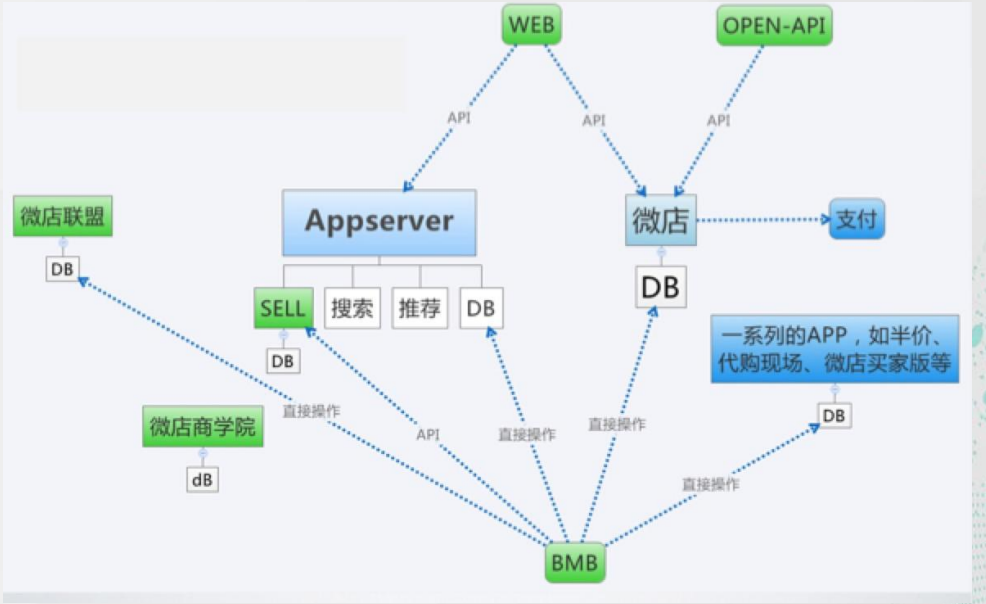
微店技术演进之路

陈国成

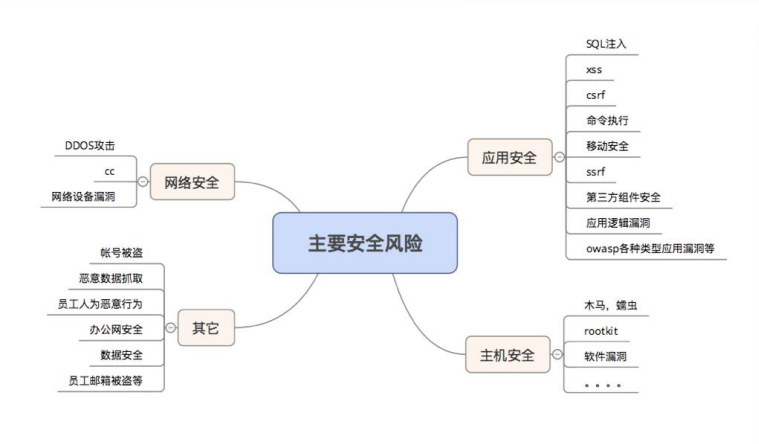


10 - 300人
海豹突击队

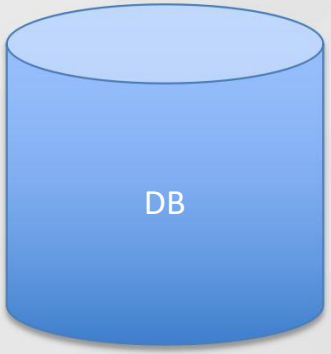
LAMP、F5、redis
技术特点

100w – 3000w
注册卖家数

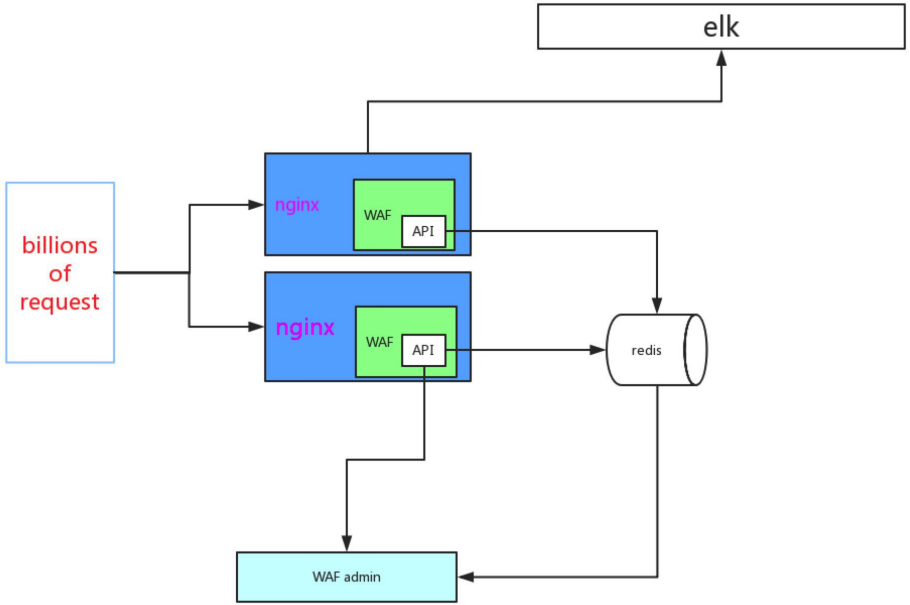
第一阶段，开天辟地

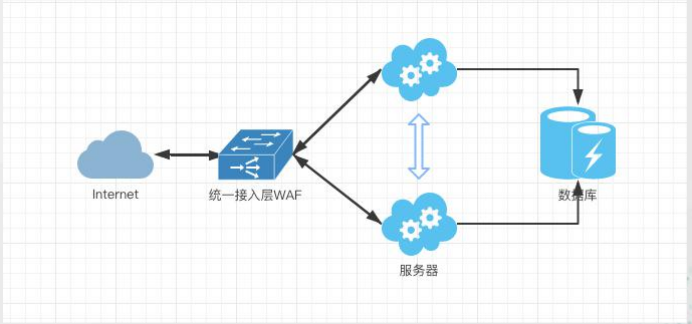


问题一、安全
日均受攻击 **560 W**次

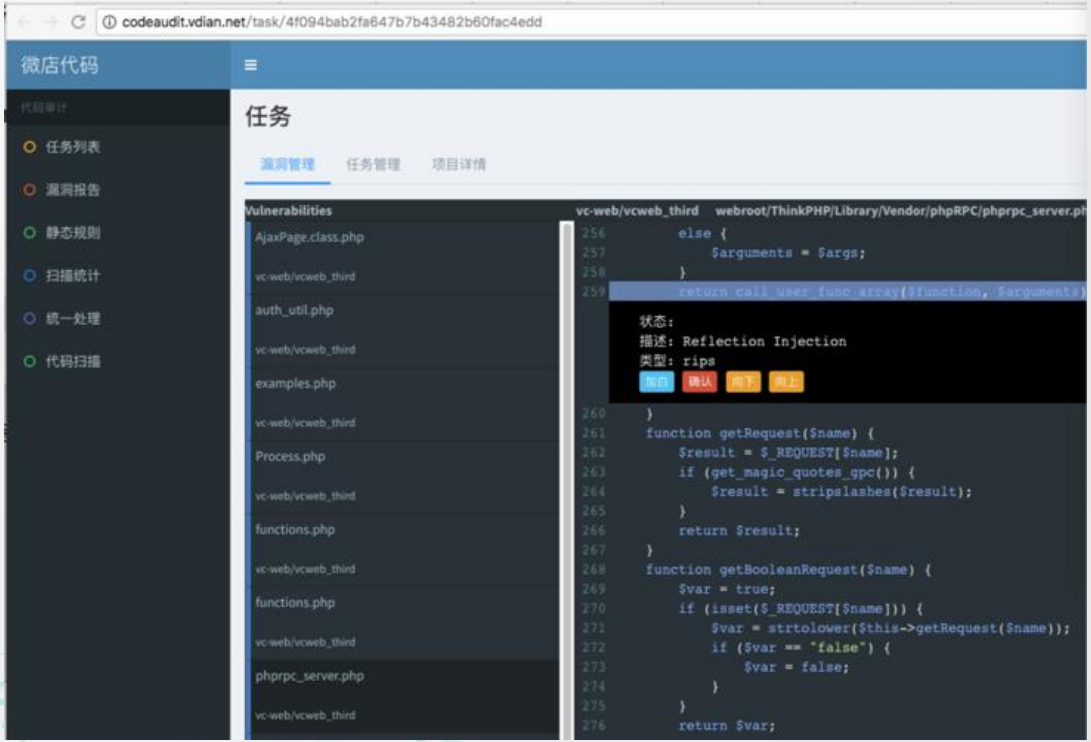


问题二、全站性能很差；稳定性问题频发
1E、单机单实例、性能/稳定性低下（**2**个**9**）、**SQL**治理差

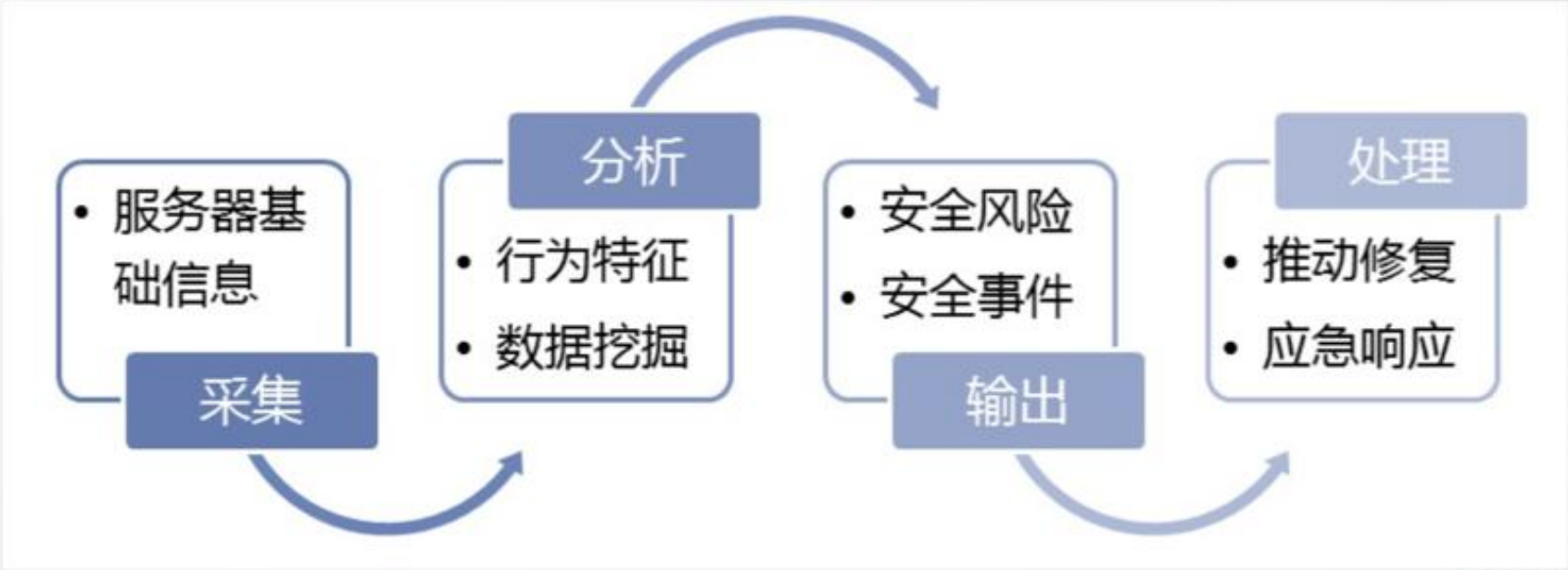




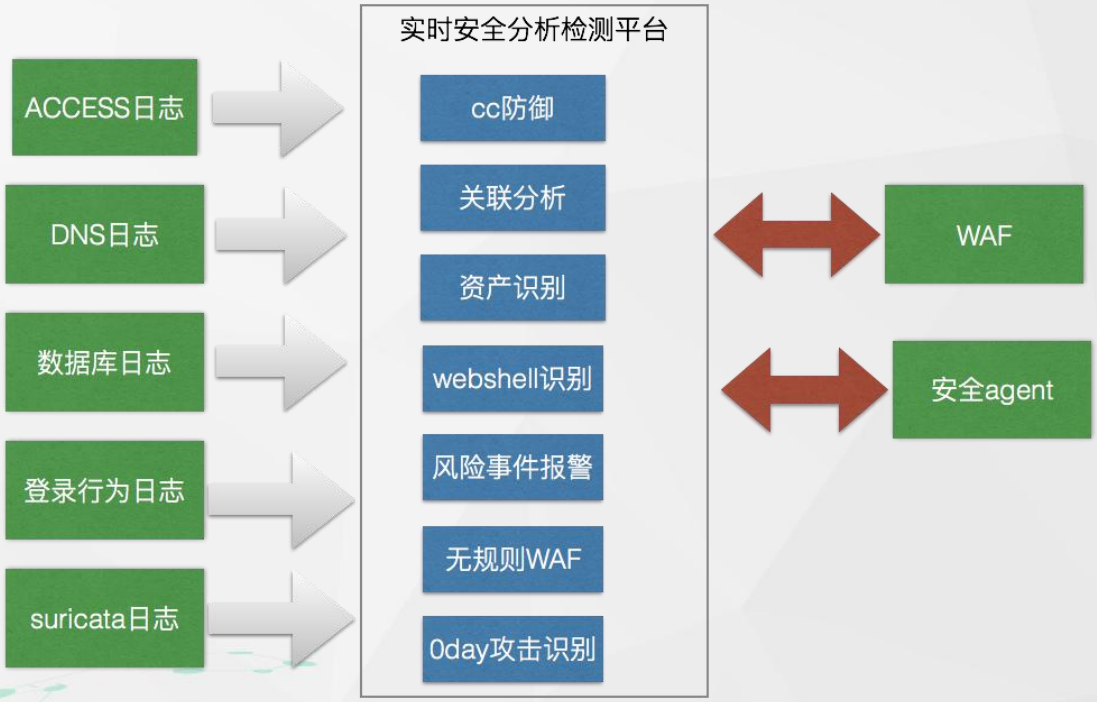
- 统一接入层（lua on nginx）运行。对业务无侵入，且100%全覆盖。
