

蘑菇街运维体系 及双11关键技术分享

演讲人：赵成

Geekbang>

极客邦科技

整合全球最优质学习资源, 帮助技术人 and 企业成长
Growing Technicians, Growing Companies

InfoQ
new

专注中高端技术人员的
技术媒体



EGO EXTRA GEEKS' ORGANIZATION
NETWORKS

高端技术人员
学习型社交网络



StuQ
new

实践驱动的
IT 职业学习和服务平台



GiT GEEKBANG
INTERNATIONAL
TRAINING
极客邦培训

一线专家驱动的
企业培训服务



旧金山 伦敦 北京 圣保罗 东京 纽约 上海
San Francisco London Beijing Sao Paulo Tokyo New York Shanghai

QCon

全球软件开发大会

2016年4月21-23日 | 北京·国际会议中心

主办方 **Geekbang**  **InfoQ**
极客邦科技

7折 优惠 (截至12月27日)
现在报名, 节省2040元/张, 团购享受更多优惠

www.qconbeijing.com



扫描获取更多大会信息

关于蘑菇街 About us

中国最大的女性时尚社交电商平台。成立于2011年，总部位于浙江杭州，目前（2015. Q3）拥有1.3亿注册用户，双十一日UV超2000万。

11.21日宣布完成D轮融资，并实施“一街双城”战略，杭州+北京，杭州偏电商方向，北京偏社交媒体方向。



自我介绍 About me

- 姓名：赵成
- 花名：谦益（谦受益，满招损。——《尚书·大禹谟》）
- 蘑菇街-平台技术-运维经理

主要讲什么

我们的运维体系

- 蘑菇街技术架构演进
- 蘑菇街运维体系演进
- 运维该怎样提供更好服务

双11技术保障，运维做了什么？

- 全链路压测
- ATS静态化方案

蘑菇街业务架构-导购期 (2011-2012)

前台业务:



后台系统:



基础应用:



蘑菇街技术架构-导购期 (2011-2012)

速度要快-LNMP

PHP是最好的语言

简单是最好的
-几十台机器

维护? 脚本搞定



80%



20%

代理层

Nginx Proxy

WEB层

Nginx+PHP-FPM

Nginx+PHP-FPM

基础设施层

MySQL

Master

Slaver

Redis

Lucene

爬虫

运维早期情况

- 早期阶段（2011-2012年）
 - 两位数机器、个位数网络设备
 - 没有运维，开发即运维，靠牛逼的脚本和一些开源工具搞定

蘑菇街业务架构-转型期 (2013)

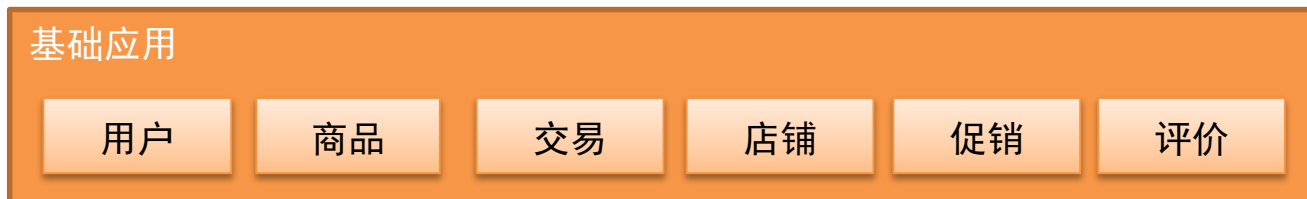
外部频道:



业务系统:



基础应用:



