



研发创新顶级峰会

TiD 2018

# 下一代软件研发 构建智能生态

TiD2018质量竞争力大会

• 大规模分布式系统性能测试实践分享 •

华为 周英伟

## 自我介绍

周英伟 华为CloudBU

12年测试经验，当前主要从事云性能测试服务架构与规划工作

华为云应用服务领域测试架构师

华为云性能测试服务架构师

系统容量相比传统应用数量级增长

微服务化架构，调用关系更加复杂

用户增长迅速，资源突发需求量大

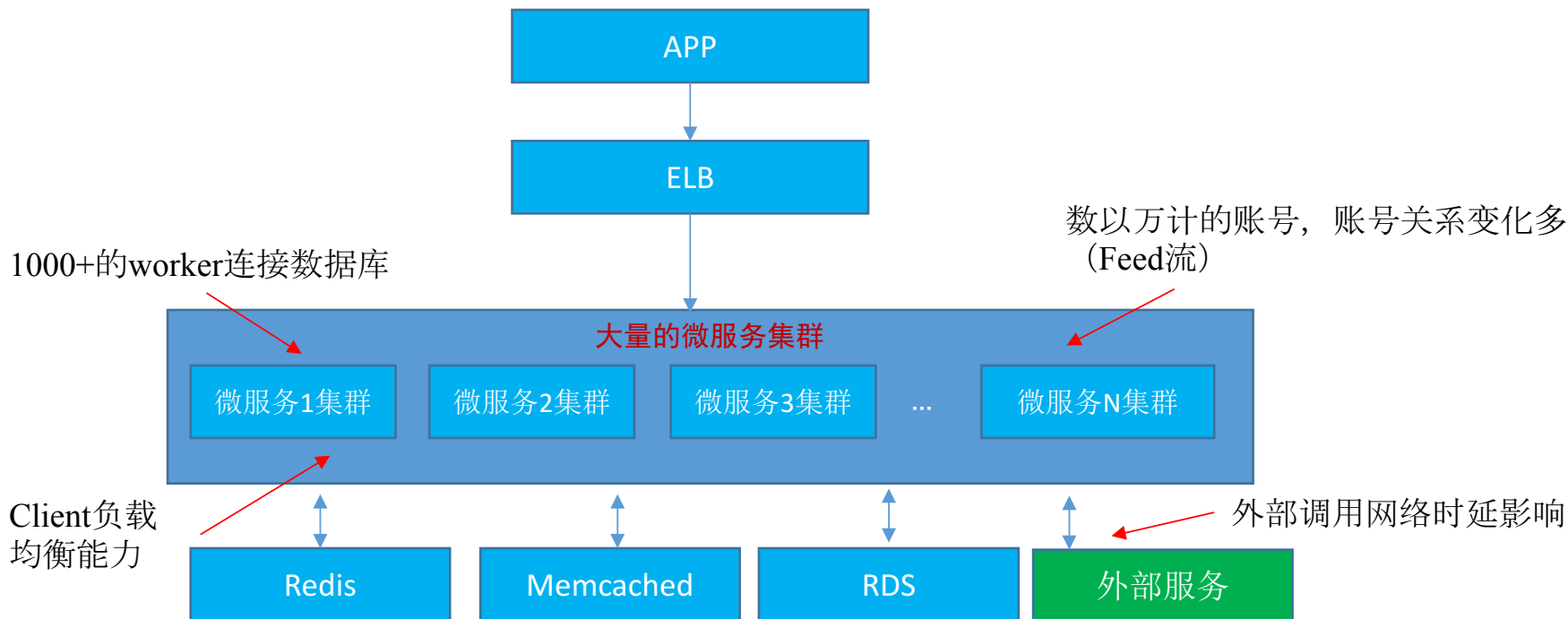
- 传统性能测试工具性能不足，自研技术门槛高
- 瓶颈在各微服务间漂移，测试技术难度大
- 如何摸清资源扩容模型，有限资源下如何验证性能
- 一旦性能问题流入现网，问题定位周期长