- 完善的漏洞防护，覆盖owasp主要漏洞类型。
- 实时规则防护，0day防护。
- 全天候实时阻断
- 虚拟补丁，规则动态实时更新
- 覆盖owasp Top10主要类型安全问题，完全规避防范SQL注入问题。提供cc防护、爬虫拦截、限流支持。10w级黑名单加黑支持。
- 与安全日志分析系统联动，实时的风险感知并拦截
- 0误伤



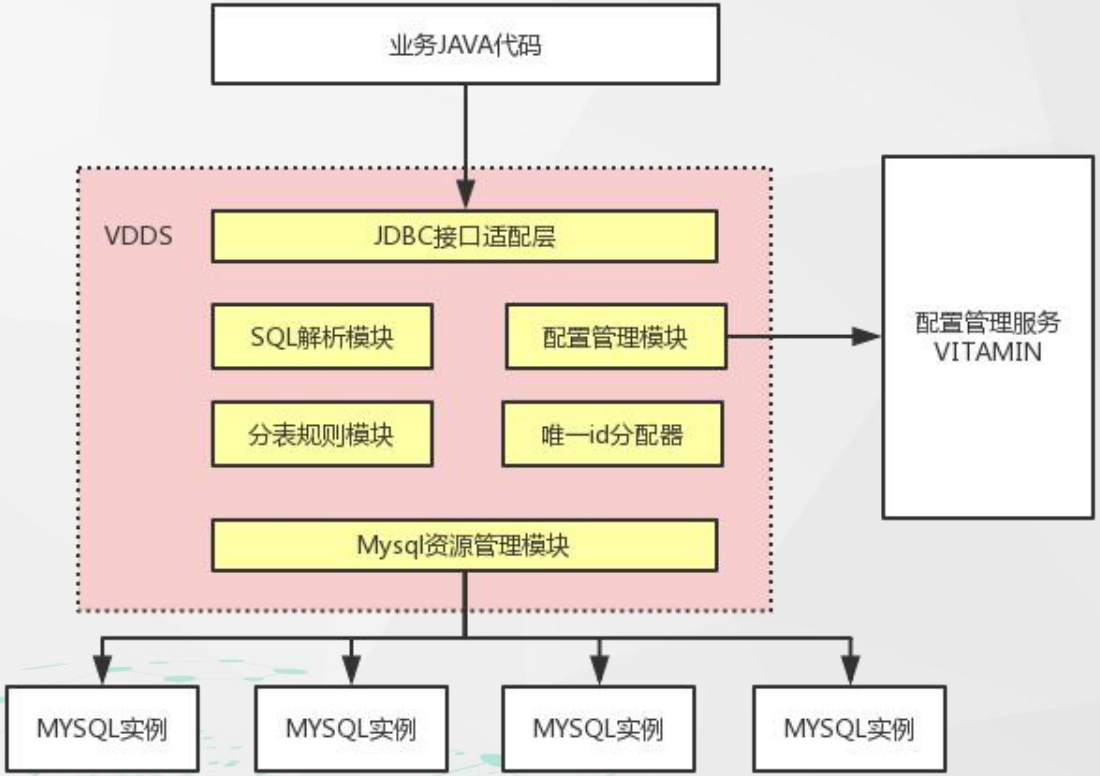
- 支持语言java、php、Node.js等语言
- 动态（语法分析）、静态（正则表达）以及集成第三方扫描引擎（rips、findsecbugs等）相结合的扫描模式
- 跟发布系统流程上结合覆盖所有发布上线项目，强制安全扫描



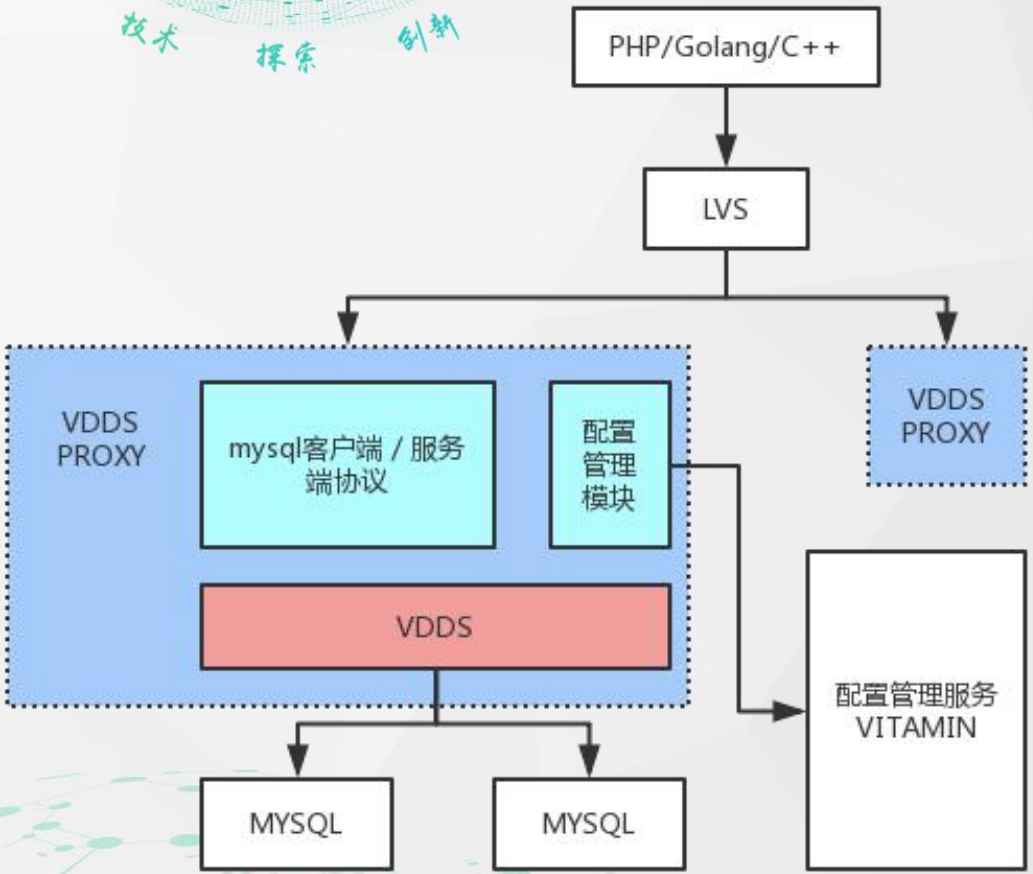
- HIDS 轻量级agent和云端组成，实现对文件监控、登录行为监控、命令审计、webshell查杀、网络监控拦截等功能。
- 解决主机层安全风险，并可对员工行为进行审计



- ✓ 建模：
  - ✓ 关键字、频率、webshell、httpheader等
  - ✓ 去噪
  - ✓ 菜刀请求特征
  - ✓ 基于统计的模型
- ✓ 机器学习攻击检测—无规则引擎实现（进行中）



