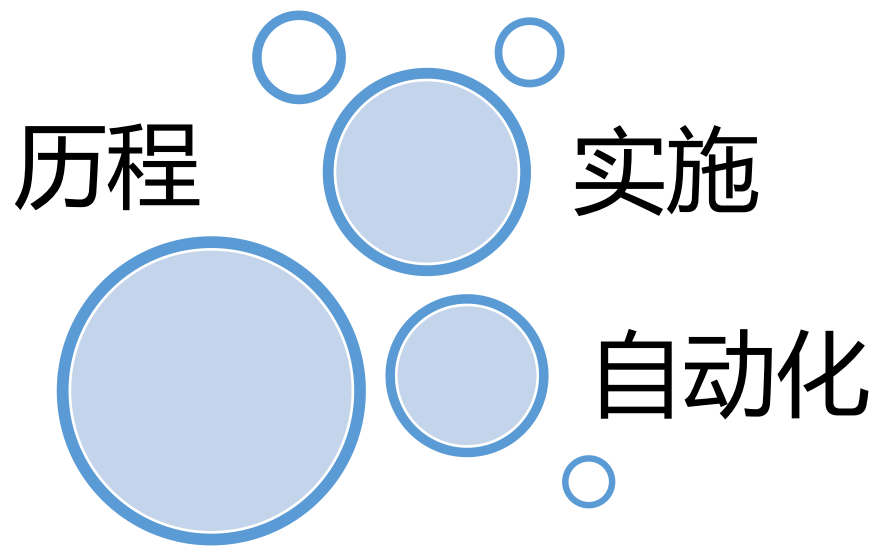


# 饿了么全链路压测 实践与体系建设

吴骏龙/严佳奇  
饿了么测试基础设施部



## 背景

自2015年开始，随着互联网行业的快速发展，饿了么公司的业务也进入了快速扩张阶段，用户量屡创新高。在业务快速发展的背景下，如何持续地保障全站的容量安全，同时又能兼顾服务器资源的合理分配及扩缩容，避免资源浪费，这是一个极具挑战的任务。