## 如何更加系统的开展性能测试活动

- 被测对象分析
- 测试场景分析建模
- 测试需求分析
- 工具选型与搭建
- 测试执行
- 性能测试分析与调优

# 被测对象分析（某社交类APP）

从系统架构分析可能出现的瓶颈点，作为重点测试场景

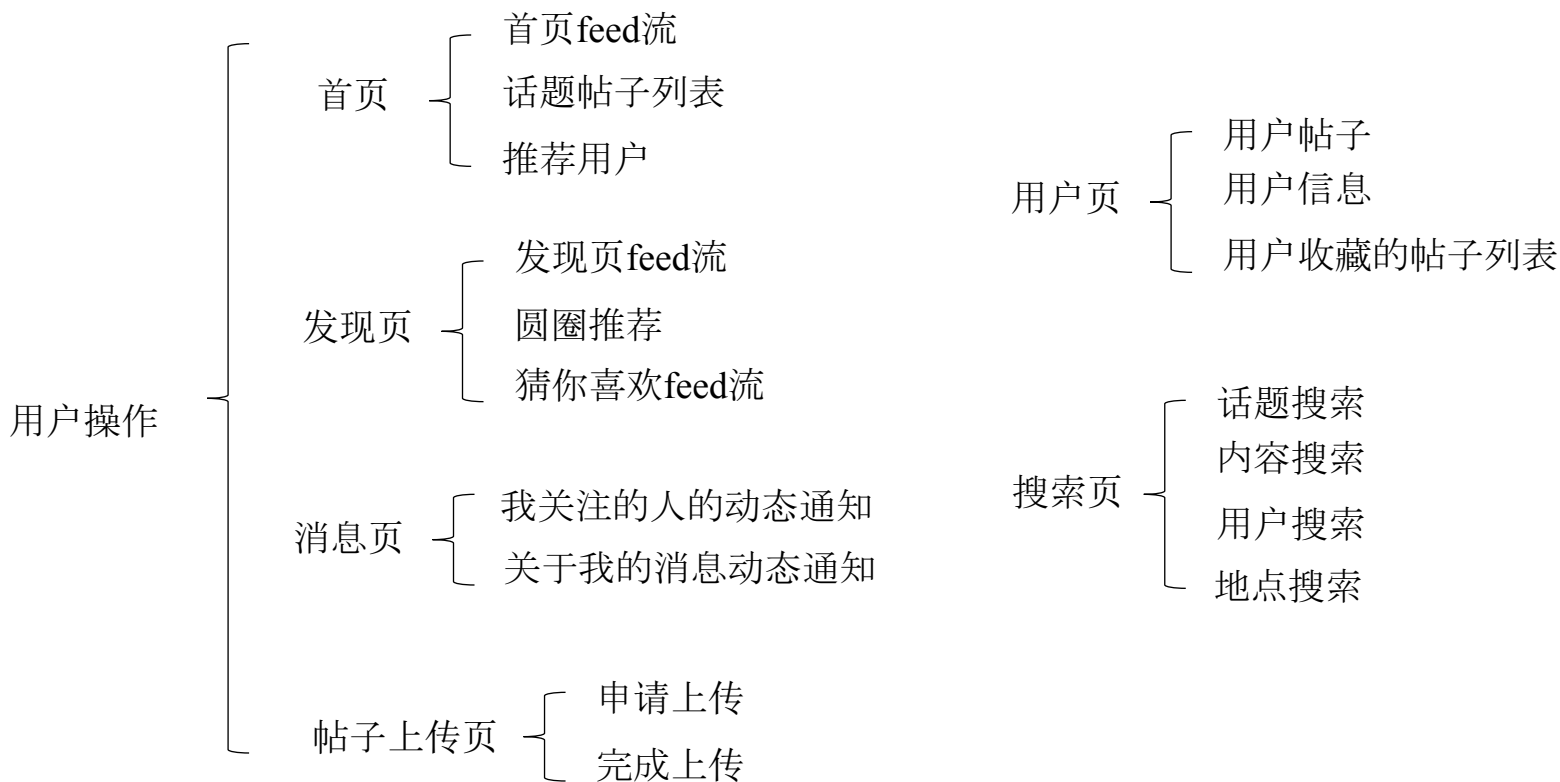


Feed流会频繁操作后台的Redis等服务，每次操作会产生100+次网络操作，200+次key/Value运算，因此会成为系统的主要性能瓶颈

备注：Feed是将用户主动订阅的若干消息源组合在一起形成内容聚合器，帮助用户持续地获取最新的订阅源内容，在社交类应用中被广泛使用

业务特点：用户增长迅速、突发事件高流量并发

Step1：以使用场景为主线，构建性能模型（使用角色、使用阶段等）



Step2：分析每个操作场景的影响因子，如好友、关注数量等，建立每个场景的测试模型

## 单场景一级接口测试

模型名称	是否需要登录	说明
发现页feed流	否	加压方式：初始压力XXX并发，每隔3分钟在原来基础上增加500并发，直到总并发压力达到XXX。 目标：查看各项性能指标，获得单场景模式下的最大并发数，和响应时间。
首页feed流	是	加压方式：初始压力XXX并发，每隔3分钟在原来基础上增加500并发，直到总并发压力达到XXX。 目标：查看各项性能指标，获得单场景模式下的最大并发数，和响应时间。
用户信息	是	加压方式：初始压力XXX并发，每隔3分钟在原来基础上增加500并发，直到总并发压力达到XXX。 目标：查看各项性能指标，获得单场景模式下的最大并发数，和响应时间。
用户帖子列表	是	加压方式：初始压力XXX并发，每隔3分钟在原来基础上增加500并发，直到总并发压力达到XXX。 目标：查看各项性能指标，获得单场景模式下的最大并发数，和响应时间。
推荐用户	是	加压方式：初始压力XXX并发，每隔3分钟在原来基础上增加500并发，直到总并发压力达到XXX。 目标：查看各项性能指标，获得单场景模式下的最大并发数，和响应时间。

## 单场景二级接口测试

模型名称	是否需要登录	说明
用户搜索接口	否	加压方式：初始压力XXX并发，每隔3分钟在原来基础上增加500并发，直到总并发压力达到XXX。 目标：查看各项性能指标，获得单场景模式下的最大并发数，和响应时间。
用户收藏的帖子列表	是	加压方式：初始压力XXX并发，每隔3分钟在原来基础上增加500并发，直到总并发压力达到XXX。 目标：查看各项性能指标，获得单场景模式下的最大并发数，和响应时间。