- 分库分表：Client端实现分库分表功能；提供jdbc规范的接口，接入时只需替换数据源
- 读写分离 ：根据业务的sql确定使用主库还是从库执行sql语句，降低主库的压力；读请求按比例分配
- 独立的账号体系：接入业务使用的是vdds的账号，数据库用户名/密码加密且对业务隔离，保证了数据库账号的安全性
- 配置自动变更：配置改变之后，自动推送到所有机器上，不需要业务重启机器
- 灵活的hint机制：业务可以通过hint直接指定物理表，多运用于扫表的场景



- 平滑下线

✓ vdds proxy下线时，通过lvs心跳机制，不接受新db请求，同时等待老连接完成请求在关闭vdds proxy实例，降低业务端报错的概率

- 配置自动生效

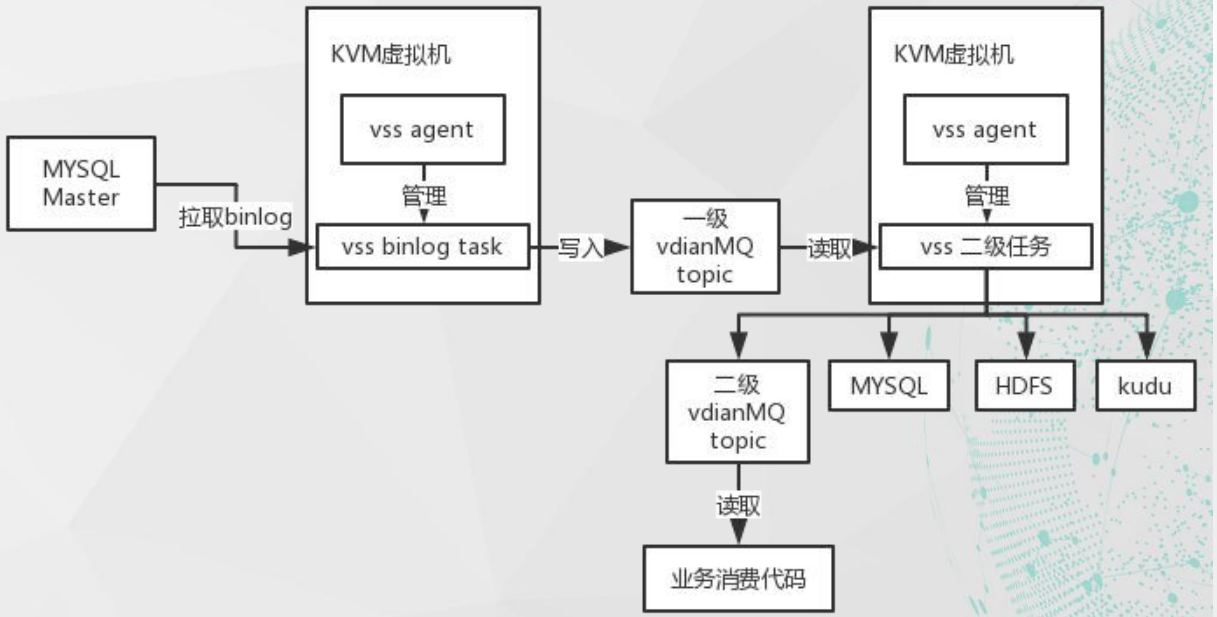
✓ 给vdds proxy增加逻辑库的时候，自动推送到所有vdds proxy实例，初始化对应的vdds配置

支持mysql preparedStatement协议

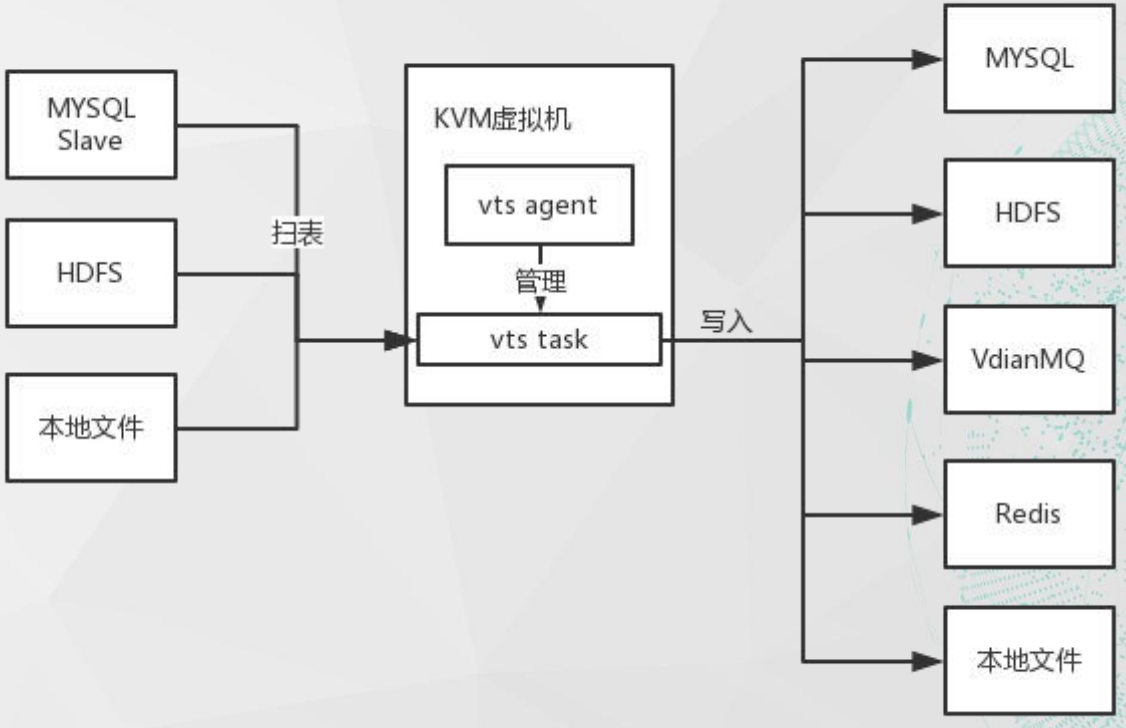
- 负载均衡

✓ 业务的db连接分配到多台vdds proxy，保证服务的稳定性

- vss（增量数据同步）
  - mysql，消息，hdfs，kudu的支持
  - 高可用
  - 灵活的过滤规则
  - 动态加载目标写入代码
  - 任务配置自动生成
  - 负载均衡



- vts（全量数据同步）
  - 存储，mysql，消息，hdfs，redis，本地文件支持
  - 高可用
  - 动态加载过滤规则
  - 数据拉取和写入的速度可控
  - 负载均衡



- 消息队列作为中间件的核心产品，在电商平台体系中扮演着极其重要的作用，包括异步系统解耦，流式数据处理，binlog同步等
  - ✓ 已有800多个topic
  - ✓ 每天有三千万多万的消息产生
  - ✓ 每天亿级消息消费
- 由于电商平台业务的复杂和多样性，导致对消息中间件有着特殊的功能需求及性能和可靠性要求：
  - ✓ 持久性，消息堆积，消息回溯
  - ✓ 支持严格顺序消息：binlog同步
  - ✓ 高吞吐，低延迟
  - ✓ 高可用，容灾

