

C<sub>trip</sub> E<sub>asy</sub> T<sub>est</sub>

自动化/性能的无缝融合



# ● 自我介绍



## 蔡晓昕 (小新)

2015年加入携程



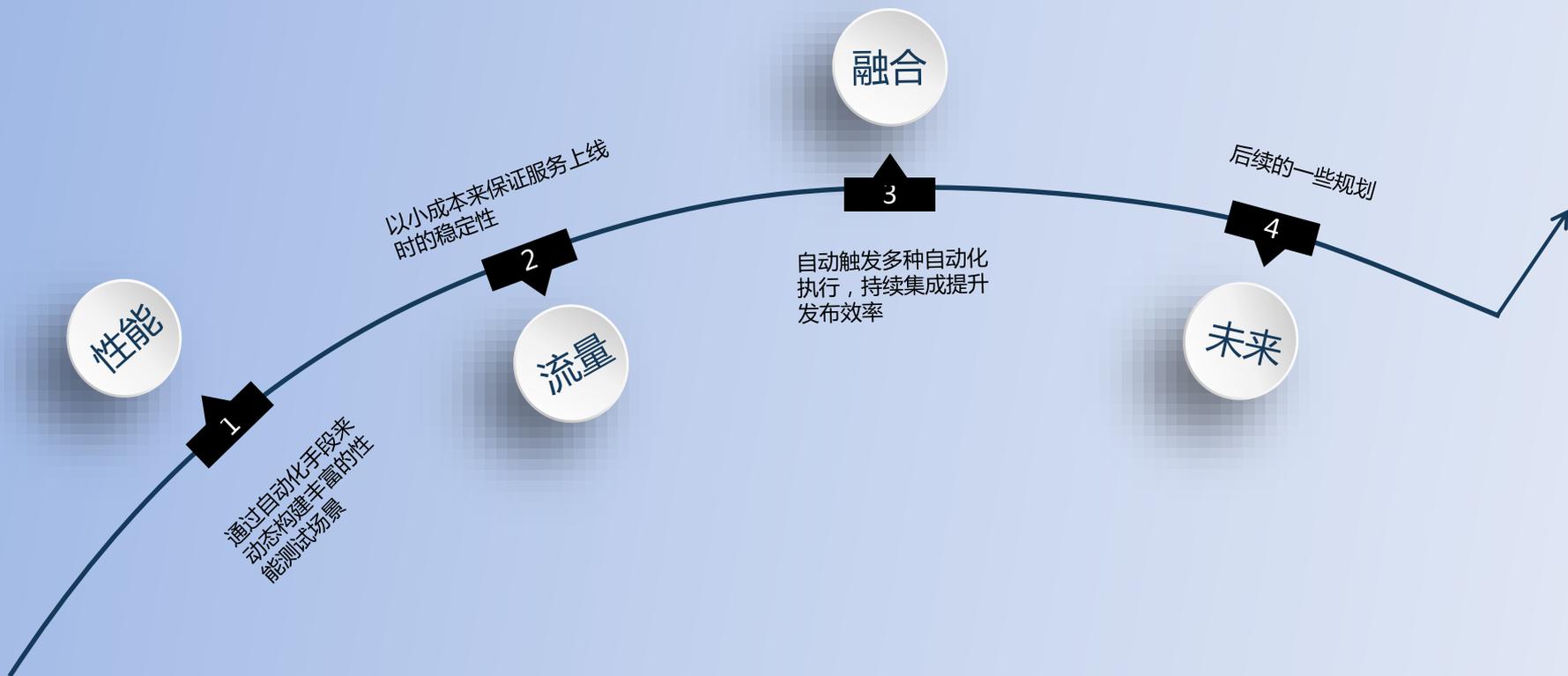
负责酒店移动应用  