热门搜索接口	否	加压方式：初始压力XXX并发，每隔3分钟在原来基础上增加500并发，直到总并发压力达到XXX。 目标：查看各项性能指标，获得单场景模式下的最大并发数，和响应时间。
内容搜索接口	否	加压方式：初始压力XXX并发，每隔3分钟在原来基础上增加500并发，直到总并发压力达到XXX。 目标：查看各项性能指标，获得单场景模式下的最大并发数，和响应时间。
话题帖子列表	否	加压方式：初始压力XXX并发，每隔3分钟在原来基础上增加500并发，直到总并发压力达到XXX。 目标：查看各项性能指标，获得单场景模式下的最大并发数，和响应时间。
话题搜索接口	否	加压方式：初始压力XXX并发，每隔3分钟在原来基础上增加500并发，直到总并发压力达到XXX。 目标：查看各项性能指标，获得单场景模式下的最大并发数，和响应时间。
地点/附近地点搜索接口	否	加压方式：初始压力XXX并发，每隔3分钟在原来基础上增加500并发，直到总并发压力达到XXX。 目标：查看各项性能指标，获得单场景模式下的最大并发数，和响应时间。

如需测试某个对性能的影响，可递增方式改变因子值进行测试

按照页面权重分配压力模型，实际在生产环境比例会不断变化，因此在性能摸底过程中需要不断调整摸底

示例：全页面混合压测模型

发现页混合场景压测模型

首页混合场景压测模型

消息页混合场景压测模型

用户页混合场景压测模型

搜索混合场景压测模型

全页面混合场景压测模型

场景名称	模型名称	是否需要登录	压测比例	说明
发现页混合场景 35%	发现页feed流	否	40%*0.35	加压方式：按照每个使用场景建议的比例分配人数，按照“起始XXX用户，每10分钟逐步增加XXX用户，直到XXX并发用户”的方式，逐步增加并发人数。
	发现页圆圈推荐	否	40%*0.35	
	猜你喜欢-feed流接口	否	20%*0.35	
首页混合场景 35%	首页feed流	是	60%*0.35	
	推荐用户	是	40%*0.35	
消息页混合场景 10%	“我”关注的人的动态通知接口	是	60%*0.1	目标：查看各项性能指标，获得混合模式下的最大并发数，和响应时间。
	关于“我”的消息动态接口	是	40%*0.1	
用户页混合场景 10%	用户信息	是	40%*0.1	
	用户帖子列表	是	40%*0.1	
	用户收藏的帖子列表	是	20%*0.1	
搜索混合场景 10%	用户搜索接口	否	25%*0.1	
	内容搜索接口	否	25%*0.1	
	话题搜索接口	否	25%*0.1	
	地点/附近地点搜索接口	否	25%*0.1	