容量安全

## 业务特点

### 时效性

从用户下单到商家接单再到物流配送到家，整个流程要控制在一定时间范围之内

### 高并发

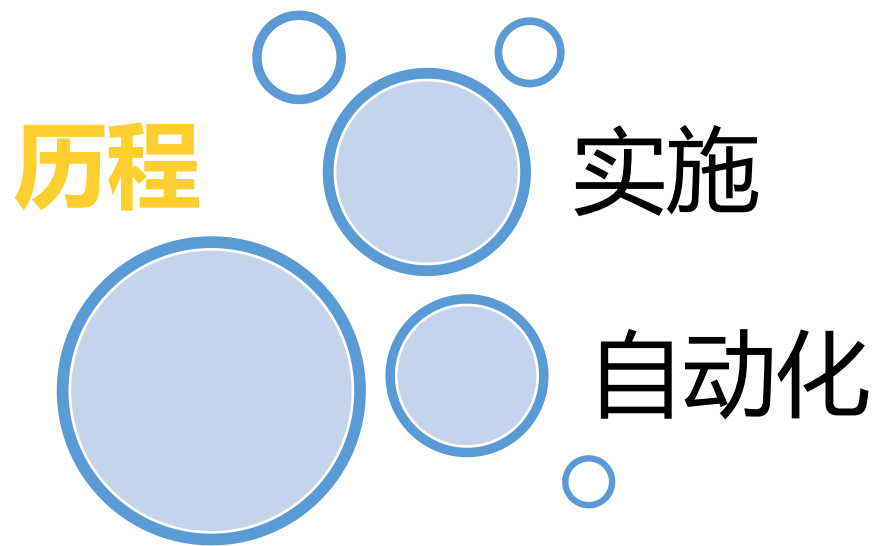
大量用户产生的千万订单集中分布在中午和傍晚两个时间段内

### 秒杀活动

秒杀活动所产生的瞬时流量甚至能超过午高峰的最高值

### 常规化

大流量的冲击不是偶发的，而是一个常态化的过程



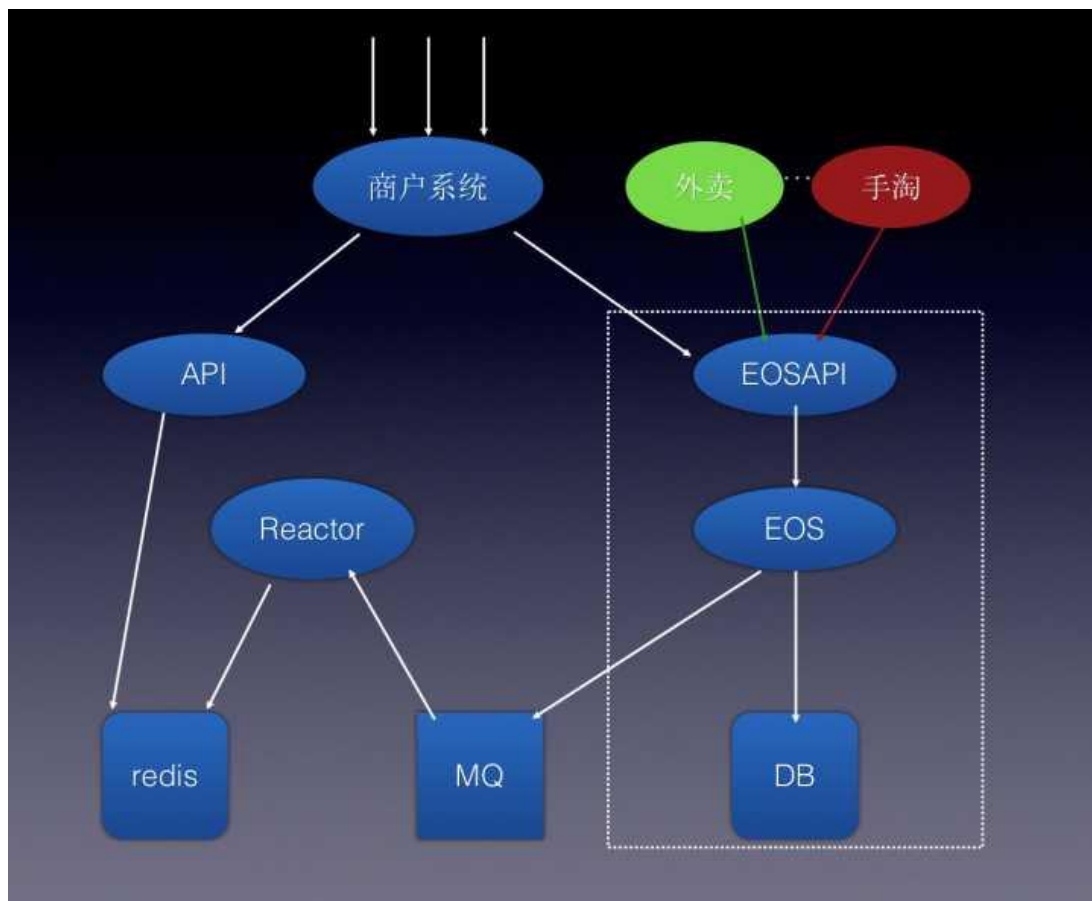
## 第一阶段 缩减服务器

在低峰时间段，通过逐台缩减集群内服务器的数量，使单台服务器的请求量不断加大，从而评估当前集群容量及预估后续随着单量的增长所需服务器的数量。

优点	缺点
真实流量，评估出的容量最真实	风险性比较高，达到瓶颈须立刻恢复
不用编写压测用例和准备压测数据，节省大量压测准备的时间	由于请求量恒定，通过缩减服务器数量的方式并不能评估出底层基础组件的容量，从而会导致容量评估不准确
无脏数据产生	对于访问流量有一定要求，流量太小的话达不到瓶颈