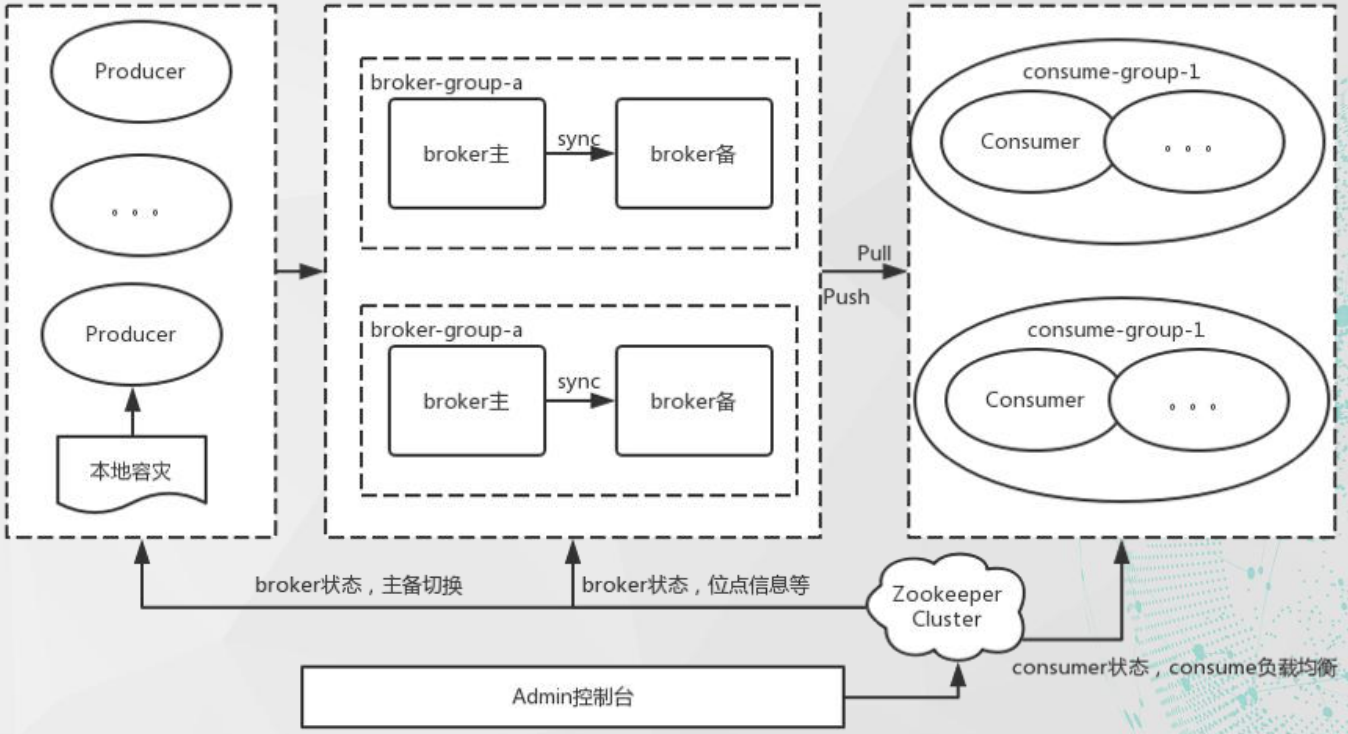
- 开源框架选型

RabbitMQ：使用erlang开发，出现问题难以把控，且顺序性消息及消息可靠性无法提供保障

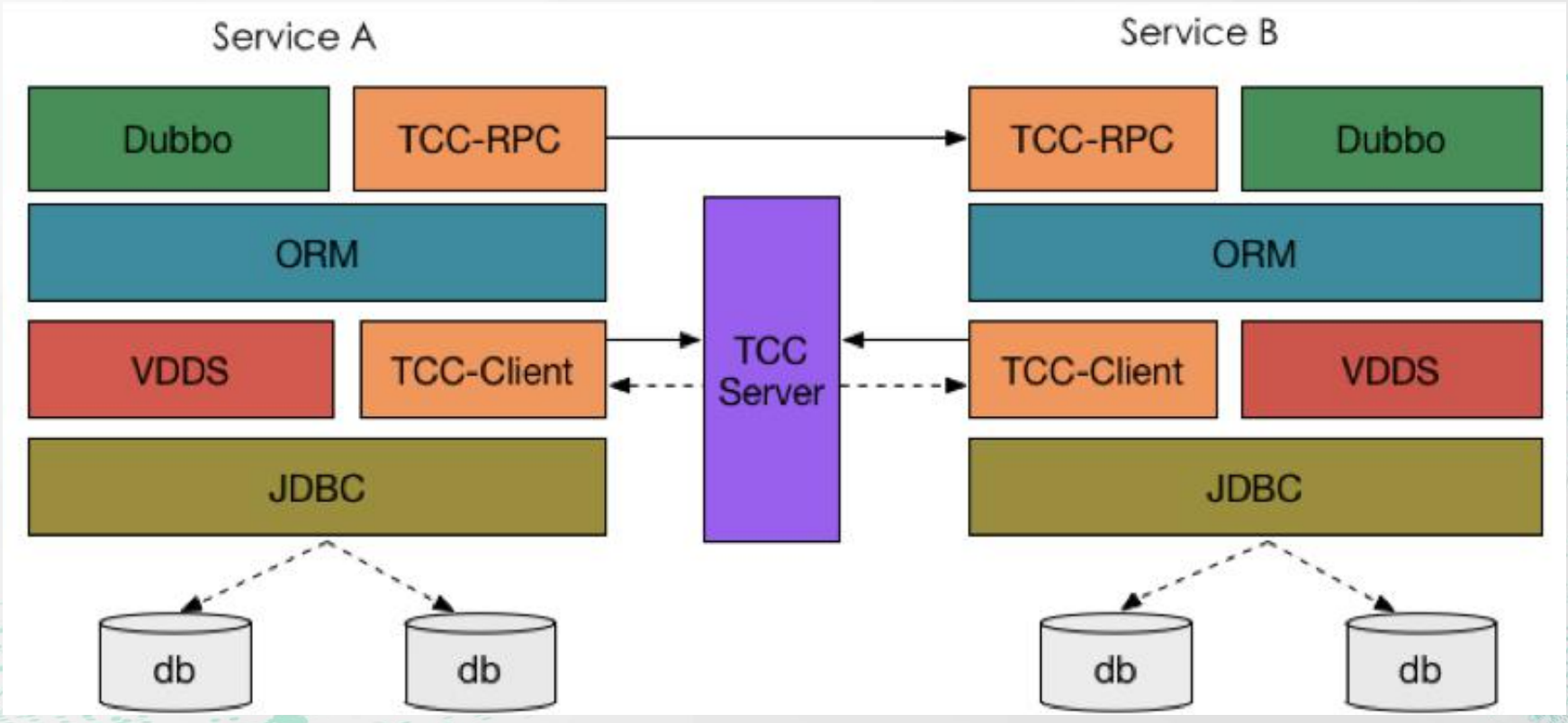
Kafka：分区越高load越高；多个分区导致随机写，影响整体吞吐

RocketMQ（开源版本）：不支持事务消息；缺少容灾支持；运维成本比较大

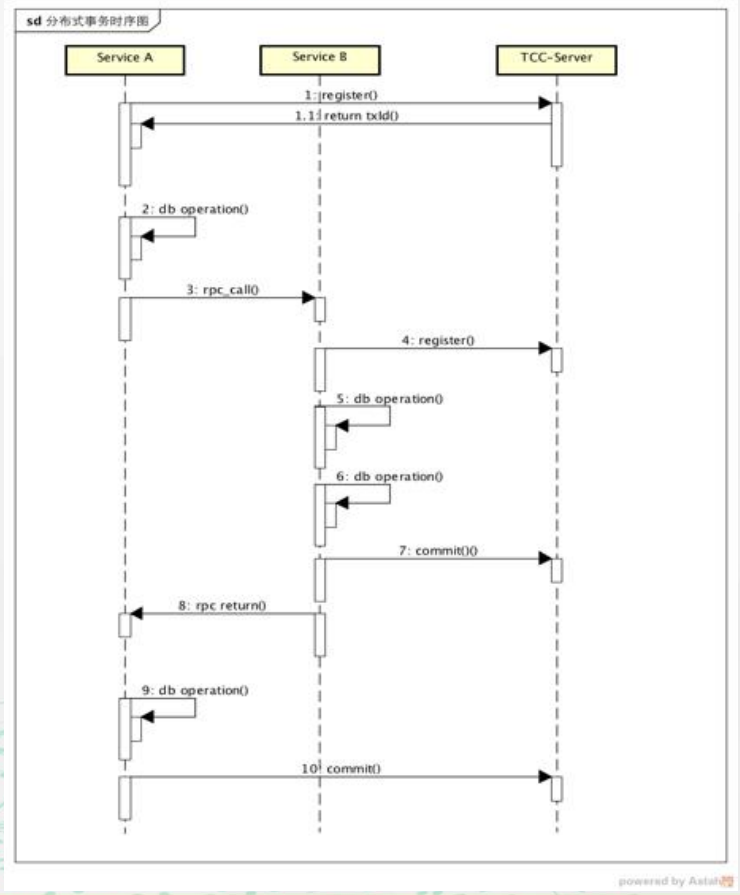
- 事务消息：
- 全链路消息轨迹：
- 高可用，容灾
- Admin运维
- 其他feature：
  - ✓按照tags过滤消息
  - ✓支持位点重置
  - ✓支持消费分组以及分组内的负载均衡



	方式	代码侵入性	数据库侵入性
Ebay 两阶段提交	同步；阻塞协议	弱	弱
基于消息	异步	较强	
Alipay XTS	Try同步； confirm/cancel异步	强	主事务分支事务记录
Taobao TXC	同步	弱	Log表业务同库



中间件 ， tcc（分布式事务框架）



- TCC处理流程

每个service需要register和commit;
每个业务db写操作,附加日志表写作为本地事务提交
TCC-Server负责事务注册、回滚;必须保证性能和稳定性
每接入一个应用,多两次RPC调用，开销为15ms左右
支持应用级别事务动态降级

- Tcc特性

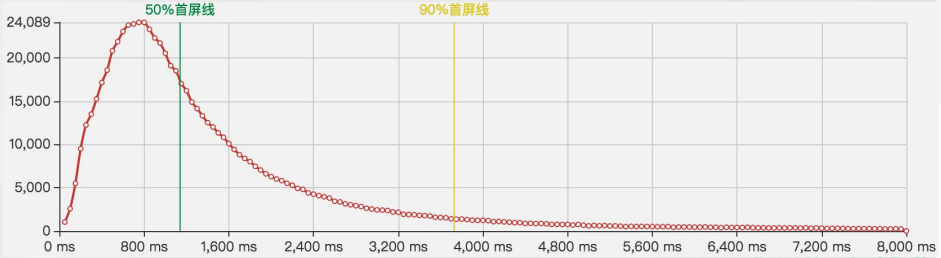
支持Local和Remote两种模式
Local模式收敛在单一系统中，解决跨库事务
Remote模式支持跨系统之间RPC调用的分布式事务

数据层封装遵循JDBC规范，透明代理VDDS，方便后续无缝同步升级

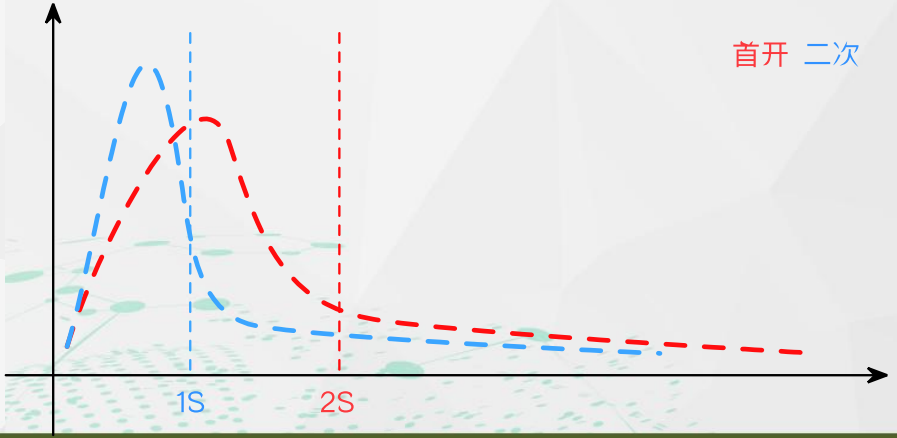
支持Hint方式指定回滚逻辑和自定义回滚方法
默认不需要显示回写回滚参数，基本无侵入
支持显示调用回写特殊参数