## 识别关键场景测试需求

1. HTTP协议/Rest接口
2. 用户登陆认证，模拟多用户操作
3. 支持接口串联场景，需要上下文关联
4. 性能暂无基线，需要支持递增模式快速摸底
5. 各页面用户量未知，需要灵活调整混合模型配比
6. 由于社交类应用业务增长迅速，因此需要支持按需使用，随时扩大工具的并发量
7. 需要支持10万以上的并发
8. 测试结果易于观察、保存
9. 提供监控能力，便于快速定位

测试服务选项原则：功能满足、效率高（即开即用）、成本低  
云服务更适合测试高扩展性的大规模分布式系统

测试服务需求	服务满足度
HTTP协议/Rest接口	支持GET/PUT/POST/PATCH/DELETE
用户登陆认证，模拟多用户操作	全局变量支持模拟多用户
支持接口串联场景，需要上下文关联	单事务可以支持多接口串联
性能暂无基线，需要支持递增模式快速摸底	支持自定义持续时间和每个时间段的并发量
各页面用户量未知，需要灵活调整混合模型配比	支持多事务组合测试，且为每个事务分配不同的压力
由于社交类应用业务增长迅速，因此需要支持按需使用，随时扩大工具的并发量	10万并发以内随时扩容，10万并发以上提交工单当天支持
需要支持10万以上的并发	支持
测试结果易于观察、保存	提供图形化报告，TPS，TP99,时延区间，带宽等统计
提供监控能力，便于快速定位	AOM/APM提供基础资源监控，调用链

分层开展性能测试，在集成阶段确保性能测试活动可开展

关键点

接口测试



系统测试

接口指标分解自系统指标，快速摸底  
(适合递增模型)