## 第二阶段 单独业务压测

针对单个业务在低峰期进行压测，这就需要压测人员对业务有一定的敏感度，对业务和系统架构有比较深的了解。



## 第三阶段 线上全链路压测

- 模拟各平台下单，商户接单和物流配送

写

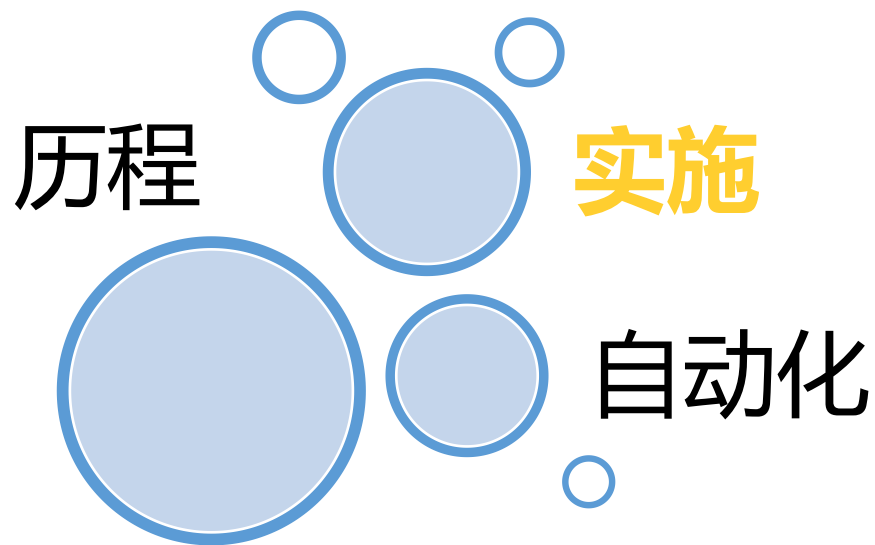
- 模拟大量的用户查询操作，覆盖所有关键路径的接口

读

通过不断从各个入口加压，既能做到对各个服务的容量做到评估，也能观测底层服务(包括各个中间件)的性能指标，对整个业务系统的容量评估也相对精准。

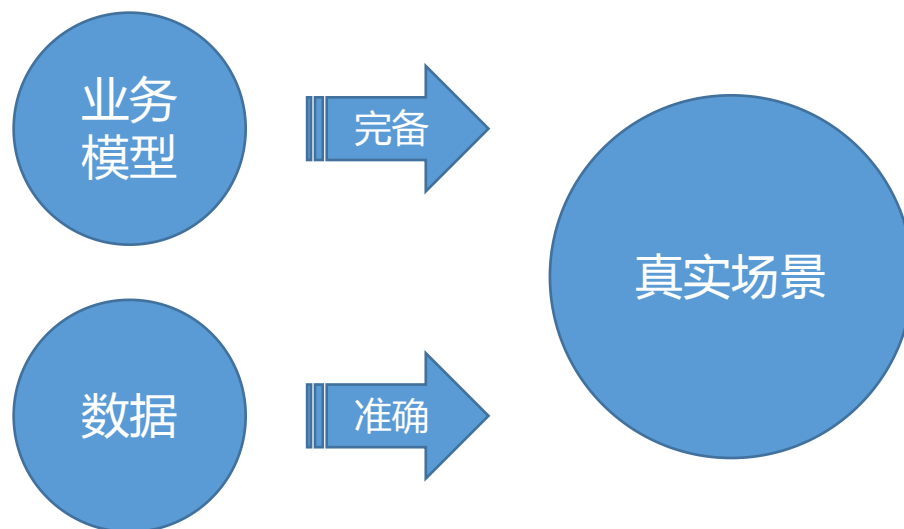






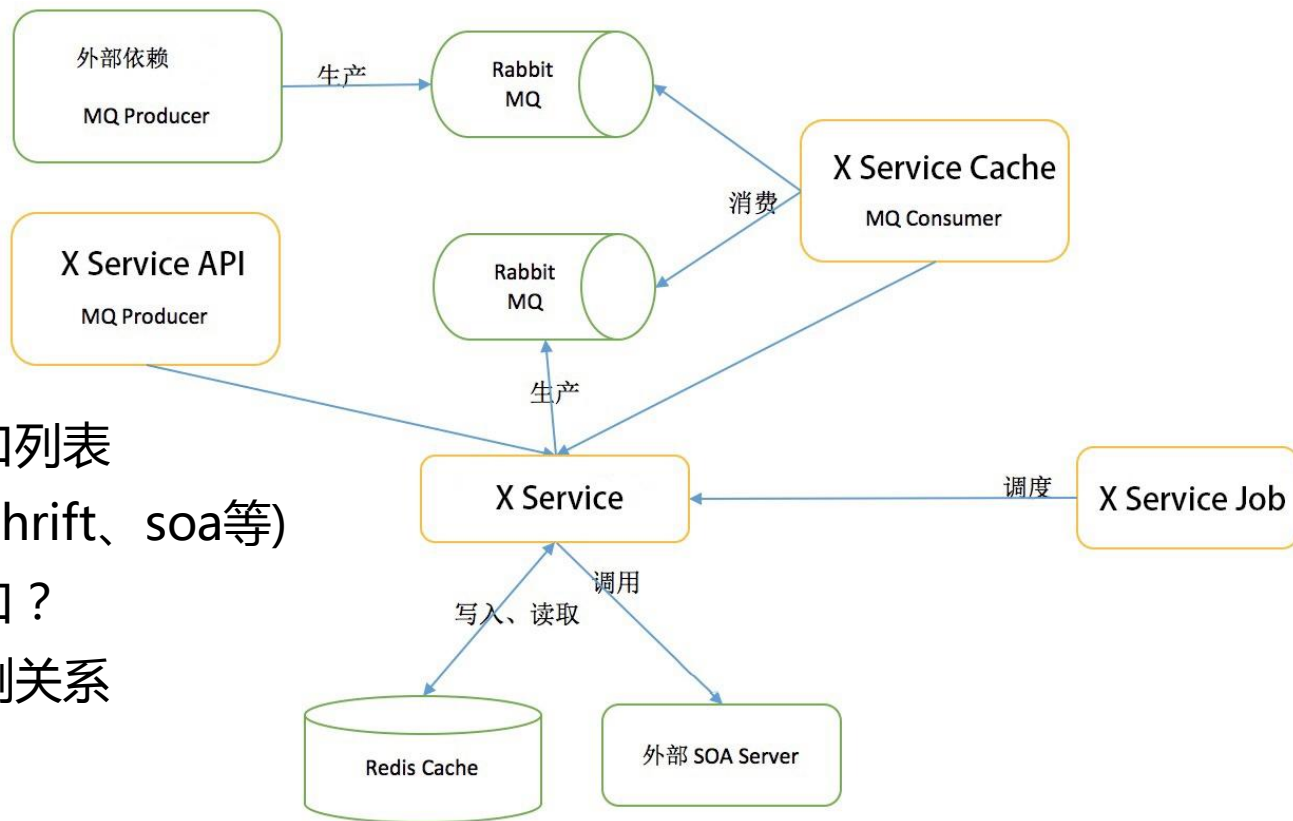
## 业务模型的梳理

业务模型梳理需要结合业务本身和业务系统的架构，这一步至关重要，梳理的是否完善直接关系到最终压测结果是否具有参考价值。



## 业务模型实例

- 是否关键路径
- 业务的调用关系
- 业务的提供的接口列表
- 接口类型(http、thrift、soa等)
- 读接口还是写接口？
- 各接口之间的比例关系



## 数据模型的构建

总原则

紧贴业务场景，最大可能地模拟真实用户的请求。

压测场景  
(写请求)

用户下单

压测方法

- 用户、商户、菜品等在数量上与线上等比例缩放
- 对压测流量进行特殊标记
- 根据压测标记对支付，短信等环节进行mock
- 根据压测标记进行数据清理

## 数据模型的构建

总原则

紧贴业务场景，最大可能地模拟真实用户的请求。

压测场景  
(读请求)

商家列表及关键字查询

压测方法

拉取线上日志，根据真实接口比例关系进行回放

## 数据模型的构建

总原则

紧贴业务场景，最大可能地模拟真实用户的请求。

压测场景  
(无日志服务)