自动化测试，以及测试平台研发



携程酒店  
APP自动化测试经理



热衷于新技术的研究，  
利用技术手段优化测试过程，  
提高测试效率



1

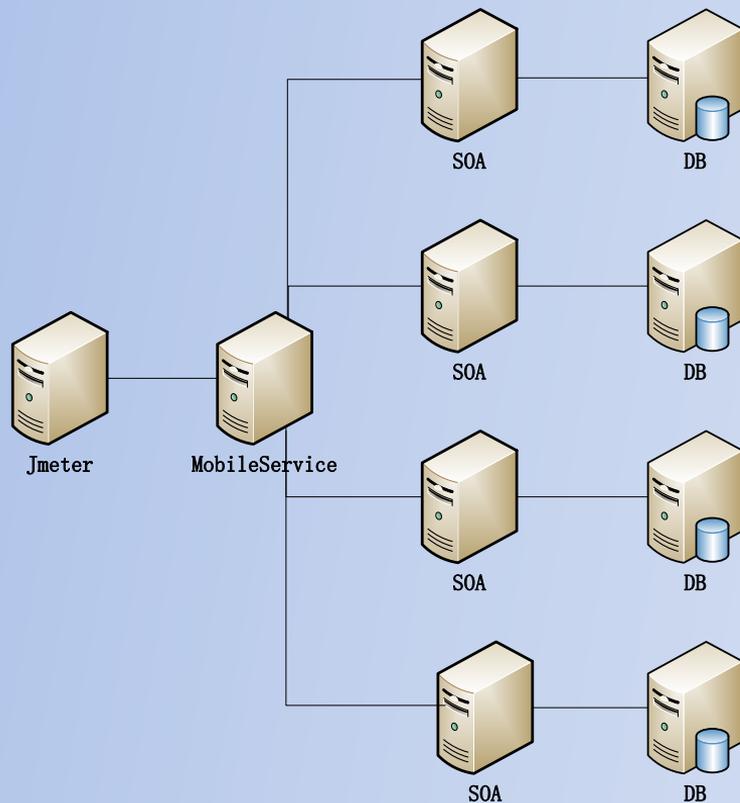
性能场景自动化构建

初试

**关键字：一切为了高仿真**