蘑菇街技术架构-转型期

速度还要快

PHP是最好的语言

容量不够？扩容

维护？脚本+简单
维护平台

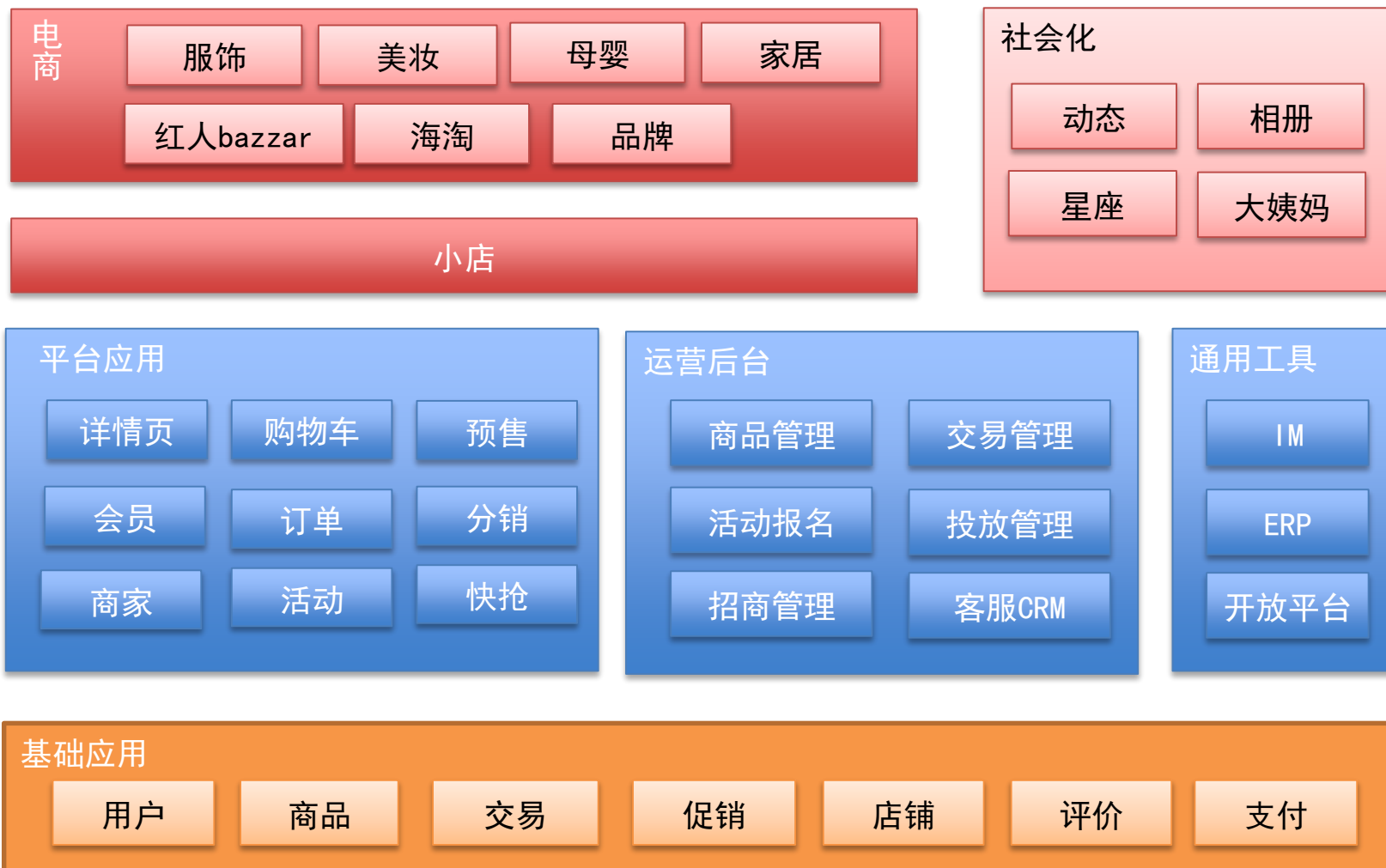


运维的发展

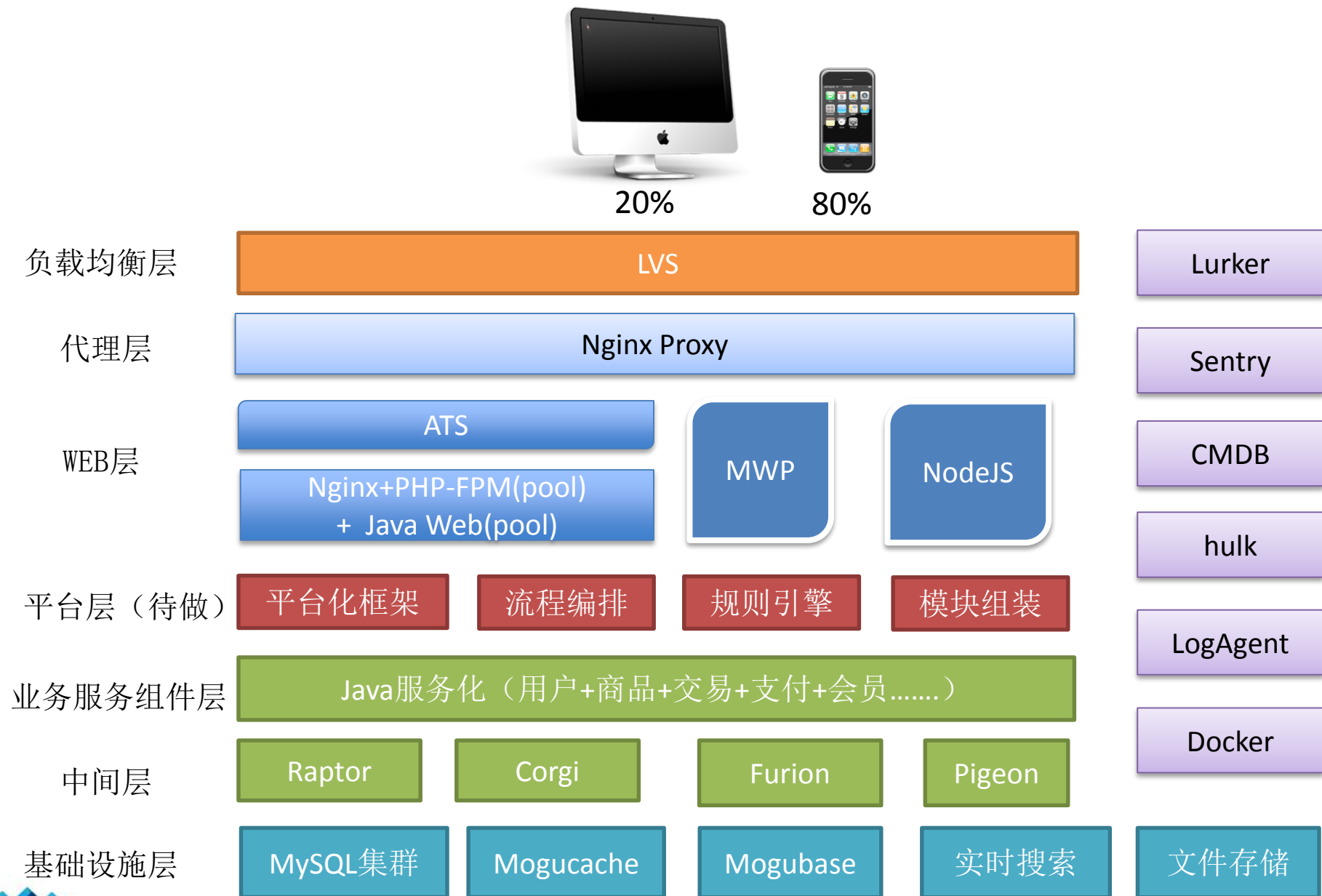
- 中间阶段（2013年-2014年）
 - 三位数服务器、两位数网络设备
 - 2-3名专职运维同学（主机&网络&DB&缓存&.....）
 - 问题响应式的工作方式
 - 工具化的运维平台
 - 机器资源管理（CMDB的雏形）
 - PHP发布系统
 - 从指标维度监控系统（主机、QPS、RT、调用次数....）