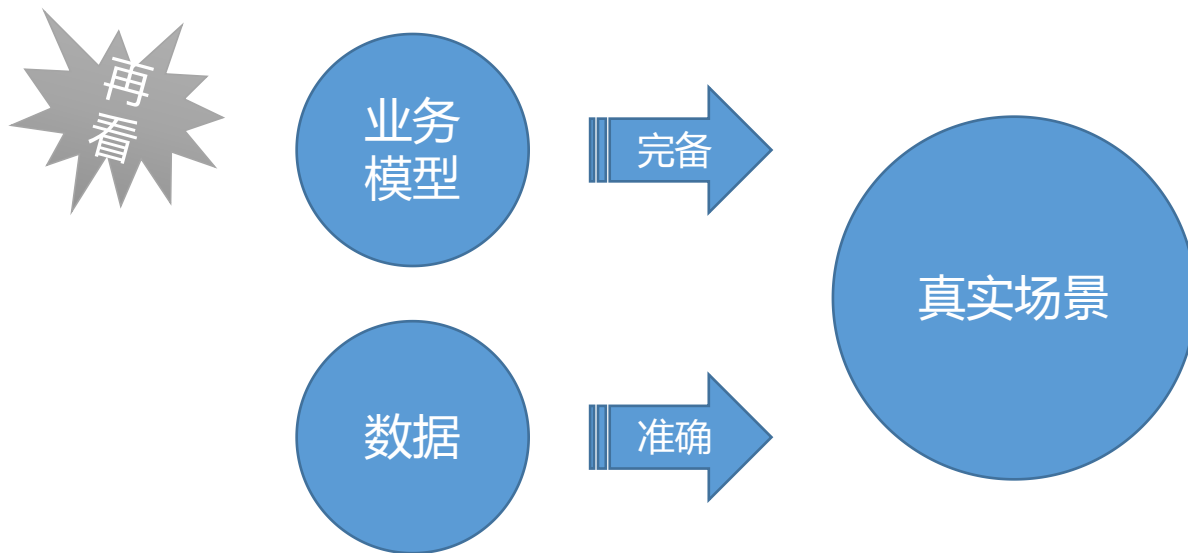
商家资质查询

压测方法

- 构建压测数据使缓存命中率为0%时，服务接口性能，数据库性能
- 缓存命中率为100%时，服务接口性能
- 缓存命中率达到业务预估值时，服务接口性能

## 实践中遇到的一些坑

- 压测用户数据未考虑sharding分布，导致DB单点过热
- 用户数量过少，导致单个测试用户订单量过多
- 商家数量过少，导致菜品减库存锁争抢激烈



## 压测指标

### 应用层面

- 错误率
- 吞吐量
- 响应时间
- GC