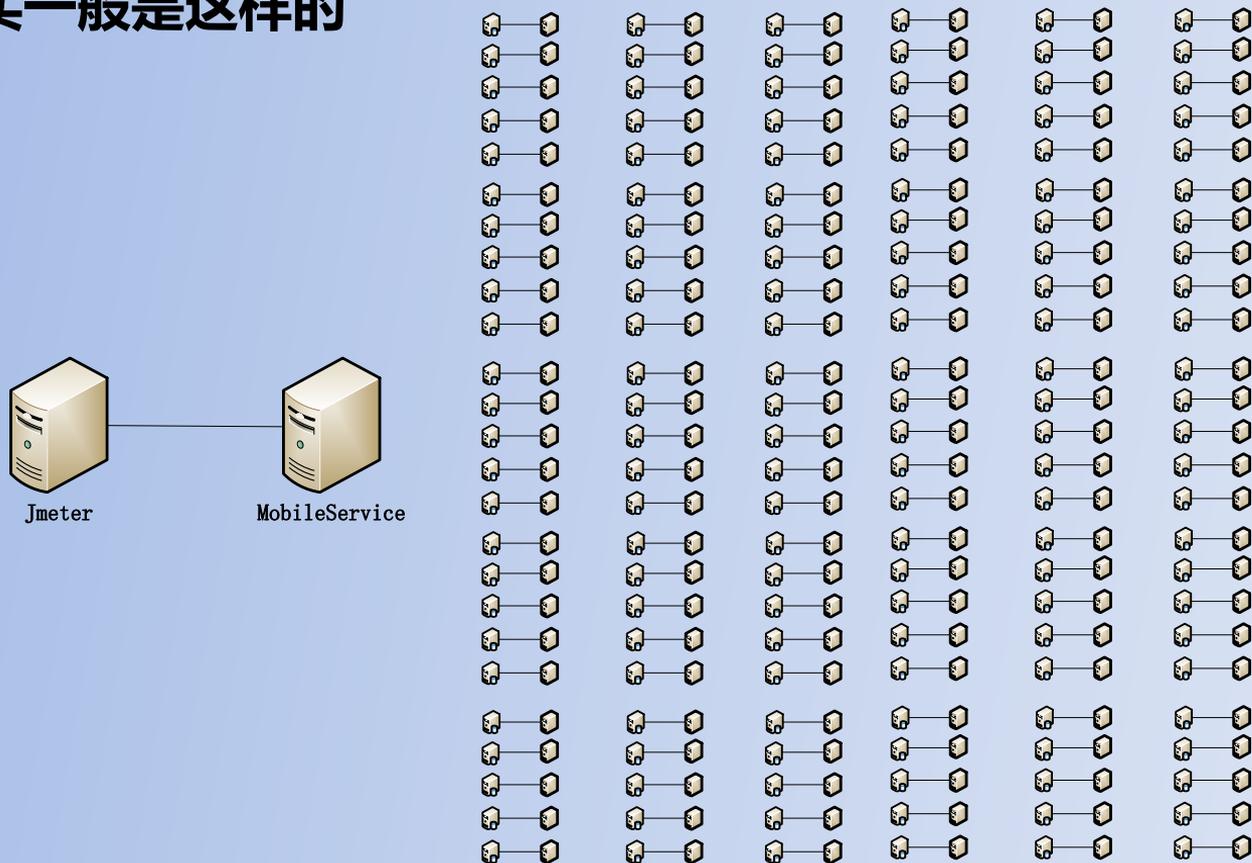


## 一开始的性能测试一般都是这样想的



# ● 发现问题了

## 然而现实一般是这样的

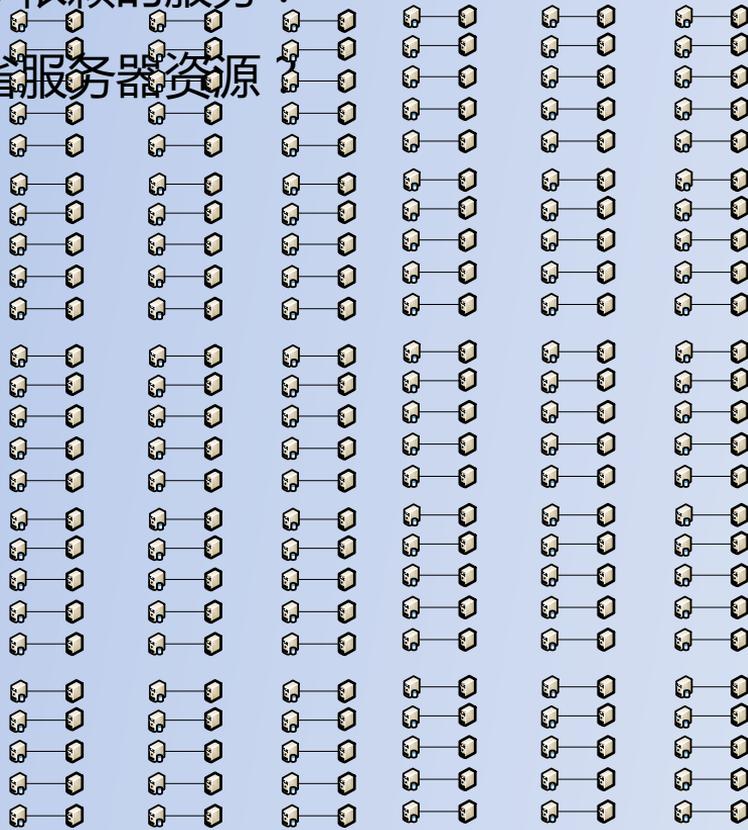
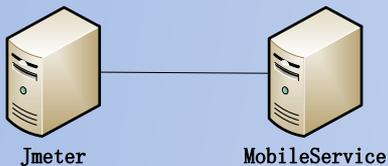