➢ 数据评估参考：

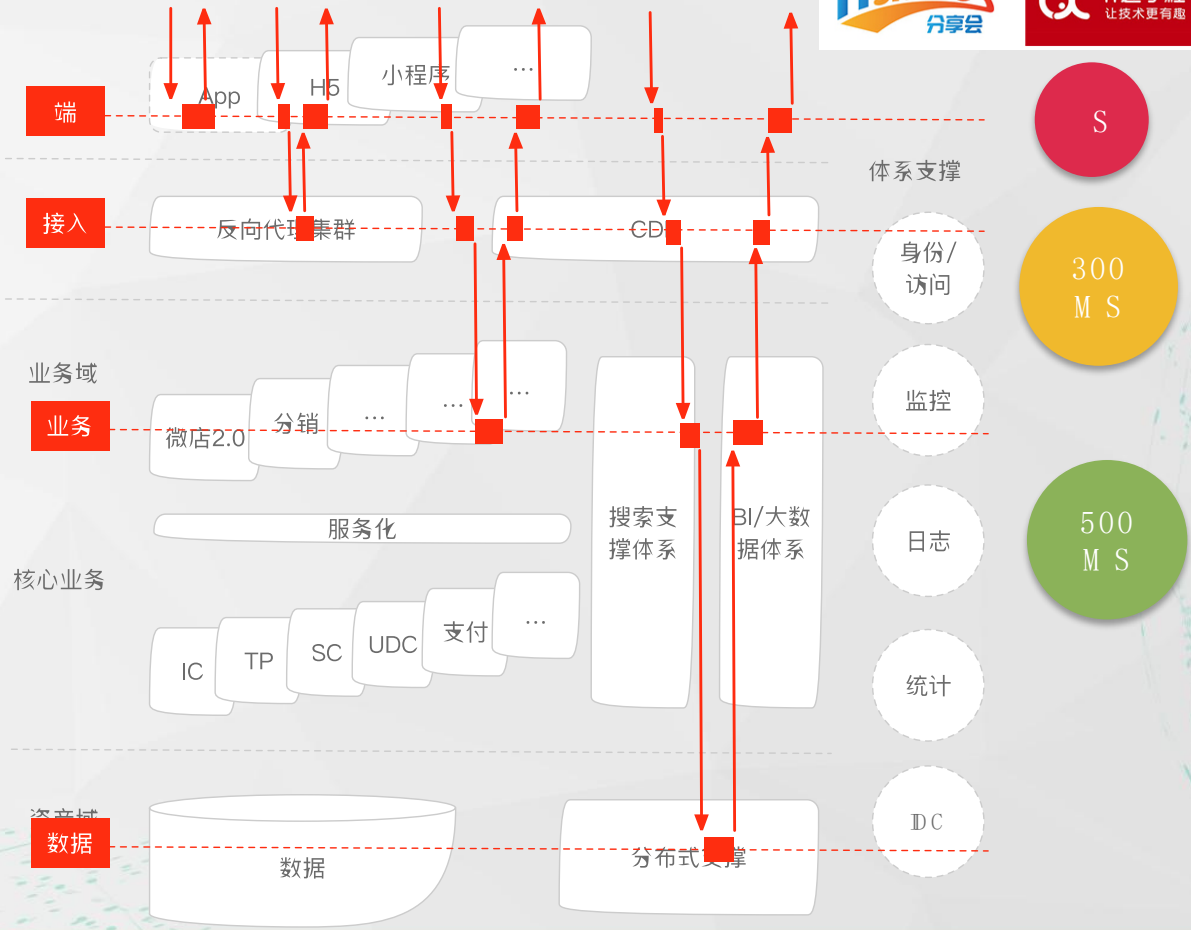
页面加载速度每提高1秒，转化率增加2%...反之，如果超过4秒，25%的用户会选择离开...
用户最满意的打开网页时间是2-5秒，如果等待超过10秒，99%的用户会关闭这个网页

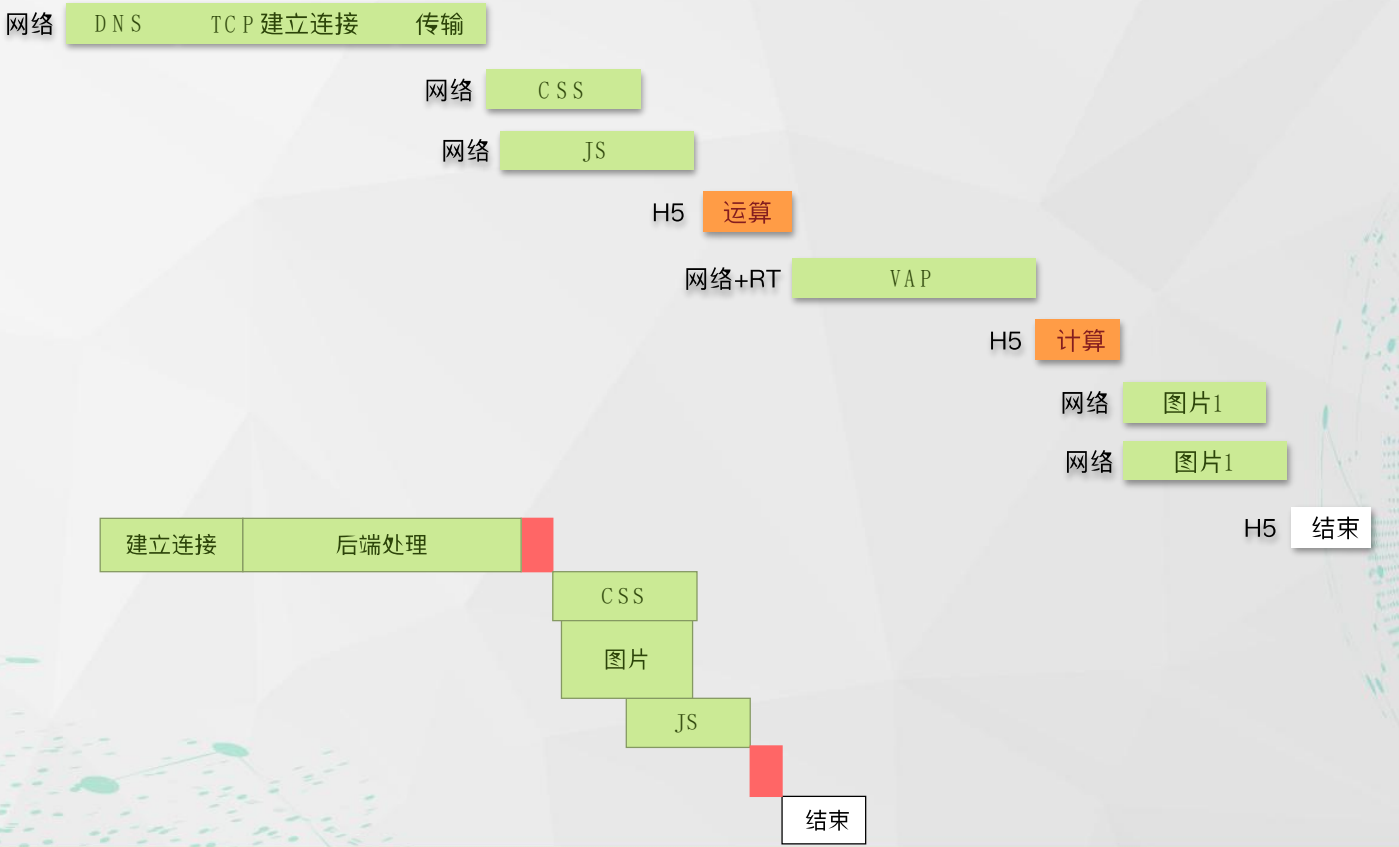


Google：网站访问速度每慢400ms就导致用户搜索请求下降0.59%
Amazon：每增加100ms网站延迟将导致收入下降1%
雅虎：如果有400ms延迟会导致流量下降5-9%



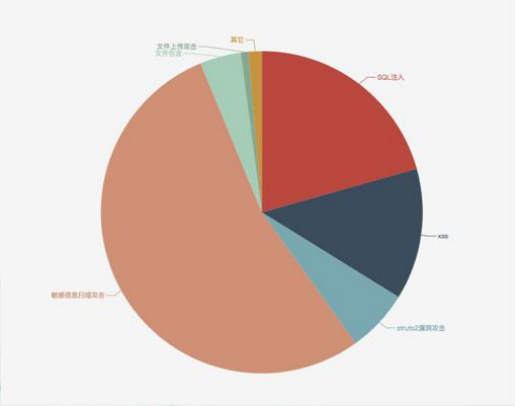
秒开

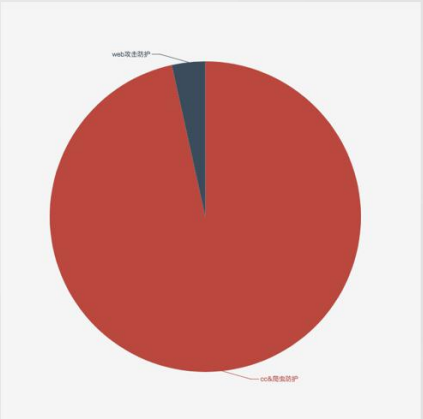

端
App
H5
小程序
…
接入
反向代理集群
CDN
体系支撑
身份/访问
监控
日志
统计
DC
业务域
业务
微店2.0
分销
…
…
…
服务化
搜索支撑体系
BI/大数据体系
核心业务
IC
TP
SC
UDC
支付
…
数据
数据
分布式支撑
S
300 M S
500 M S


网络
DNS
TCP建立连接
传输
网络
CSS
网络
JS
H5
运算
网络+RT
VAP
H5
计算
网络
图片1
网络
图片1
H5
结束
建立连接
后端处理
CSS
图片
JS
结束

性能top5(昨日)

排名	url（点击可查看业务性能）	性能得分	首屏DOMContent Load	首屏Render	首屏Load	秒开率	spmb
1	https://h5.weidian.com/m/secbuyseller/index.html	99.54	56	255	311	99.6	nu69g4r2
2	https://h5.weidian.com/m/fxtraining/task_msg.html	97.15	22	551	573	93.35	40nb3sow
3	https://h5.weidian.com/m/fxtraining/task.html	96.61	66	558	623	91.77	2njpqfem
4	https://fxh5.weidian.com/retailer/manage/apply.html	93.87	26	778	804	84.85	s3mpygq1
5	https://fxh5.weidian.com/retailer/manage/supplier_list.html	92.41	139	754	893	80.8	veqw3xr5