基于场景，不断调整事务负载配比寻找瓶颈点

契约化，以测试单接口为主

逐步加压，记录扩容过程、寻找扩容模型

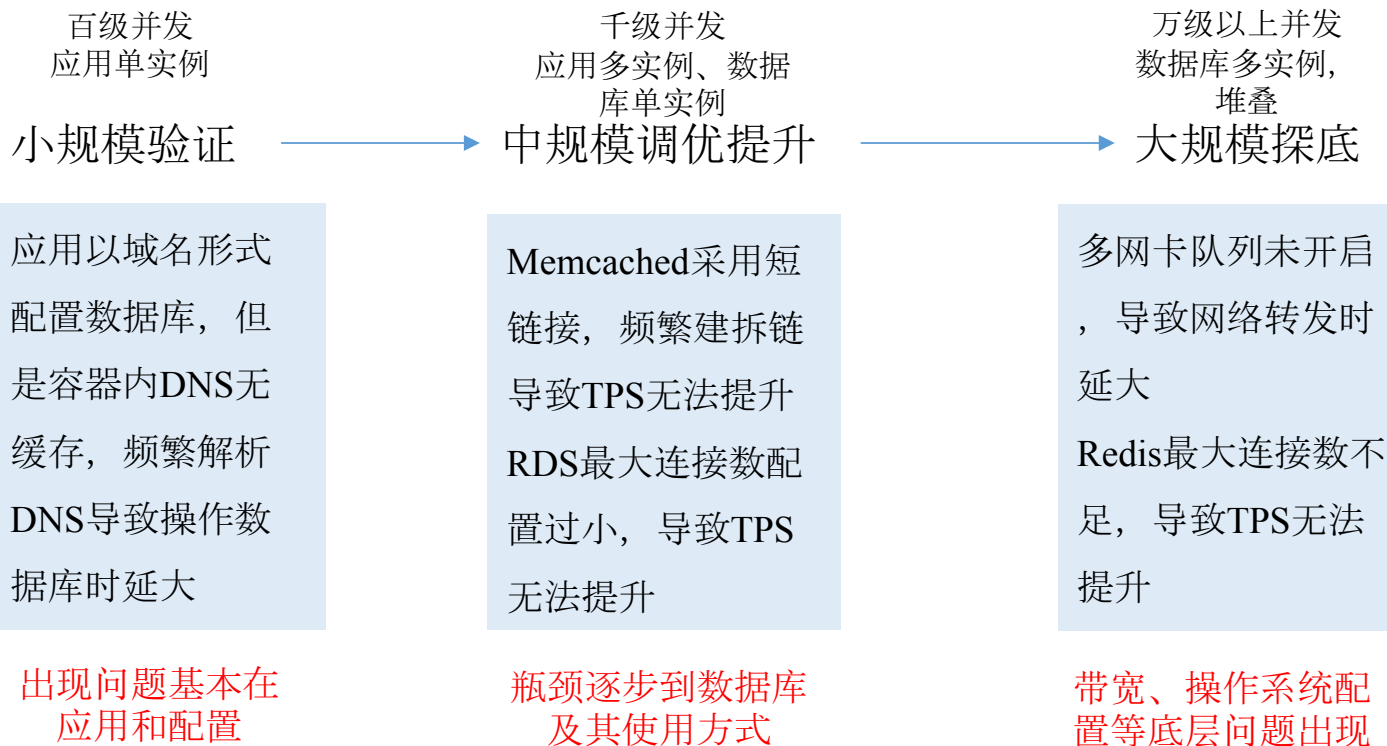
先与集成，尽早测试、尽可能自动化

模拟桩有效性需要通过实测对比校准

测试水平扩容能力

一次只变化一个因子，带背景业务

性能是一个逐步提升的过程，测试过程中需要找到扩容的模型，从不足50的TPS提升至万级



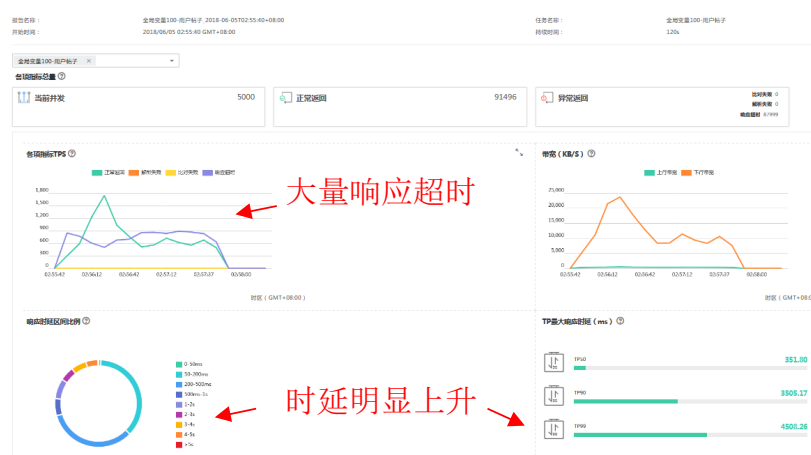
## 如何从测试工具侧快速分析被测对象可能存在的问题

- 存在部分响应超时：
  - 服务器繁忙,如某个服务节点CPU利用率高
  - 网络IO超过VM/EIP带宽
  - 等待后端微服务、数据库的超时时间设置过长
- 运行一段时间后全部响应超时或者检查点校验不通过：
  - 大压力导致系统中某个微服务奔溃
  - 后端数据库无响应
- TPS未随着并发数增长而上升：
  - 系统性能到达瓶颈，持续并发加压过程中响应时延增加（可观察响应区间统计）
  - 可通过进一步加压是否会出现非正常响应验证
- TP90响应时延较短，TP99时延高：
  - 系统性能接近瓶颈
  - 可通过进一步加压是否会出现非正常响应验证