蘑菇街业务架构-社会化电商



蘑菇街技术体系现状-社会化电商



业务的快速发展对运维带来的挑战

- 问题1：业务线越来越多，机器越来越多
 - 挑战：更合理和高效的资源管理及调度效率
- 问题2：版本和代码迭代周期更短，随着服务化推进，开发速度更快
 - 挑战：发布速度不能落后，减少人肉成本
- 问题3：业务上线后，以前只要监控主机信息和通用的QPS、RT就好了，每个应用越来越多的个性化监控怎么办？
 - 挑战：统一的监控平台，提供定制化的模板

CMDB

发布系统

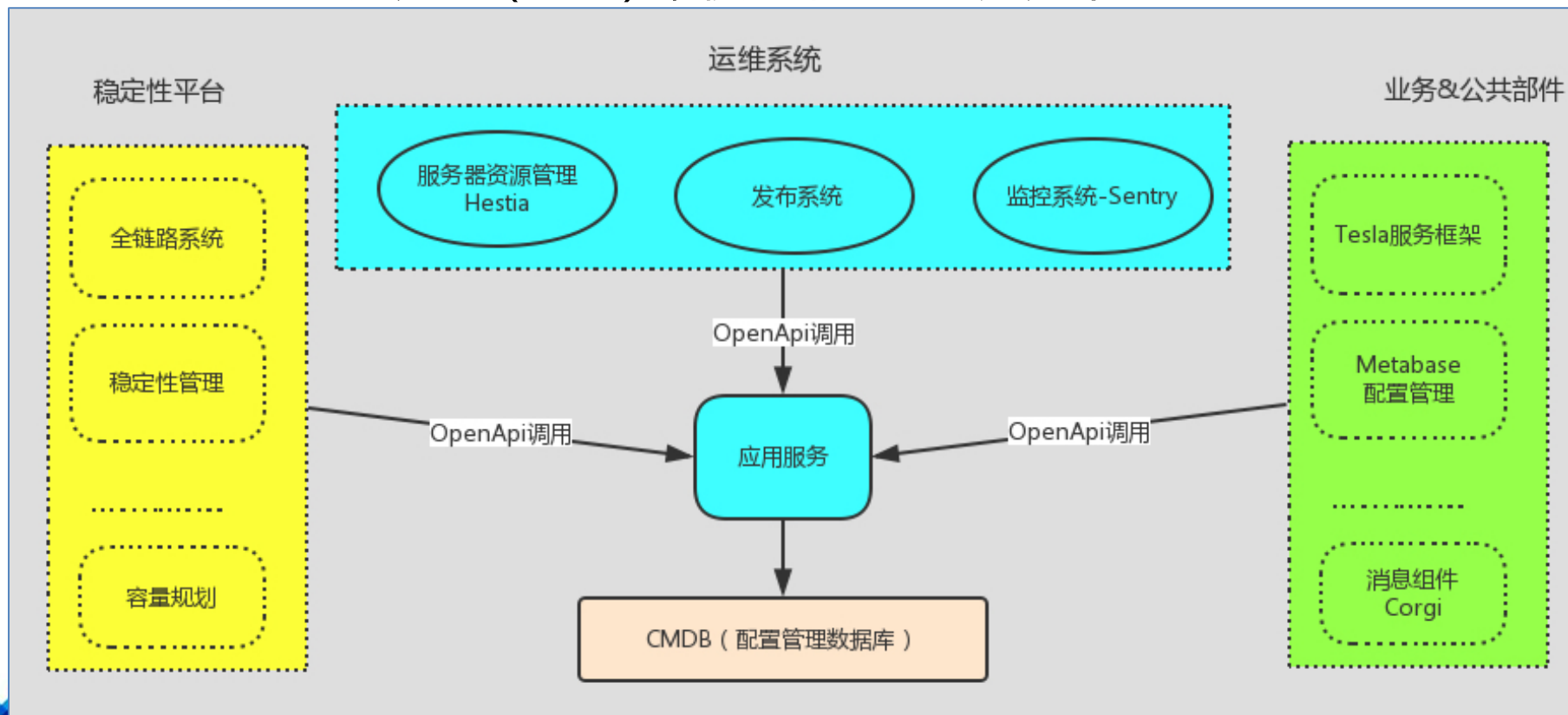
监控平台



我们应该怎么做

思路：

- 建立以**应用服务**为核心的管理标准体系
- 打造CMDB、流程申请、持续集成和监控为一体的自动化运维系统，而不是孤立的单点系统
- 把运维能力服务化（API），使运维的能力无处不在