### 服务器资源

- CPU利用率和负载
- 内存
- 磁盘I/O
- 网络I/O
- 连接数

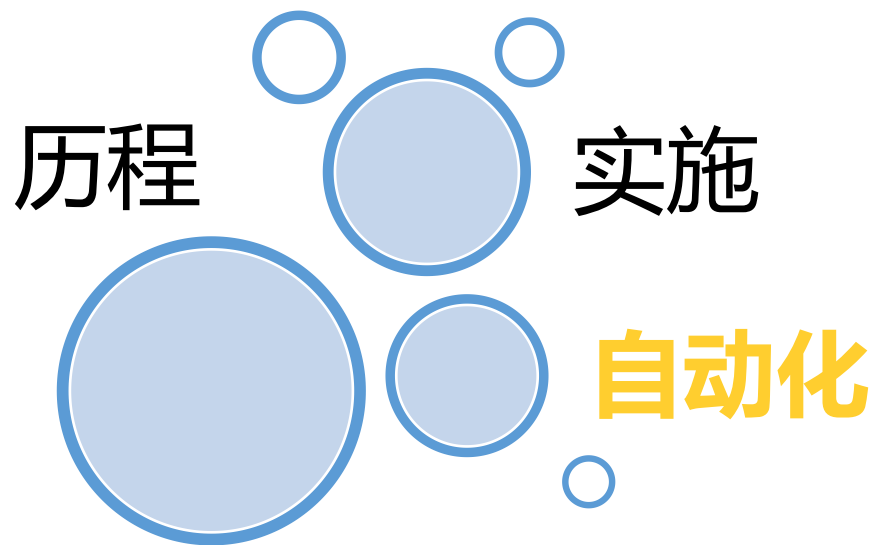
### 基础服务

- MQ
- Redis
- DB
- 其它中间件

### 注意点

- 响应时间不要用平均响应时间，关注95线；
- 吞吐量和响应时间挂钩
- 吞吐量和成功率挂钩



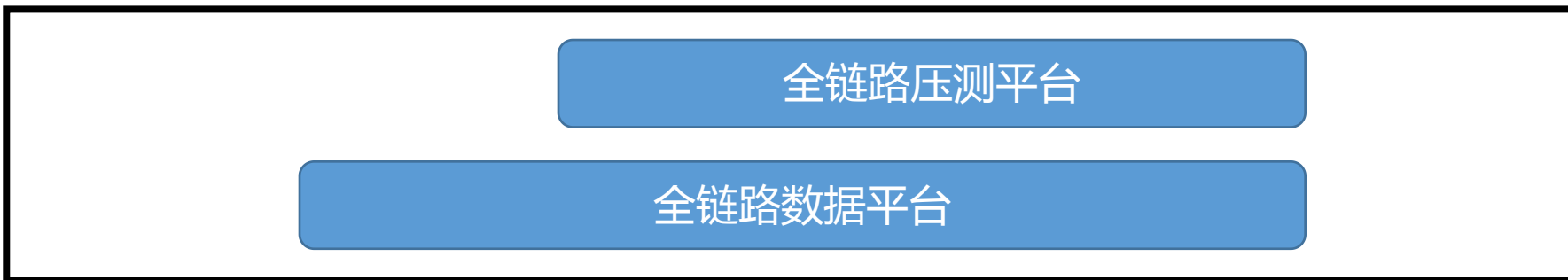
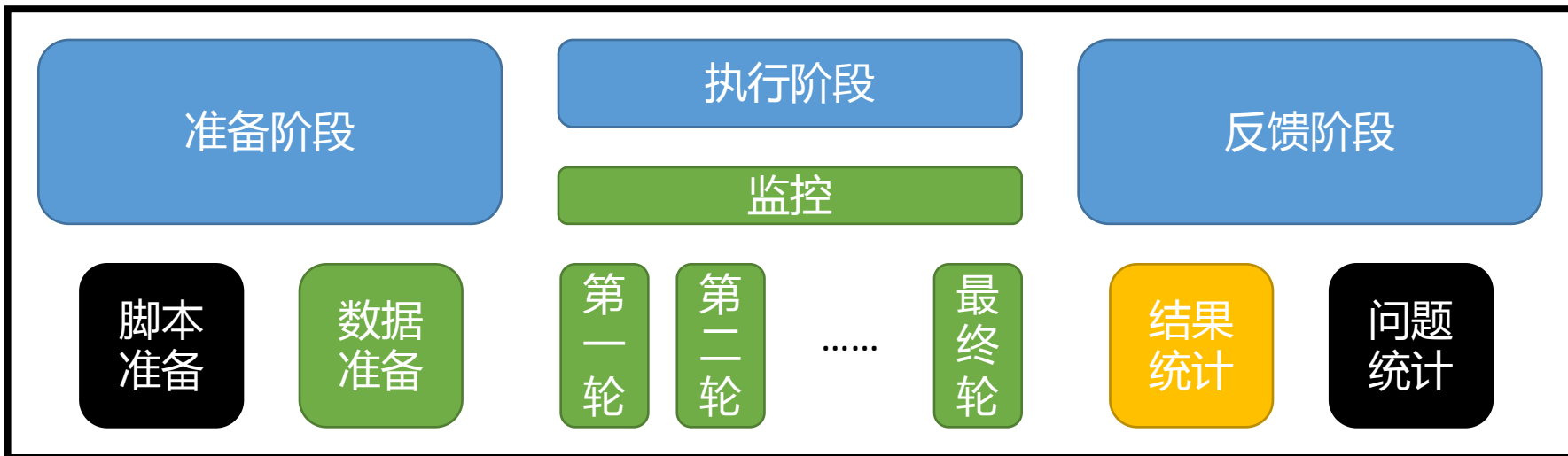