### 100并发

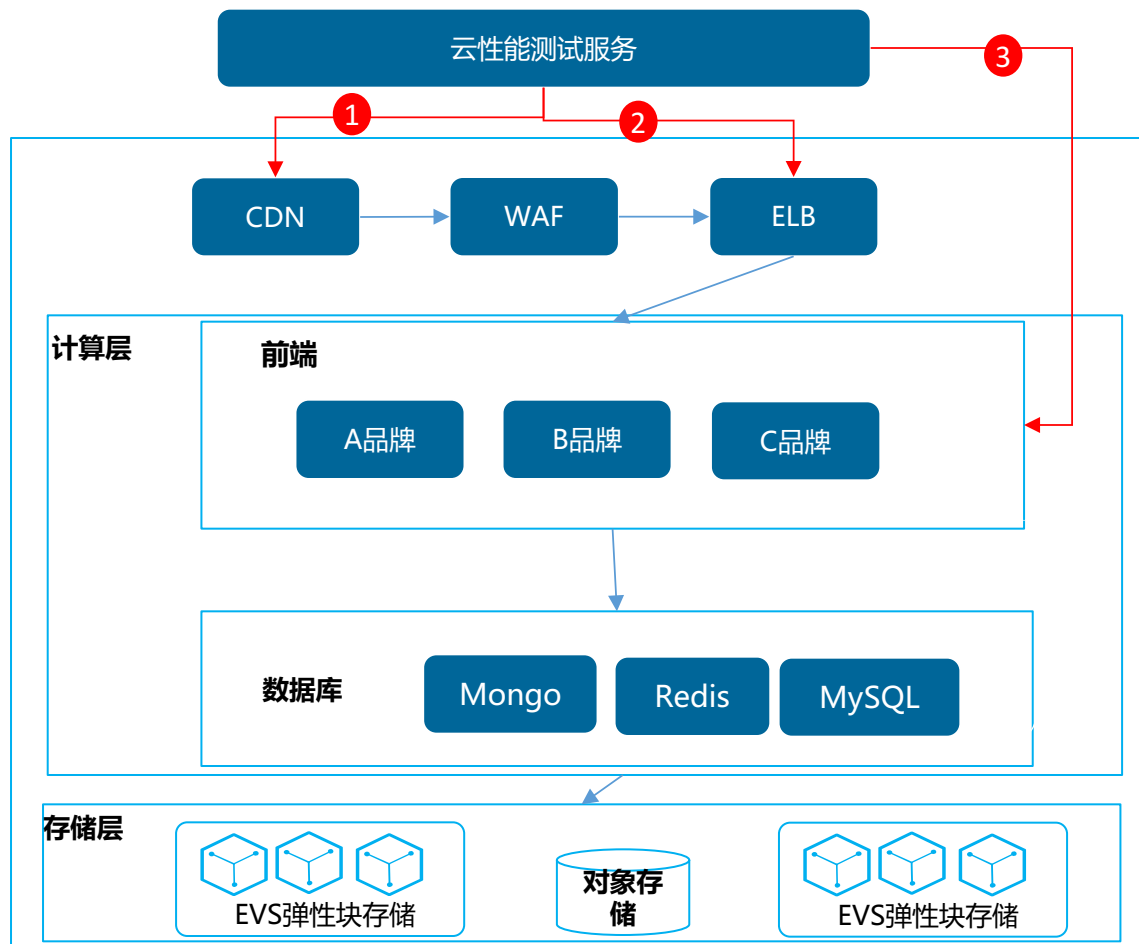


### 5000并发



1. **扩容**，链路中的某一应用可能出现cpu使用率较高或者连接池资源不够用(rpc、jdbc、redis连接池等)但本身对于拿到连接的请求处理又很快，这一类需要横向扩展资源。
2. **应用逻辑优化**，比如存在慢sql、逻辑的不合理如调用db或者redis次数过多、没有做读写分离造成写库压力过大。
3. **超时时间的合理设置**，对于应用之间的rpc调用或者应用与其他基础组件之间的调用，均需要设置合理的超时时间，否则过长的等待将造成整个链路的故障。
4. **缓存的应用，请求尽可能从前端返回**，而不是每一个都要让后端应用处理后再返回，减轻后端应用及数据库压力，提高系统吞吐能力。
5. **限流**，对于超出承载能力的QPS或并发，可以进行拦截并直接返回提示页面。
6. **降级**，对于非核心链路上的应用，允许故障关闭而不影响核心链路
7. 扩容和优化也是有限度的，在评估容量内，保障核心交易链路正常是重中之重，对于非核心功能模块考虑降级场景