关于应用服务管理

应用服务管理

- 公司
- 一级部门（研发部）
- 二级部门（电商基础）
- 业务开发部（交易）
- **应用服务**
(trade_ordership_service)
- 应用服务分组

trade_ordership_service 应用服务标识
将会是后续一系列运维系统的桥梁！！



案例介绍

让我们看一个从服务器管理—申请—代码发布—线上监控的案例

关于应用服务管理-Hestia服务和资源管理

- 从业务的维度来管理主机-CMDB的核心概念
- 支持扩容、上下线、设备保障、权限等常规流程申请
- 自动化任务的配置和下发

Hestia

服务管理 网络管理 批量任务

精确 搜索

服务管理 共4台

扩容 online offline 报修 申请权限 批量任务

<input type="checkbox"/>	主机名	IP	状态	PE	负责人	应用分组	操作
<input type="checkbox"/>	mofa016027	16.27	online			trade_ordership_servicehost	详情
<input type="checkbox"/>	mofa015030	15.30	online			trade_ordership_servicehost	详情
<input type="checkbox"/>	qihe12023	12.23	online			trade_ordership_servicehost	详情
<input type="checkbox"/>	qihe11030	11.30	online			trade_ordership_servicehost	详情

全部显示

< 1 >

公司

研发部

电商基础&会员

交易

- trade_php_service
- trade_log
- trade_cart_web
- trade_ordership_service
-
- k_service
- rate_service
-
- ce
- ice

关于应用服务管理-Mops流程申请系统

- 资源不够了，通过应用trade_ordership_service进行扩容申请

Mops-workflow

HELLO, 谦益
qianyi@mogujie.com

包括申请人,标题,指标

待我处理的27

我申请的0

我审批过的154

我申请已完成的0

谦益提交的服务器申请单

申请人 主管 运维主管 运维负责人 结束

标题

谦益提交的服务器申请单

应用

新应用申请点这里trade_ordership_service

应用分组

trade_ordership_servicehost(PE:普智)

应用类型

app

机型

vm_8core20g100g

机房

公司

服务器数量

10

关于应用服务管理-发布系统

- 以trade_ordership_service为标示，进行代码发布

APP信息

id	名称	代码库	创建人	创建时间
193	mudu/trade_ordership_service	http://[redacted]/trade_ordership_service	谦益	2015-12-10 22:59:29

发布信息

id	版本	branch	commit	状态	创建人	创建时间	操作
4582	1.0	master	49f2601d6e446e63a9ec97583cc06318f2b4bb22	build done	谦益	2015-12-10 23:00:10	<button>build</button>

trade_ordership_servi

获取

输入tesla应用名

更新

<input type="checkbox"/>	ip	主机名	应用分组	tesla服务名	机房	应用分组负责人	状态
<input type="checkbox"/>	[redacted]16.27	mofa016027	trade_ordership_servicehost	trade_ordership_service	[redacted]	[redacted]	<div></div>
<input type="checkbox"/>	[redacted]5.30	mofa015030	trade_ordership_servicehost	trade_ordership_service	[redacted]	[redacted]	<div></div>
<input type="checkbox"/>	[redacted].12.23	qihe12023	trade_ordership_servicehost	trade_ordership_service	[redacted]	[redacted]	<div></div>
<input type="checkbox"/>	[redacted]11.30	qihe11030	trade_ordership_servicehost	trade_ordership_service	[redacted]	[redacted]	<div></div>

点击开始分发部署

完成部署

关于应用服务管理-监控系统Sentry

- 通用+自定义监控配置，运维+开发可以时刻关注自己的服务状态和质量

sentry监控

GC

0

Zoom Out

6 hours ago to a few seconds ago

⚙

</> 设置下拉框

Variables

Add

Edit

变量名称

类型

Refresh on load

排序类型

Variable values (showing 20/0)

name

query

☐

不排序(速度快)

对应tag名称

参考metric(确定哪个metric下的tag列表)

All option

☐

Add

Close

运维的现状

- 专业的运维团队
 - 系统运维
 - 应用运维
 - DBA
 - 运维开发
- 运维的能力向平台化和服务化发展（DevOps，依赖于能力而不是人）
 - CMDB服务化平台
 - PHP+Java持续集成发布平台
 - 统一的监控平台
 - 全链路服务质量分析平台
 - 稳定性平台
 - 容量评估平台（待做）
- 工作方式的改变
 - 从问题响应式，向整体解决方案提供方向发展

