.07
07/07/15-23:04:39	543.00	543.00	846.00	12.1K	0.00	112.00	12.0K	846.00	141.73
07/07/15-23:04:40	557.00	557.00	851.00	12.1K	0.00	123.00	11.9K	851.00	123.74
07/07/15-23:04:41	468.00	468.00	747.00	12.0K	0.00	114.00	11.9K	747.00	94.25
07/07/15-23:04:42	535.00	535.00	815.00	11.9K	0.00	117.00	11.8K	815.00	140.95
07/07/15-23:04:43	413.00	413.00	786.00	11.9K	0.00	90.00	11.8K	786.00	134.78
07/07/15-23:04:44	500.00	500.00	820.00	11.9K	0.00	112.00	11.8K	820.00	158.84
07/07/15-23:04:45	443.00	443.00	757.00	11.9K	0.00	106.00	11.8K	757.00	137.94
07/07/15-23:04:46	615.00	615.00	912.00	12.0K	1.00	118.00	11.8K	912.00	90.89
07/07/15-23:04:47	467.00	467.00	854.00	11.9K	0.00	127.00	11.8K	854.00	77.87
07/07/15-23:04:48	448.00	448.00	760.00	11.9K	0.00	119.00	11.8K	760.00	182.74
07/07/15-23:04:49	431.00	431.00	727.00	11.9K	0.00	109.00	11.8K	727.00	215.47
07/07/15-23:04:50	468.00	468.00	776.00	11.9K	1.00	97.00	11.8K	776.00	312.88
07/07/15-23:04:51	562.00	562.00	832.00	11.9K	0.00	93.00	11.8K	832.00	96.65
07/07/15-23:04:52	593.00	593.00	914.00	11.9K	0.00	97.00	11.8K	914.00	135.25
07/07/15-23:04:53	681.00	681.00	971.00	12.0K	0.00	107.00	11.9K	971.00	102.36
07/07/15-23:04:54	585.00	585.00	888.00	12.0K	0.00	103.00	11.9K	888.00	99.11
07/07/15-23:04:55	644.00	644.00	928.00	12.0K	0.00	98.00	11.9K	928.00	128.23

# 第三个阶段

- 发布管理问题
- FPM继续增多，带来的连接池问题
- 过渡依赖缓存问题
- PHP处理异步任务的问题
- 业务水平拆分问题
- 多维度数据库分拆问题
- PHP性能问题
- 全文检索的需求变得迫切

# 解决方法

- 收紧发布入口，推进模块化发布
- 前端从主仓库中剥离
- 增加数据库中间件做连接池和Hash（目前使用cobar）
- 使用NSQ作为队列，将大部分耗时或者异构的处理异步
- 数据库按照买卖双方维度进行冗余拆分
- 使用Opcache，升级PHP版本
- 缓存集群化，目前部署超过300G内存的 32个实例 Codis
- ElasticSearch 集群部署+性能调优+优化源码