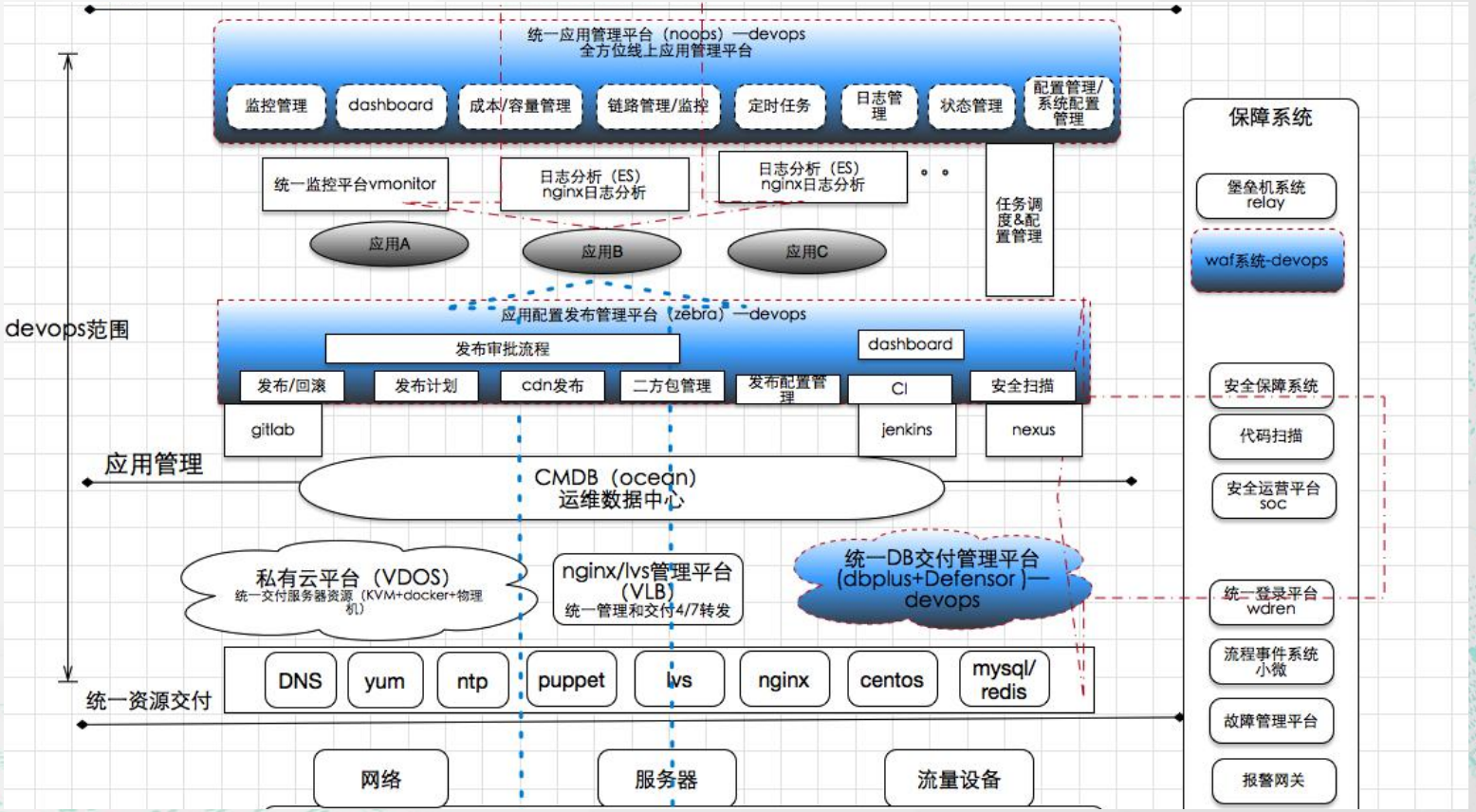


5000w
IDC成本

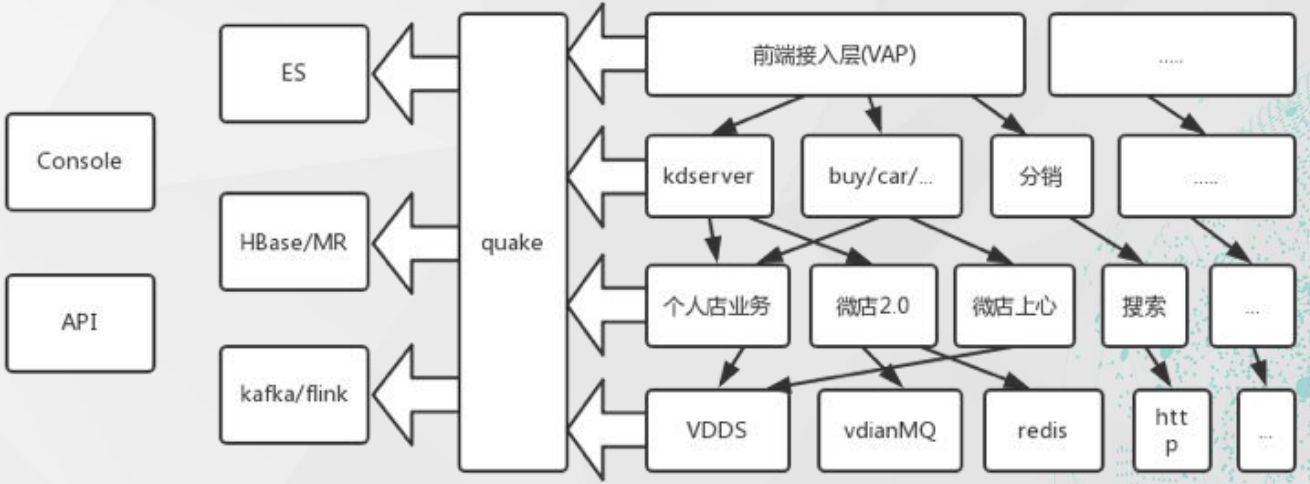
70%，< 15%
服务器利用率

600+
技术人员

全站分布式，语言收敛



运维体系，以应用为中心

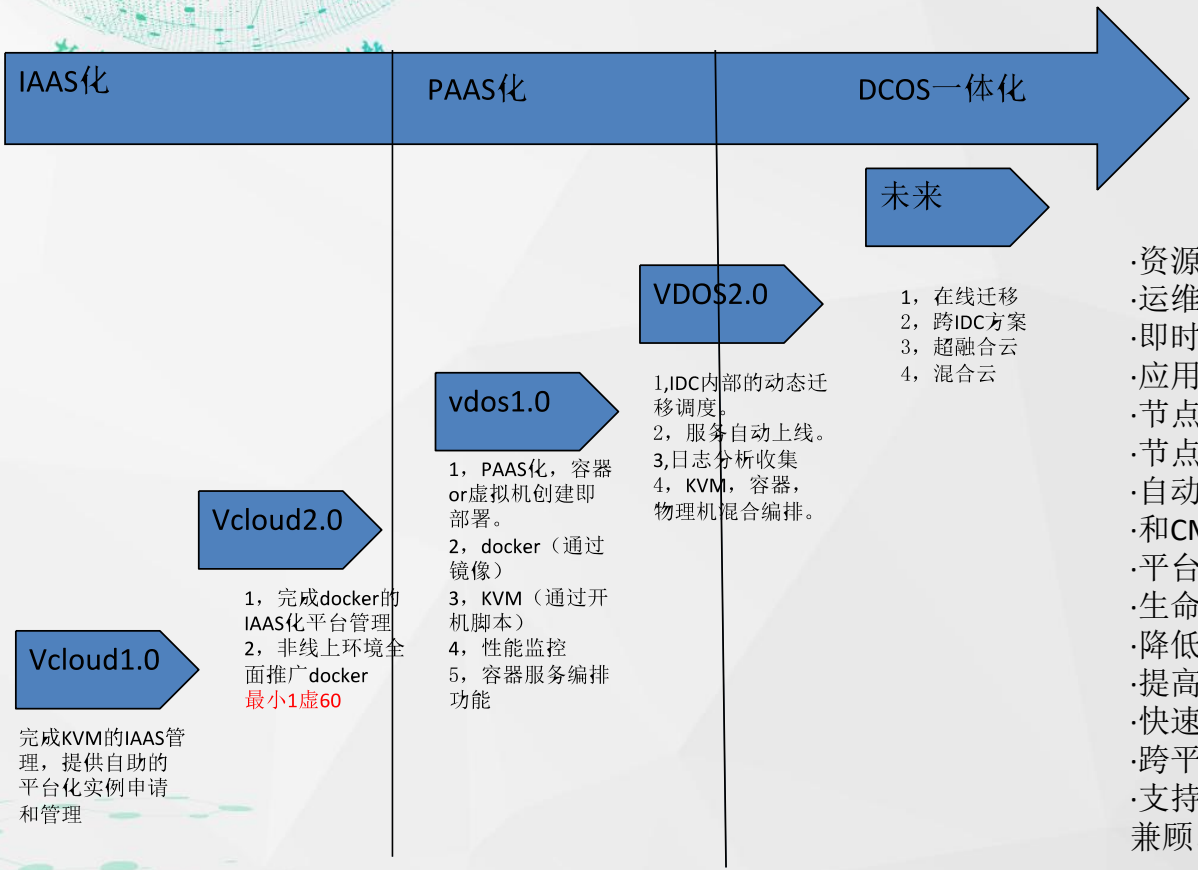


TraceId由最前端java应用生成，会放到tomcat中
Rpc链路的串接使用0.1.4.1.5的方式，异步情况做了特殊处理
所有远程调用框架的日志格式一致，存储位置一致，通过不同类型来区分

TraceId分四部分16B
Seq: short类型2B 1～8000循环递增
时间戳：long类型8B 当前时间
机器IP：int类型4B 当前机器ip，转成int
进程号：short类型2B 当前进程号，避免统一机器起多个应用冲突,同时对大于short的做截取处理（进程id是int的）

Rpc ID	AppName	RemoteIp	Trace Type	Status	Trace Name	Size	Time Line	Message
0	▼ vap-server(10.2.131.202)	172.19.35.116	HTTP	OK	/h5/ares/item.getRecommendItems/1.0	1	293ms	200
0.1	▼ ares(10.2.101.81)	10.2.101.81	DUBBO	OK	com.vdian.vap.common.aresService.item.getRecommer	15982	292ms	OK
0.1.1	▼ pluto(10.2.114.91)	10.2.114.91	DUBBO	OK	com.vdian.pluto.client.service.RecEngineService.execut	13896	230ms	OK
0.1.1.1	▶ mercury(10.2.129.88	10.2.129.88	DUBBO	OK	com.vdian.mercury.service.MercuryService.service:1.0	848	5ms	OK
0.1.1.2	▶ mercury(10.2.129.88	10.2.129.88	DUBBO	OK	com.vdian.mercury.service.MercuryService.service:1.0	3000	14ms	OK
0.1.1.3	▶ mercury(10.2.129.88	10.2.129.88	DUBBO	OK	com.vdian.mercury.service.MercuryService.service:1.0	861	14ms	OK
0.1.1.4	▶ mercury(10.2.129.88	10.2.129.88	DUBBO	OK	com.vdian.mercury.service.MercuryService.service:1.0	861	13ms	OK
0.1.1.5	▶ mercury(10.2.129.88	10.2.129.88	DUBBO	OK	com.vdian.mercury.service.MercuryService.service:1.0	26594	20ms	OK
0.1.1.6	▼ fenxiao-core(10.2.12	10.2.129.168	DUBBO	OK	com.koudai.fenxiao.client.service.FxItemService.queryIt	763	144ms	OK
0.1.1.6.1	fenxiao-core(10.2.	FX_MAP_APP	VDDS	OK	SINGLE	0	2ms	SELECT,fx_item_info,10.2.117.
0.1.1.6.2	fenxiao-core(10.2.		REDIS	OK	supplyItemSetting_mGet	0	1ms	ok
0.1.1.6.3	fenxiao-core(10.2.		REDIS	OK	fenxiao-core1_get	0	0ms	ok
0.1.1.6.4	fenxiao-core(10.2.		REDIS	OK	supplyItemSetting_mGet	0	0ms	ok
0.1.1.6.5	fenxiao-core(10.2.		REDIS	OK	fenxiao-core1_get	0	0ms	ok
0.1.1.6.6	fenxiao-core(10.2.		REDIS	OK	supplyItemSetting_mGet	0	0ms	ok
0.1.1.6.7	fenxiao-core(10.2.		REDIS	OK	fenxiao-core1_get	0	0ms	ok
0.1.1.6.8	fenxiao-core(10.2.		REDIS	OK	supplyItemSetting_mGet	0	0ms	ok
0.1.1.6.9	fenxiao-core(10.2.		REDIS	OK	fenxiao-core1_get	0	0ms	ok
0.1.1.6.10	fenxiao-core(10.2.		REDIS	OK	supplyItemSetting_mGet	0	0ms	ok
0.1.1.6.11	fenxiao-core(10.2.		REDIS	OK	fenxiao-core1_get	0	0ms	ok
0.1.1.6.12	fenxiao-core(10.2.		REDIS	OK	supplyItemSetting_mGet	0	1ms	ok
0.1.1.6.13	fenxiao-core(10.2.		REDIS	OK	fenxiao-core1_get	0	0ms	ok
0.1.1.6.14	fenxiao-core(10.2.		REDIS	OK	supplyItemSetting_mGet	0	0ms	ok
0.1.1.6.15	fenxiao-core(10.2.		REDIS	OK	fenxiao-core1_get	0	0ms	ok
0.1.1.6.16	▼ fenxiao-core(10.2.11	0.2.101.92	DUBBO	OK	com.vdian.vmp.client.service.detail.DetailPromotionQue	1016	56ms	OK
0.1.1.6.16.2	▼ vmpcoupon(10.2-		DUBBO	OK	com.koudai.vmp.service.SellerShopCouponReadService	443	3ms	OK
0.1.1.6.16.2.1	vmpcoupon(10-		VDDS	OK	SINGLE	0	1ms	SELECT,vmp_shop_coupon,10