双11技术保障，运维做了什么？

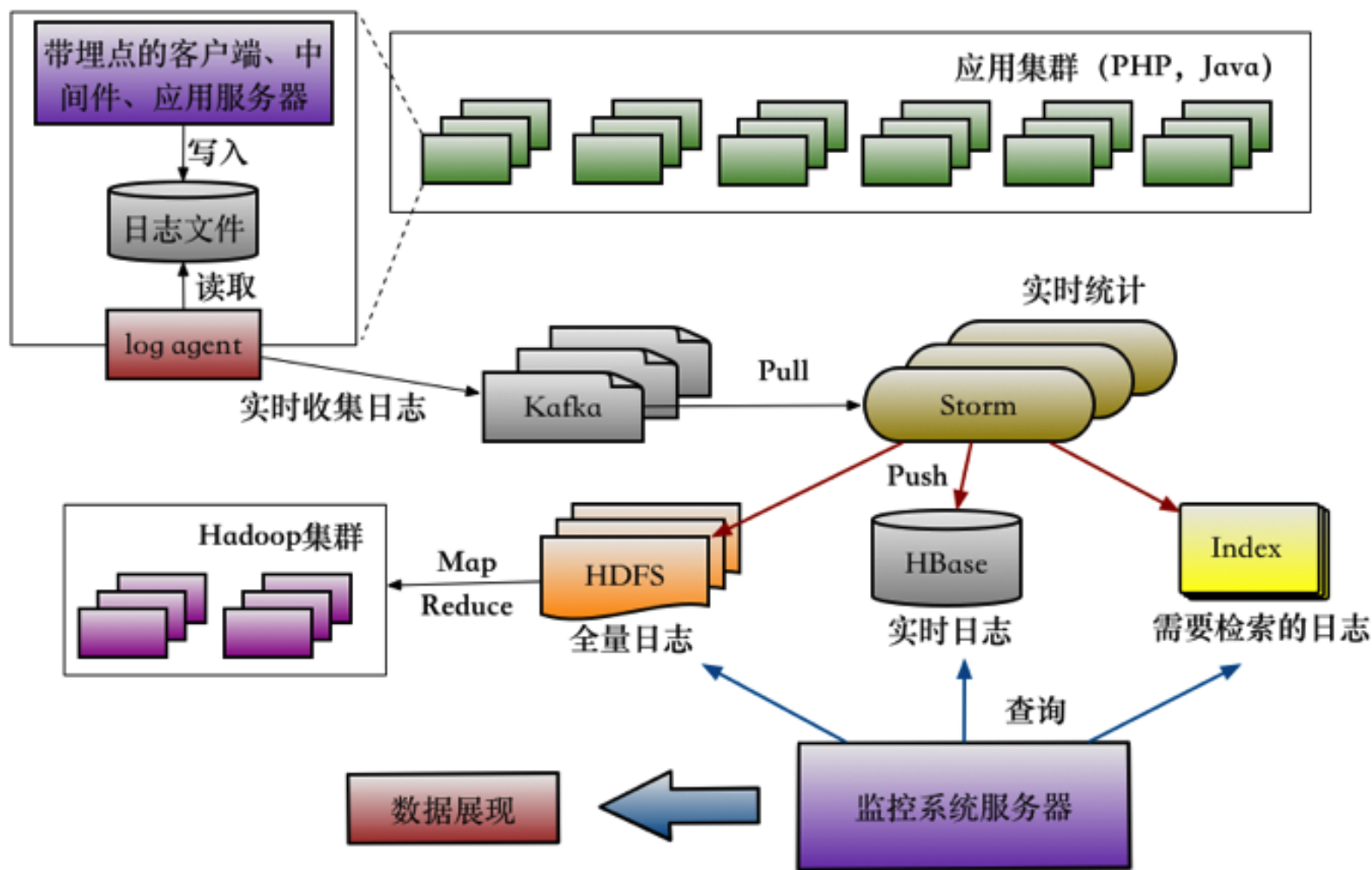
- 双11大促前：
 - 机房&网络&硬件&安全等基础设备保障
 - 大批量采购机器，但是没分
 - **全链路压测，容量评估**
 - **发现问题，技术改造，ATS静态化**
 - 扩容，核心链路增加机器
- 双11期间：
 - 看监控，看数据，看曲线
 - 限流和降级开关推送
 - Web接入层应对峰值的紧急扩容

双11关键技术分享—全链路系统

- 全链路背景
 - 复杂的分布式系统，页面上的一次链接点击，在后端可能会产生几十次的RPC调用，Web、服务化、缓存、消息、DB……都有可能涉及，如果出了问题，如何快速定位到故障点？要扩容，如何合理评估？
 - 关键概念，全局唯一的TraceId

全链路技术架构

全链路监控系统架构



全链路应用-快速发现问题点和瓶颈点

服务名	方法名	type	IP	耗时(ms)	Timeline
nmap_rate_rate_mgj_v3	rateList	PHP	10.17.37.67	1492.60	
SessionService	checkSign	Tesla	10.11.11.154	1.43	
UserService	getUserById	Tesla	10.15.7.25	1.24	
TraceService	getDeviceByRequest	Tesla	10.17.37.67	1.11	
ItemService	getCacheItem	Tesla	10.11.2.205	2.75	
ItemService	getCacheItem	Tesla	10.15.18.15	1.86	
PHP-Cache	GET	Cache	10.17.37.67	1.02	
ShopExtraReadService	queryShopEmployee	Tesla	10.15.18.27	1.93	
RAPTOR	executeQuery	DB	10.15.18.27	1.00	
RateReadService	queryItemRatesWithChildRate	Tesla	10.17.37.67	1334.11	
UserService	getUserByIds	Tesla	<div> 主机名: guomai037067 负责人: 大洪 PE: 大洪 应用名: web_mobile_rate 分组名: web_mobile_ratehost hestia sentry </div>	6.04	
PHP-Cache	MGET	Cache		1.17	
ShopReadService	queryShopByUserIds	Tesla		2.23	
RAPTOR	executeQuery	DB	10.15.18.27	1.00	