# 第四阶段

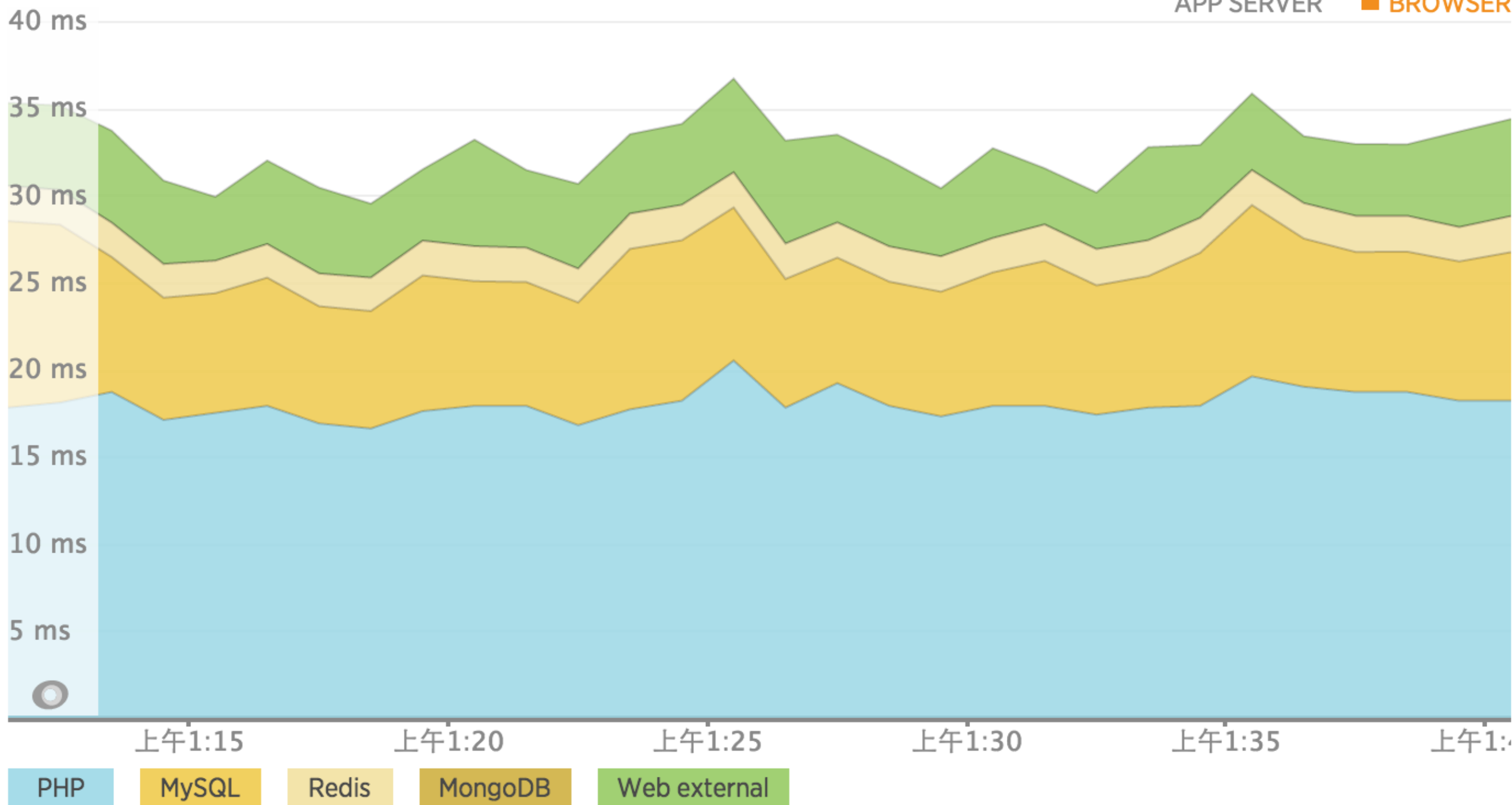
- PHP服务拆分带来的问题
- 水平扩展后中间件带来的问题
- 更细致的监控和恢复策略
- 数据库大字段带来的性能损耗
- 读写分离后写库的极限性能问题
- 分库分表后离线计算的数据同步问题

- Local Proxy + 长连接
- 重新设计中间方案（可运维+可扩展+高可用）
- 引入外部监控系统（监控宝+APM监控）
- 自己研发监控系统（PHP堆栈信息 JVM信息 GC信息）
- 每个数据库节点都是2 Master+1 slave
- 构建分布式KV系统（基于aerospike）
- SQL KILLER
- 更换DB 集群为SSD
- 基于Binlog的异构数据源复制（基于canal+自研）

## Web transactions response time ▾

32.6 ms  
APP SERVER

2.65 s  
BROWSER





[job.youzan.com](http://job.youzan.com)

只和最优秀的人一起打拼

薪资  $\geq$  30K



与你同行，是我们的荣幸。