## 全链路压测整体体系

人工

全自动

半自动



## 全链路压测平台

一站式平台

配置

触发

运行

结果  
收集

清理

测试用例  
分类管理

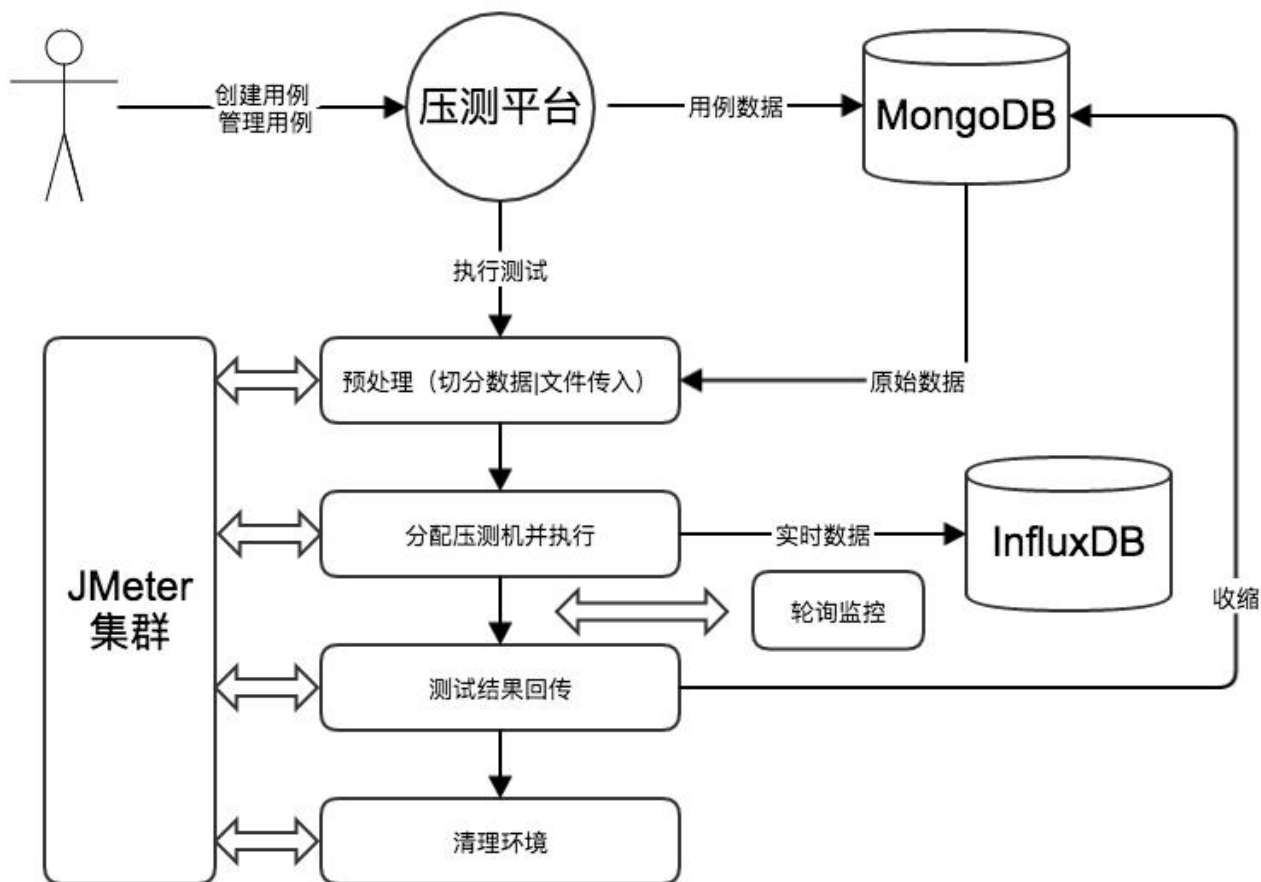
一键触发

测试过程  
监控

热数据(实时)  
冷数据(详细)