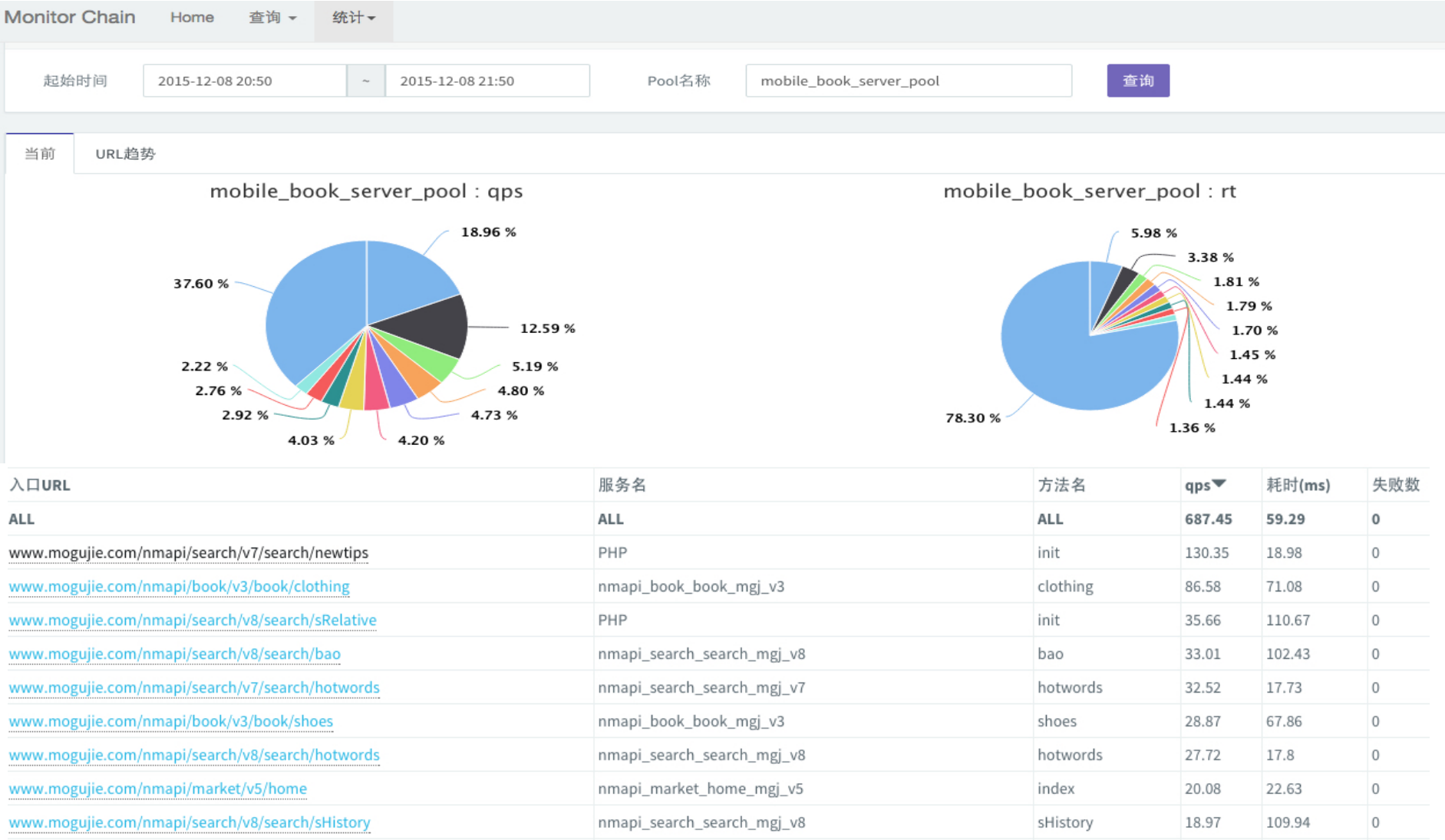


全链路应用-调用合理性分析

没有明显的瓶颈点，每一次调用RT也很正常，但是全链整体的RT却很高，问题又出在哪里了呢？

☐ 1	s_search	search	PHP	10.17.36.151	1802.08	<div></div>	详情
1.1	TraceService	getDeviceByRequest	Tesla	10.17.36.151	10.73	<div></div>	详情
1.2	KeywordService	search	Tesla	10.17.36.151	12.11	<div></div>	详情
1.3	LinksService	getLinks	Tesla	10.17.36.151	2.42	<div></div>	详情
1.4	PHP-Cache	GET	Cache	10.17.36.151	0.88	<div></div>	详情
1.5	PHP-Cache	MGET	Cache	10.17.36.151	6.97	<div></div>	详情
1.6	PHP-Cache	MGET	Cache	10.17.36.151	1.33	<div></div>	详情
1.7	PHP-Cache	GET	Cache	10.17.36.151	1.42	<div></div>	详情
1.8	PHP-Cache	GET	Cache	10.17.36.151	1.01	<div></div>	详情
1.9	PHP-Cache	GET <div>GET</div>	Cache	10.17.36.151	1.18	<div></div>	详情
1.10	PHP-Cache	GET	Cache	10.17.36.151	0.87	<div></div>	详情
1.11	PHP-Cache	MGET	Cache	10.17.36.151	1.72	<div></div>	详情
1.12	PHP-Cache	MGET	Cache	10.17.36.151	0.96	<div></div>	详情
☐ 1.13	topn	rank	HTTP	10.11.6.170	13.77	<div></div>	详情
☐ 1.13.1	DispatchService	dispatch	Tesla	10.15.15.16	10.00	<div></div>	详情