业务特点：突发事件高流量突发，如瞬间由百级用户增长到万级



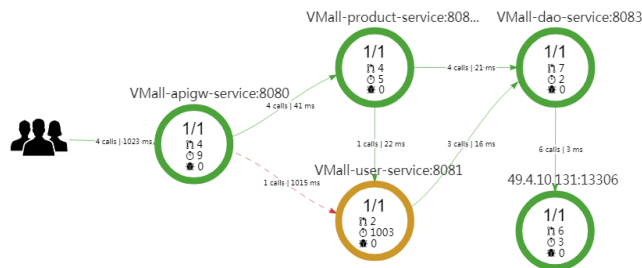
对于网络架构复杂的应用，可以通过网络架构上的分段验证，如分别从最外端的CDN入口（1）、中间的ELB（2）、业务层（3）分别做测试，验证网络架构上的瓶颈和影响

应用内部的性能瓶颈如何提升定位效率？

如：xxx并发情况下，服务A调用服务B的事务1出现问题，并直接定位至出错函数

- 在上线和活动前期通过**云性能测试服务**进行压力测试，发现部分接口的响应时间比较长，会出现比对失败和响应超时，通过**APM的调用链分析**，发现有部分SQL语句比较耗时，针对这些SQL查询语句，建立了索引，快速定位问题并迅速解决。
- 最终经过两轮测试优化后，官网首页访问响应超时与正常返回比**提升了43.3%**，预约试驾场景响应超时与正常返回比降低到0，**提升了100%**。
- 性能瓶颈定位时间，从官网未使用APM时需要1周，**缩短到**俱乐部使用APM后的**0.5天，效率提升90%**



























## 应用拓扑



## 事务监控

名称	事务名称	事务次数	事务耗时(ms)	连接数	Avgops	网络带宽
PGSQL_authentication	验证	0	0	0	0	0
PGSQL_superuser	登录	1	1023	0	0.5	500
PGSQL_query		0	0	0	0	0
PGSQL_commit		0	0	0	0	0
PGSQL_disconnect		0	0	0	0	0
PGSQL_query		0	0	0	0	0
PGSQL_productproducts	商品查询	2	26	0	1	500
PGSQL_productproducts	商品订单	1	51	0	1	500

## 调用链跟踪

名称	方法	参数	状态	耗时(ms)
small-http-service	 org.apache.catalina.connector.http.HttpHandler	product1(product1)		<1
small-http-service	 org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter			3
small-http-service	 com.netflix.hystrix.jhystrix.HystrixCommandQueue			<1
small-http-service	 com.netflix.hystrix.jhystrix.HystrixCommandQueue			<1
small-http-service	 org.apache.http.impl.client.CloseableHttpClient			3
small-http-service	 org.apache.http.impl.client.HttpClientBuilder			5
small-http-service	 org.apache.http.protocol.HttpRequestExecutor	product1(7421)		<1
small-product-serv	 org.apache.catalina.connector.http.HttpHandler	product1(7421)		1
small-product-serv	 org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter			2
small-product-serv	 com.hazelcast.core.config.Config			2
small-product-serv	 sun.net.www.protocol.http.HttpURLConnection			10
small-product-serv	 sun.net.www.protocol.http.HttpURLConnection			10
small-product-serv	 org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter			10



Anyone, Anywhere, Anytime

要解决的问题

突发流量导致应用奔溃

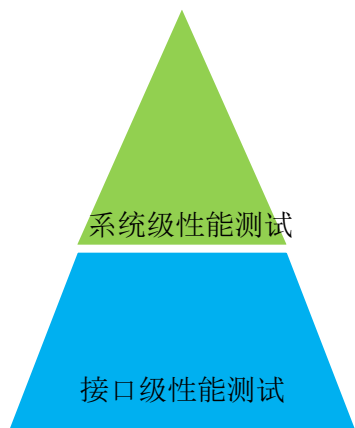
高并发情况下应用响应慢

促销活动不知道准备多少资源

线上性能问题难定位

提前预防、主动运维

Anyone，免编码开展各层次性能测试



Anywhere，测试模型灵活定义，满足各行各业



Anytime，即开即用，任何阶段可开展性能测试



华为云性能测  
试服务



个人微信





THANK YOU