压测机  
监控

## 全链路压测平台架构



## JMeter分布式执行的缺点

### 执行方式差异大

这会导致平台架构不必要的复杂度，实际用户只感知测试机的数量区别

### 资源浪费

master机通常不参与测试，而是收集slave信息

### 影响性能

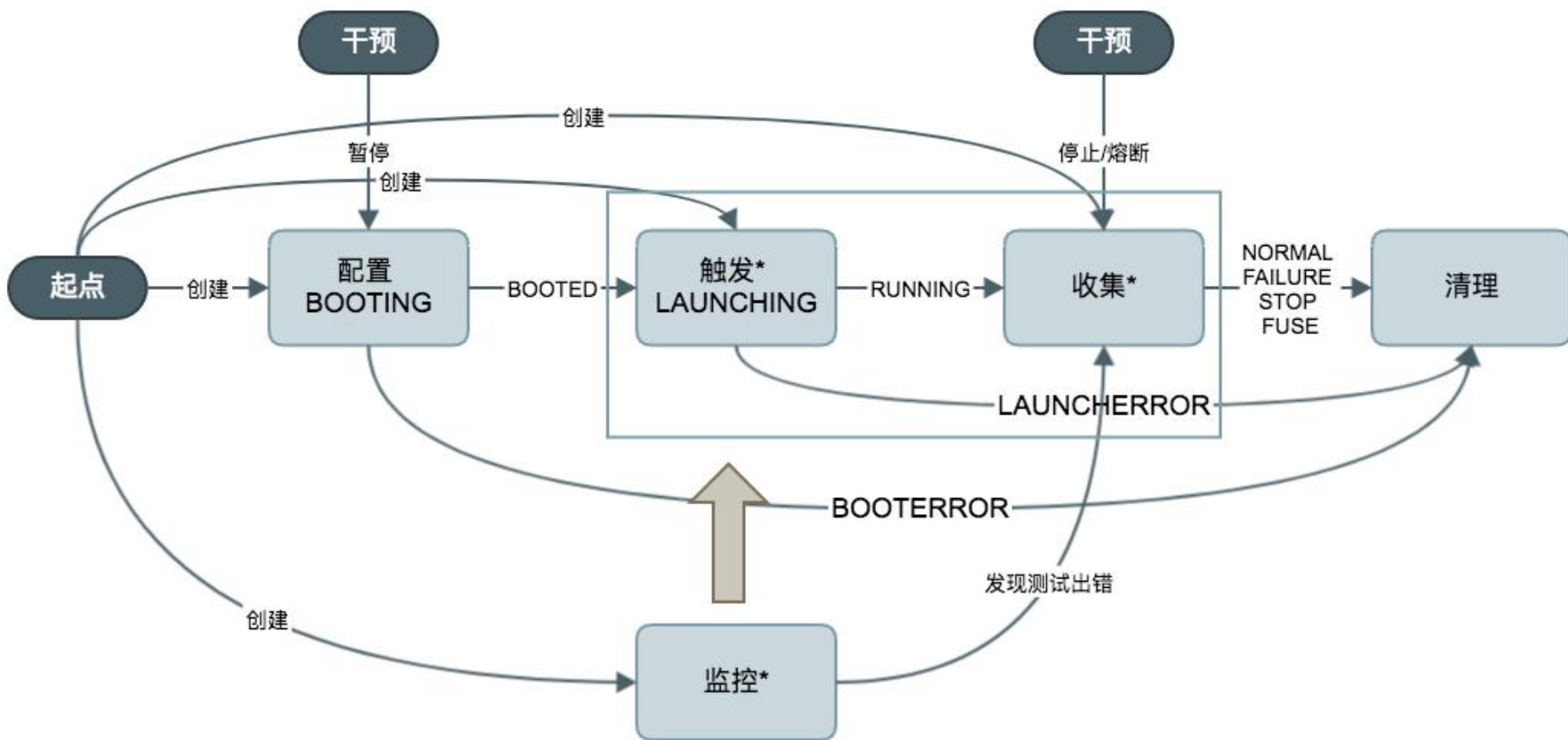
slave不间断传输数据给master，会一定程度上影响性能（尤其是带宽）

## 全链路压测平台的分布式实现

- 服务器统一调度测试过程
- 冷热数据分离

功能	JMeter实现	压测平台实现
脚本分发	master进行分发	服务器统一分发，可预配置
数据文件分发	用户自行上传至每台压测机	服务器统一分发，可切分
结果回传	各slave回传至master	冷热分离回传，聚合入库
节点失败重试	统一配置重试	自动重试+人工重试