更重要的价值：应用到平时，分析业务和服务质量的关键手段



ArchSummit 全球架构师峰会

Powered by InfoQ


全链路使用后的收益和后续

- 使用全链路后的收益
 - 提升问题的定位效率
 - 准确的评估容量
- 后续
 - Mogu-Watch，与前端打通，实现用户全链路的分析
 - 压测做到平时，与容量评估平台和资源分配打通
 - 引入云资源弹性扩容，避免应对峰值的批量机器采购



压测之后，关键技术改造—ATS静态化方案

- 静态化方案背景和简介
 - 主链路（首页-详情&活动-交易-支付），降低RT，提升容量
 - 资源类的如图片、CSS、JS等的静态化方案都会采用CDN技术
 - 对于页面内容类的数据，如商品名称、商品详情等都属于静态数据，而商品的库存、优惠等则需要获取动态结果
 - 对于活动页面、H5活动推广页面等，则可以完全静态化




鹿与飞鸟原创设计师品牌 秋装新品水洗毛边牛仔裙半身裙长裙

原价: ¥242.00
现价: **¥169.40**

评价: 5 累计销量: 11

客服: [联系客服](#)

颜色: 

尺码: S M L


数量: - 1 + 库存18件

[立即购买](#) [加入购物车](#)

❤ 282 + 分享 举报

服务承诺: 全国包邮 7天无理由退货 5天发货 退货补运费

模特实拍

支付方式: 

[商品详情](#) [累计评价 5](#) [本店同类商品](#) [手机扫码下单](#)