服务治理， vtrace（分布式链路跟踪）



·资源快速交付
·运维能力的一个重要指标
·即时交付，秒级交付
·应用平滑上下线PAAS化
·节点创建后，应用可以自动完成上线
·节点删除后，应该可以自动下线
·自动化平台化
·和CMDB，发布系统等运维系统自动化对接
·平台化管理
·生命周期全部自动化
·降低TCO
·提高虚拟化密度
·快速上下线
·跨平台
·支持libvirt虚拟化，支持docker；稳定性和灵活性兼顾。


支撑系统
VLB(4层7层管理平台)
SOA平台
APP
APP
APP
APP
APP
APP
APP
API
应用节点
资源调度模块
应用管理
统一API
健康检查 and LB管理模块
dashboard
资源配置模块
迁移模块
资源集群管理
权限管理
KVM IAAS管理
docker IAAS管理
镜像统一管理
日志查询
监控/性能
管理调度平台
镜像存储
Ceph集群
ES集群
日志
监控/性能
Graphite集群
监控 and 日志
graphite
logstash
平台数据交互
RPC
rabbitmq
websocket
通信层
Vdos-agent
容器资源节点
Vdos-agent
容器资源节点
Vdos-agent
容器资源节点
Vdos-agent
容器资源节点
IDC资源

开发自助管理自己的应用，解放PE/OP的工作量，加强开发参与，提升应用稳定性。
PE/OP向SRE转型

核心产品：
监控系统
容量水位
线上job管理；job互编排
日志采集、中转、分析
链路依赖治理
完善的dashboard，数据分析
超级agent
交付

现在

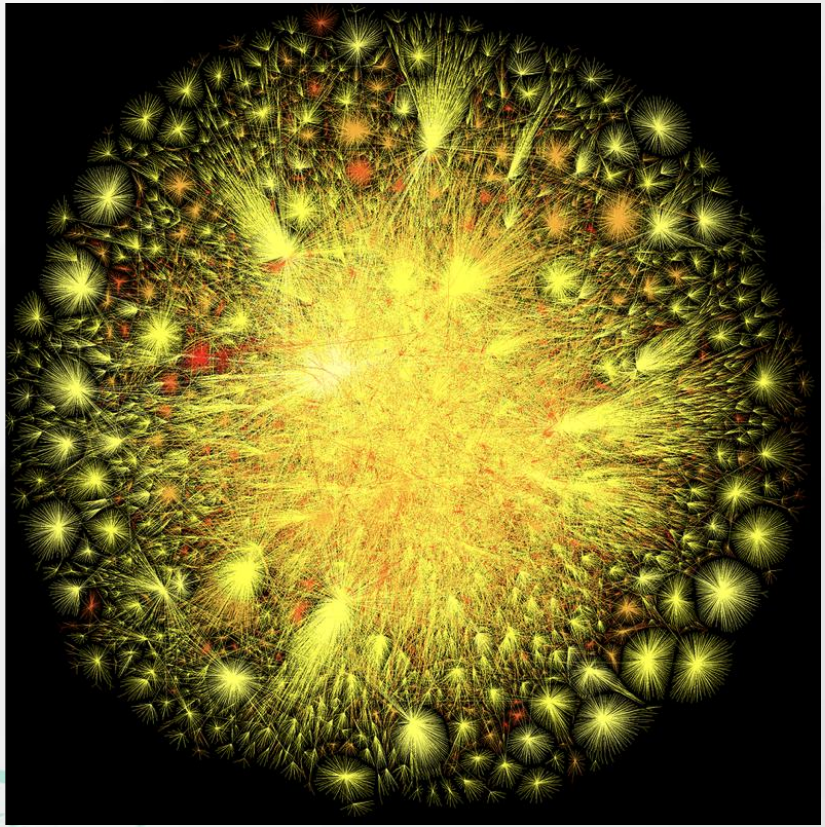
交付自助	监控自助	日4 操作自助	服务优化自助	AIops
1，自助CD 2，自助回滚 ，发3 计划管理 4，需求jira管理	• 日志监控 • 链路监控 • 系统监控 • 监控治理 • 监控报表	• 清理日志 • 集群grep • 集群cron • Ssh通道管理 • 等等job	• 水位2量 • 日志分析 • 日志收集	• 故障自恢复 • 故障预测 • 故障分析定位 • 2 量AI