## 测试状态流转



## 实时数据

### JMeter backend listener – UDP to InfluxDB (Jmeter 3.3 Core Improvement)

**Backend Listener**

Name: Backend Listener

Comments:

Backend Listener implementation: org.apache.jmeter.visualizers.backend.influxdb.InfluxdbBackendListenerClient

Async Queue size: 5000

**Parameters**

Name:	Value
influxdbMetricsSender	org.apache.jmeter.visualizers.backend.influxdb.UdpMetricsSender
influxdbUrl	influxDbHost:8089
application	application name
measurement	jmeter
summaryOnly	true
samplersRegex	*
percentiles	90:95:99
testTitle	Test name
eventTags	



## 非实时数据（冷数据）

冷数据收集 - 测试结束后基于JTL进行数据合并和聚合，数据精度需要考虑

- **趋势类曲线(如平均响应时间)**：不需要太精确，以固定60个时间片聚合
- **特征点曲线(如最大响应时间)**：需要精确，以固定60个时间片聚合取特征值
- **特征点单值(如90/95/99线)**：需要精确



## 全链路压测数据平台

测试数据抓取

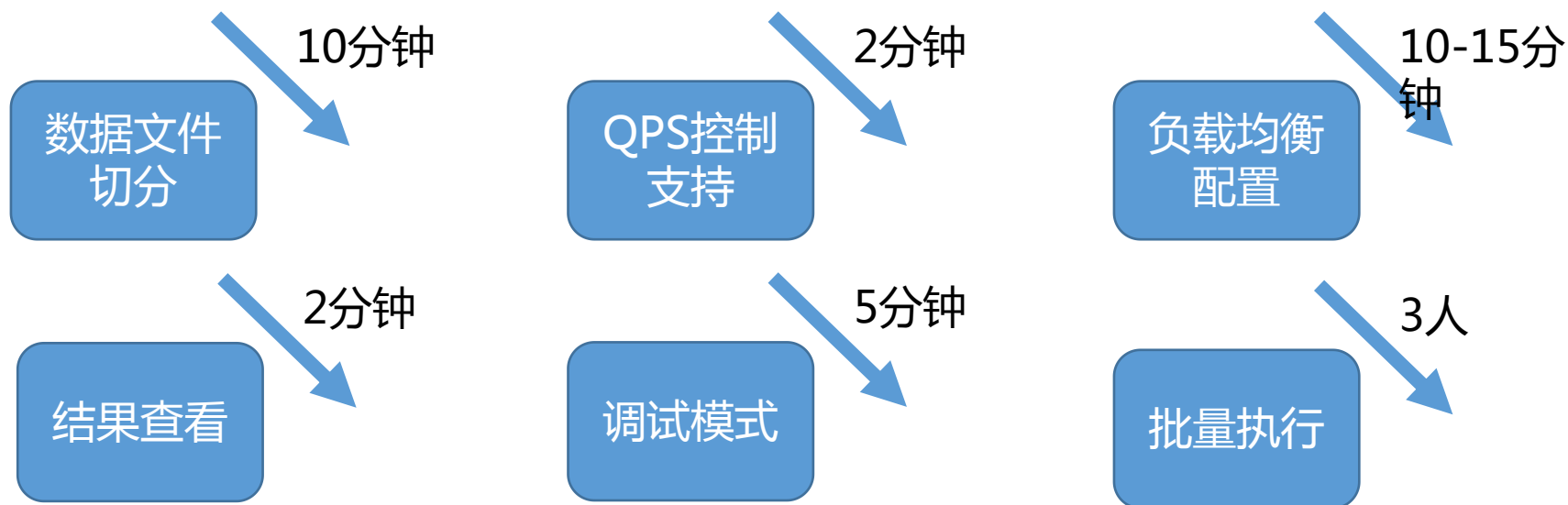
全链路压测期间  
各指标监控

高峰期基准容量测算

全链路压测报表



## 自动化 - 效率 - 成本



全链路压测耗时从约**4**小时降低到约**2.5**小时

全链路压测值守人员从**4**人次降低到**2**人次

## 成果

### 常态化 标准化

每周常规进行全链路压测，紧跟业务发展趋势

### 及时发现重要问题

容量问题、代码优化、配置问题、数据库相关问题等

### 容量保障

为产研、运维团队提供容量依据，为全站容量安全保驾护航

GIAC | 全球互联网架构大会  
GLOBAL INTERNET ARCHITECTURE CONFERENCE

GIAC

全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE



扫码关注GIAC公众号

2017.thegiac.com