商品描述

秋装新品水洗毛边牛仔裙半身裙长裙

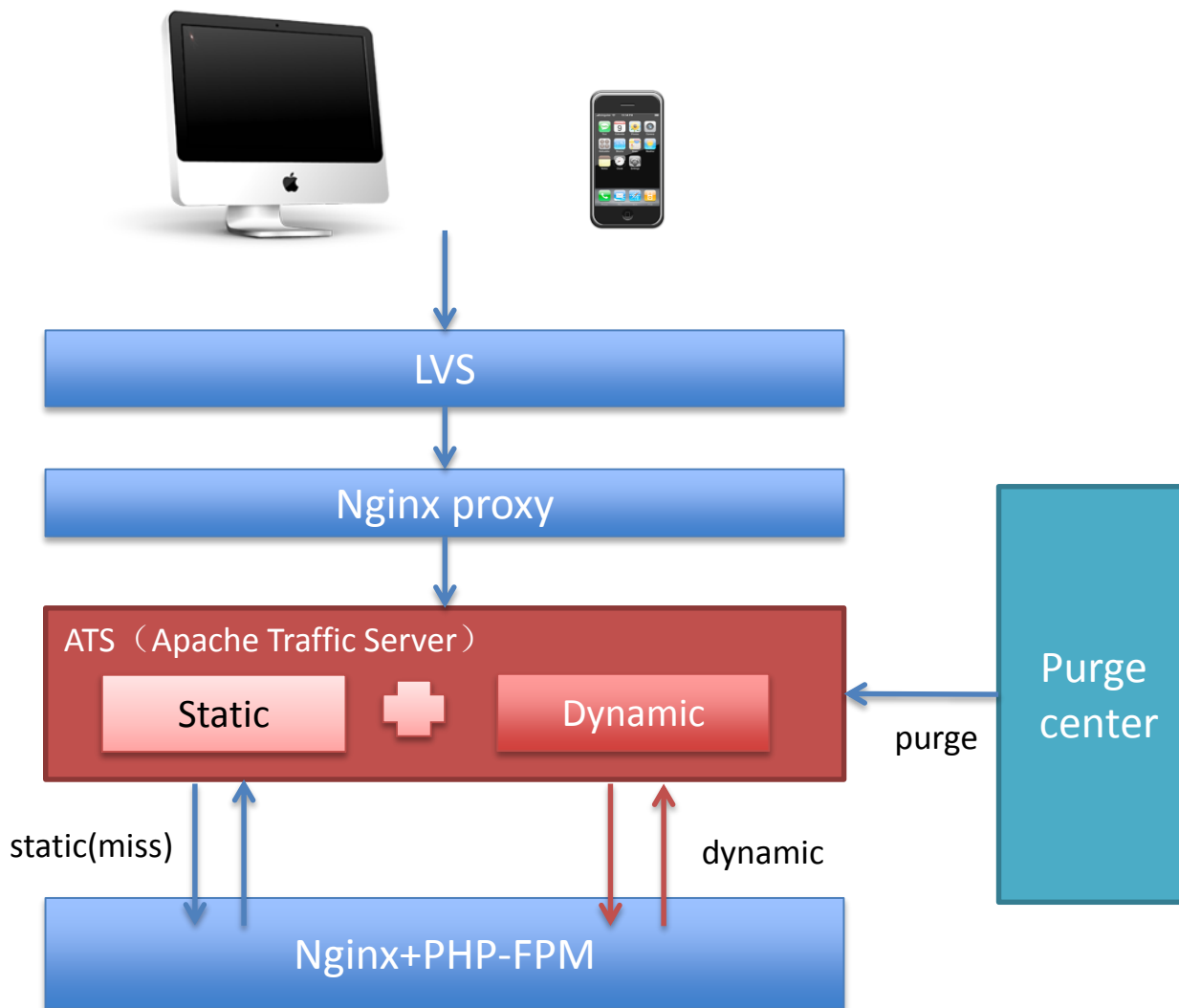
产品参数

材质: 棉	裙型: 直筒裙	裙长: 长裙 (大于65cm)
细节: 水洗,磨白,流苏	货号: 556	风格: 日韩

穿着效果

my closet is filled with a dream
2015

ATS (Apache Traffic Server) 静态化技术方案-Cheetah



ATS静态化案例-商品详情页

- **动静态分离**

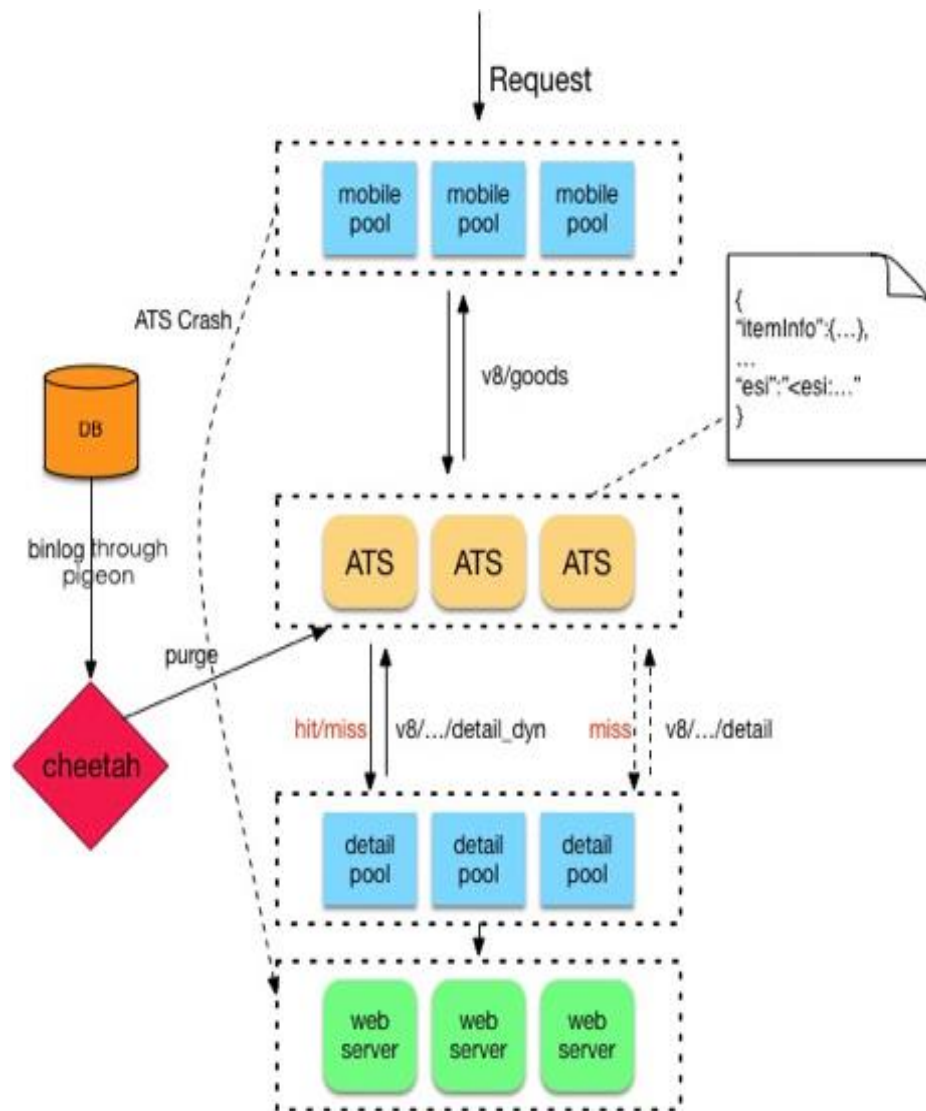
- 一次请求分成静态和动态两次，静态直接ATS处理，动态则回源到业务服务器

- **动态数据获取**

- 使用HTTP的ESI标签进行标示

- **失效机制**

- Purge Center，通过http purge方法执行
- 主动失效，通过Pigeon通知（Mysql binlog订阅系统）
- 定时失效，4 or 6小时失效
- 双11期间失效修改为6小时，避免造成回源的峰值流量



ATS静态化使用后的收益和后续

- 使用静态化后的收益
 - 详情页（全站流量的30%+）静态化在双11期间的命中率达到95%，换言之，减少了后端服务接近30%的流量压力
 - RT从原来200ms降低到50ms，用户体验大大提升
 - 容量提升，减少了后端服务器的数量
- 后续
 - 借助云资源搭建云上的ATS，更贴近用户
 - ATS Cluster方案
 - 支持HTTPS
 - 回源流控和容灾控制

限流&降级开关推送和Web应急扩容方案

- 限流&降级开关
 - 限流，Web层，防止被流量打垮
 - 降级，App层（服务化），保障核心应用
- Web应急扩容方案
 - 选择Docker 容器，批量生成效率高
 - 启动速度快
 - 资源利用率提升明显



继续交流



Thanks!