43s
平均每次构建

574，324，400+
后端应用数量，前端应用数量，每天发布

104s
平均每次部署

18，9，6
平台建设前后的OP数量变化



第四阶段，海量数据下的搜索与推荐

技术挑战

数据行数大
实时性、一致性要求高
TPS/QPS在1-2k左右

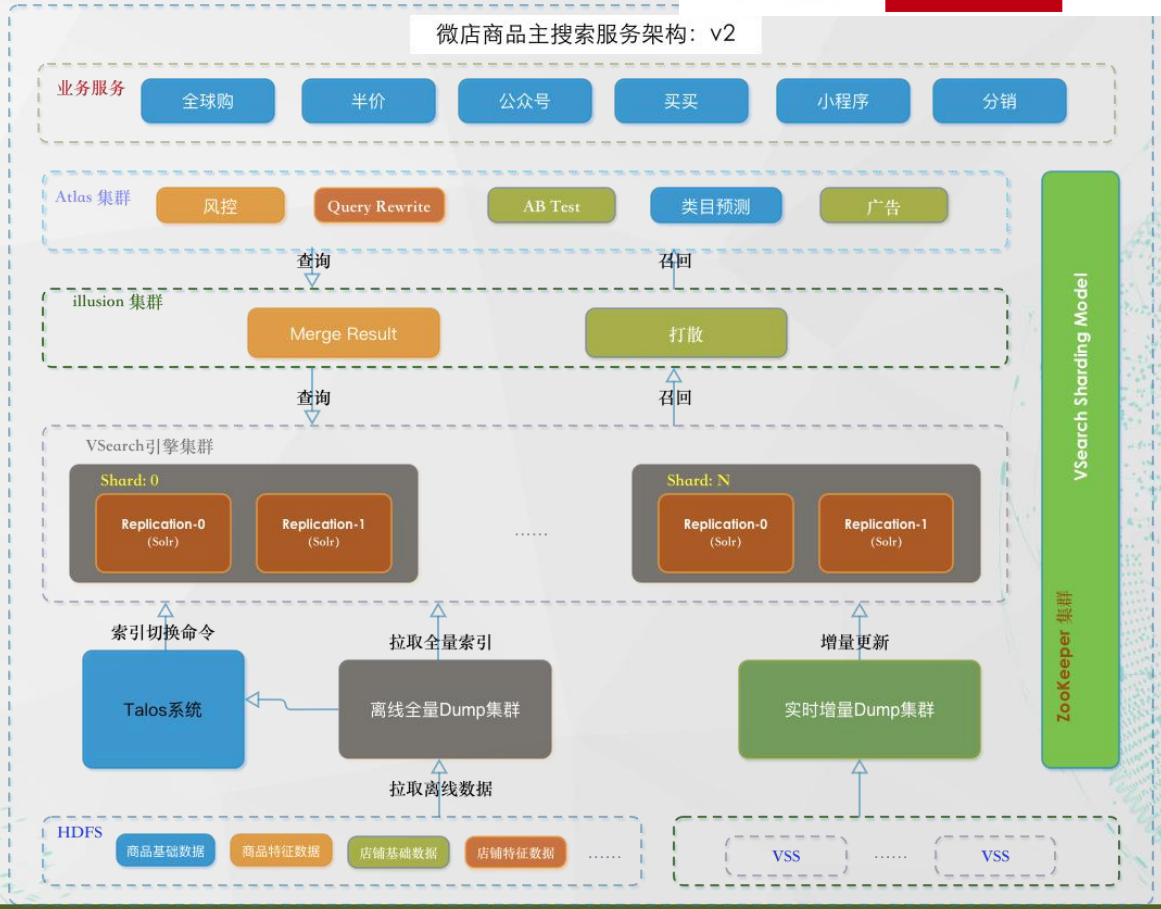
业务挑战

排序逻辑复杂

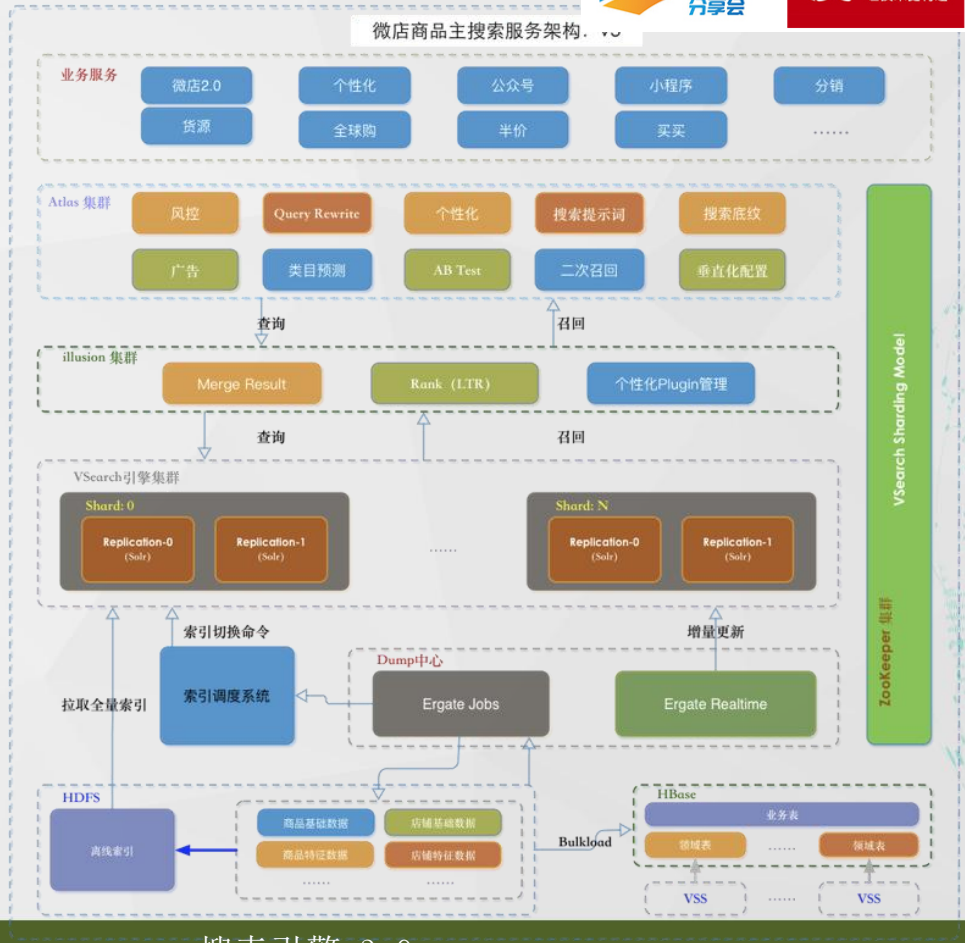
1.0架构

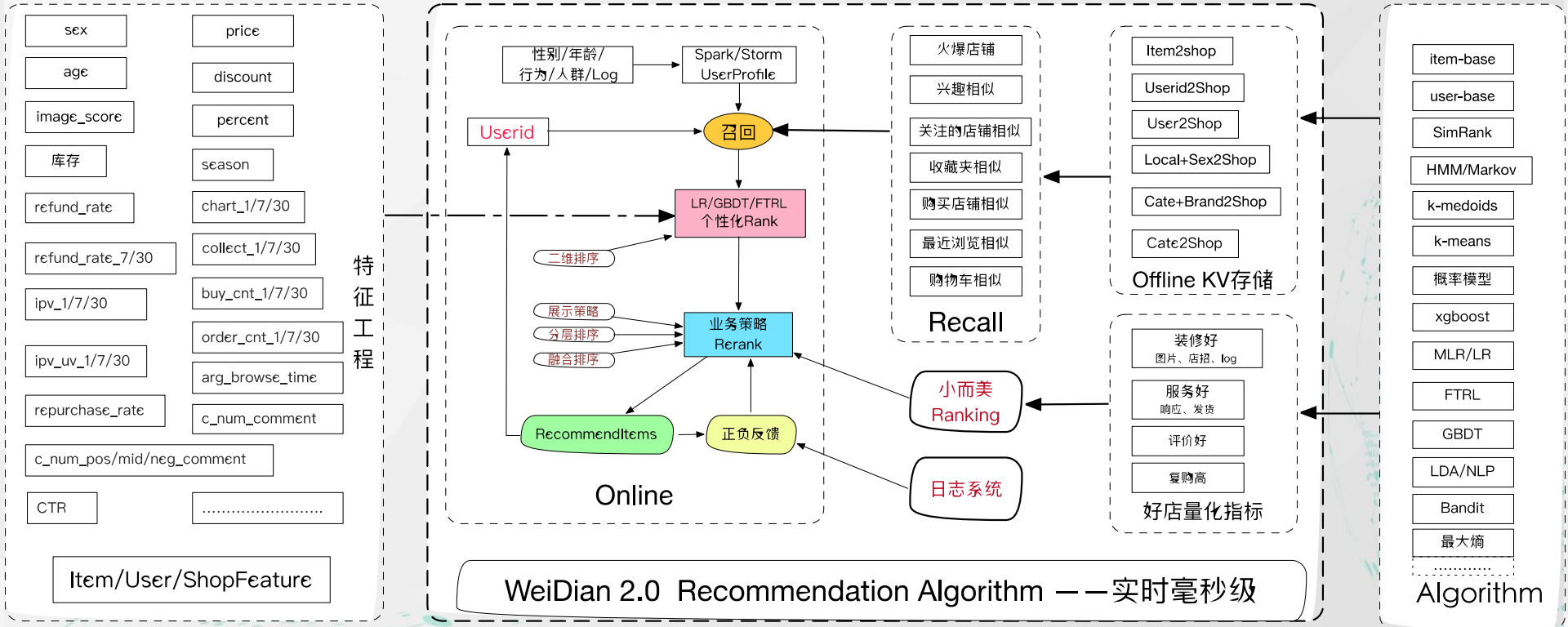
- 引擎基于solr，主从复制框架，一主，N个从
- mapreduce join 数据，批量直接更新主 solr 索引中。分钟级增量
- 排序模型简单使用少量几个销量数据加权计算。solr 函数排序

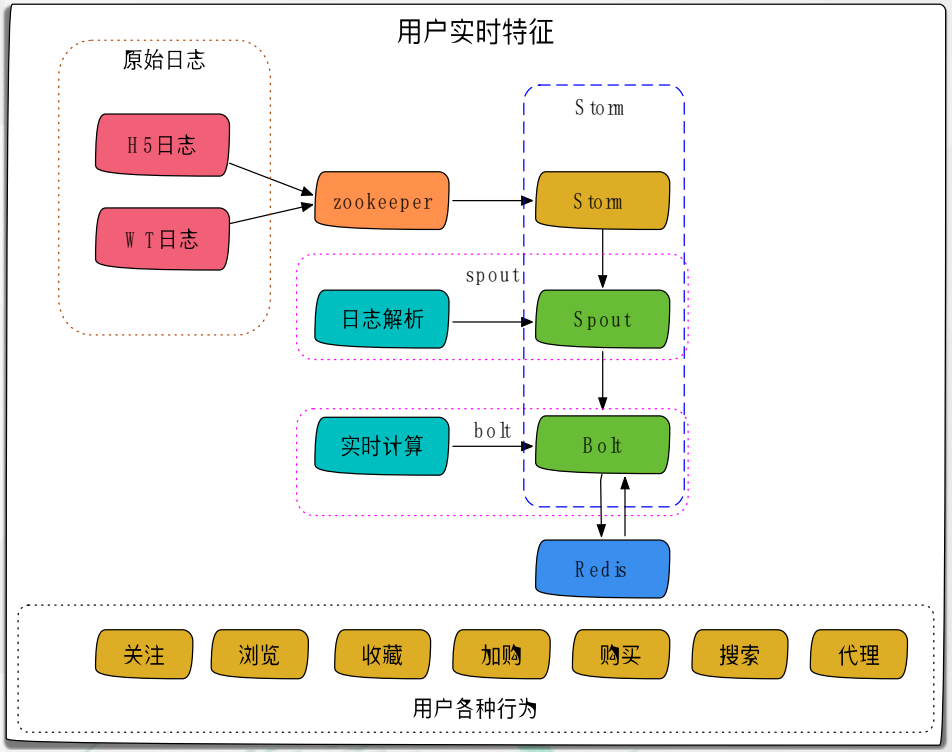
- 封装solr，构建分布式架构
- 引入merge层（ illusion），采用海选+精排架构
- 引入统一接口层（atlas）。QP实现；ABT
- 索引构建，全量 hadoop join + 实时增量；形成dump服务
- 粗糙的引擎管理系统：屏蔽降权等
- Rank。海选+精排，引入LTR
- 问题，仍然还有。。。

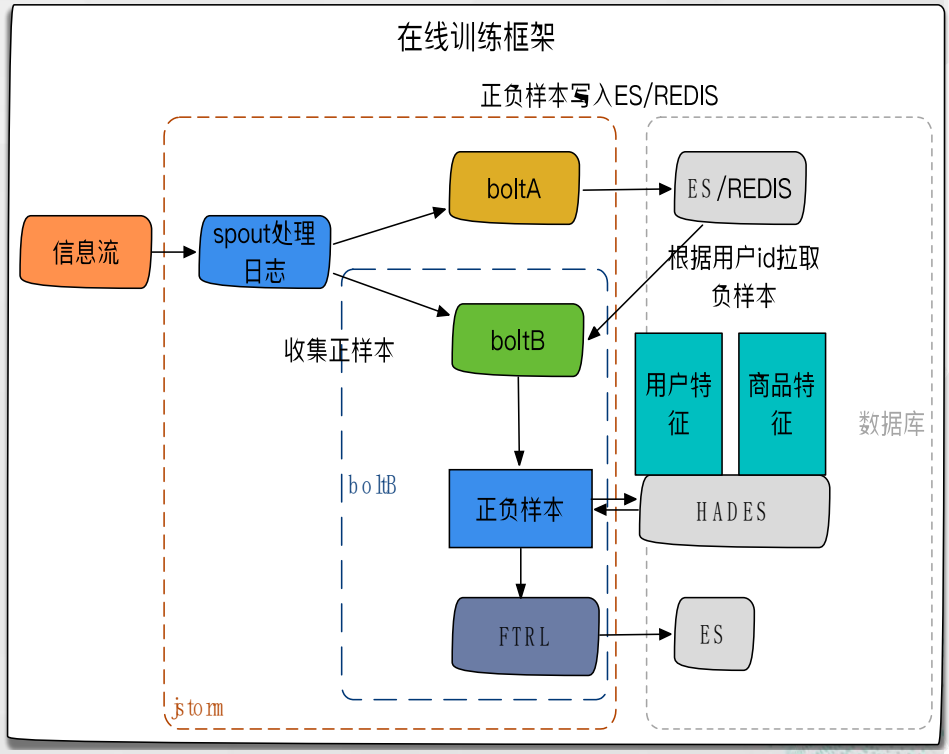


- 引擎能力升级，支持算法个性化能力，插件化支持；同时Rank在线学习，模型更新更实时。
- 实现实时引擎，时效性 < 500ms
- 索引管理引入调度器，引入全量索引切换调度。（2.0 按顺序切换）
- 索引碎片整理，减小磁盘空间，提高查询性能。
- Dump 服务升级，hbase 成为 dump 的基础。
- 不依赖业务的版本号设计（hbase 行原子），简单、稳定、一致。同时稳定、可靠的版本号为系统做幂等提供支持
- 实时对账/补偿









ITshare 分享会　IT趣学社 让技术更有趣　IT大咖说 知识分享平台

CTDC
首席技术官领袖峰会

后会有期

2018.9.8

+ 乌镇再聚 +

更高规格、更优质的服务，只为更好的遇见你！