# 图还是这张老图，但是这次我们做了点什么

有没有办法不去人工维护那么多依赖的服务？

有没有办法同时可以更加的节省服务器资源？



初试

探索

进阶

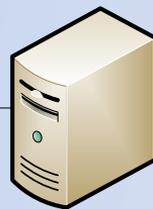
## 关键字：万服归宗（Mock）



Jmeter



MobileService



Mock





如何丰富性能测试场景？

有没有什么关键字可以从头到尾带给每个依赖来帮助我们做mock数据收集？

在写API自动化测试的时候，我们会有丰富的API测试场景

那么能否在此时做埋点，让Mobile Service把埋点带向SOA，并记入日志系统呢？

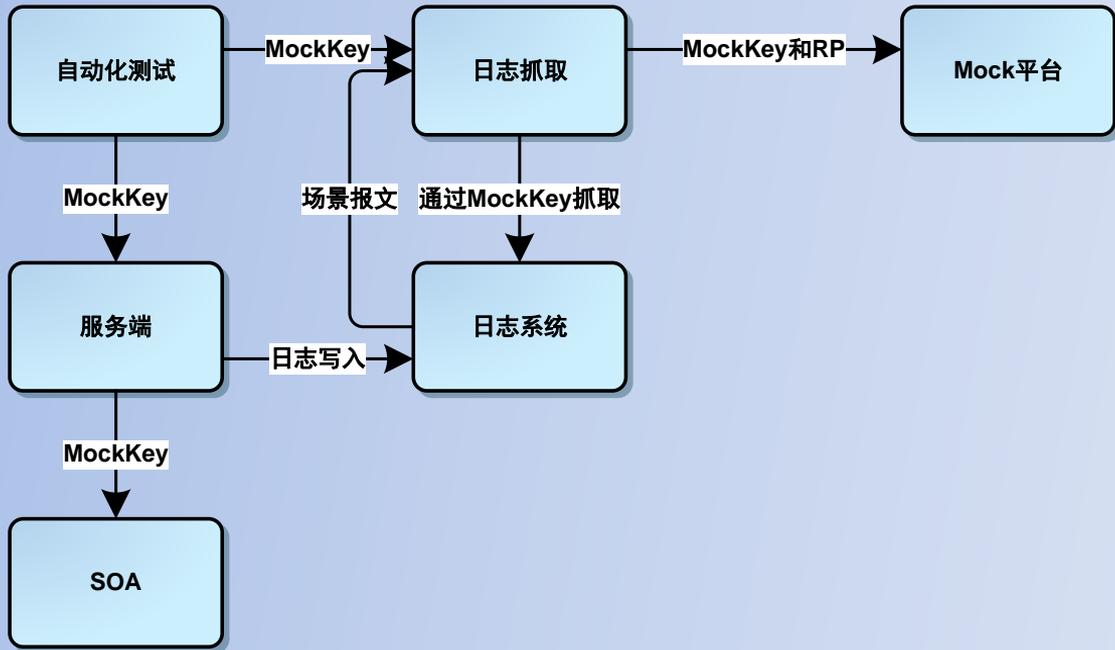


# ● 现在的性能测试数据构造

## 关键字：一键

场景入手点：自动化测试

统一关键字：在非契约的地方藏一个通用的key，直传给所有依赖的soa服务



## ● 现在的性能测试

---



总是回归很多很多老case，辛苦好几天才敢上生产  
一接生产流量，搞不好还会有各种监控报警，可能还要回退  
那么有没有可以偷懒，又好用的解决方案？

**我觉得，这个确实可以有**



2

玩转大流量

## 生产流量也不是想玩就玩~

流量太多，一时半会儿怎么玩的过来？

其实已经有一些开源的导流工具了，比如：tcpcopy。但是为什么要自己玩？

量级很大，如何回放？



## 智能采样

按照配置策略动态取数



## 智能分离

分离数据中的读写接口，只保留读的部分用于生产回放

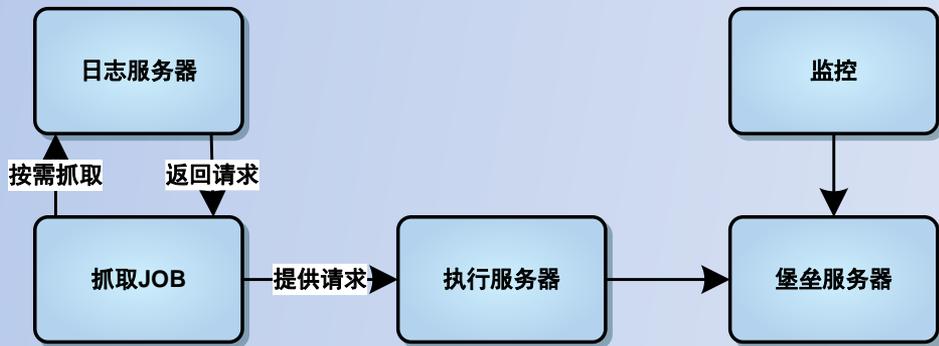


## 智能回放

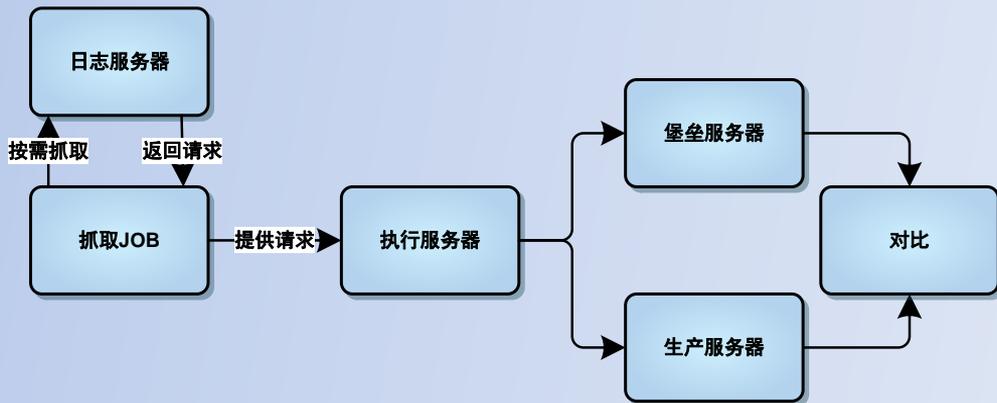
自由控制流量回放速度、场景、接口进行回放

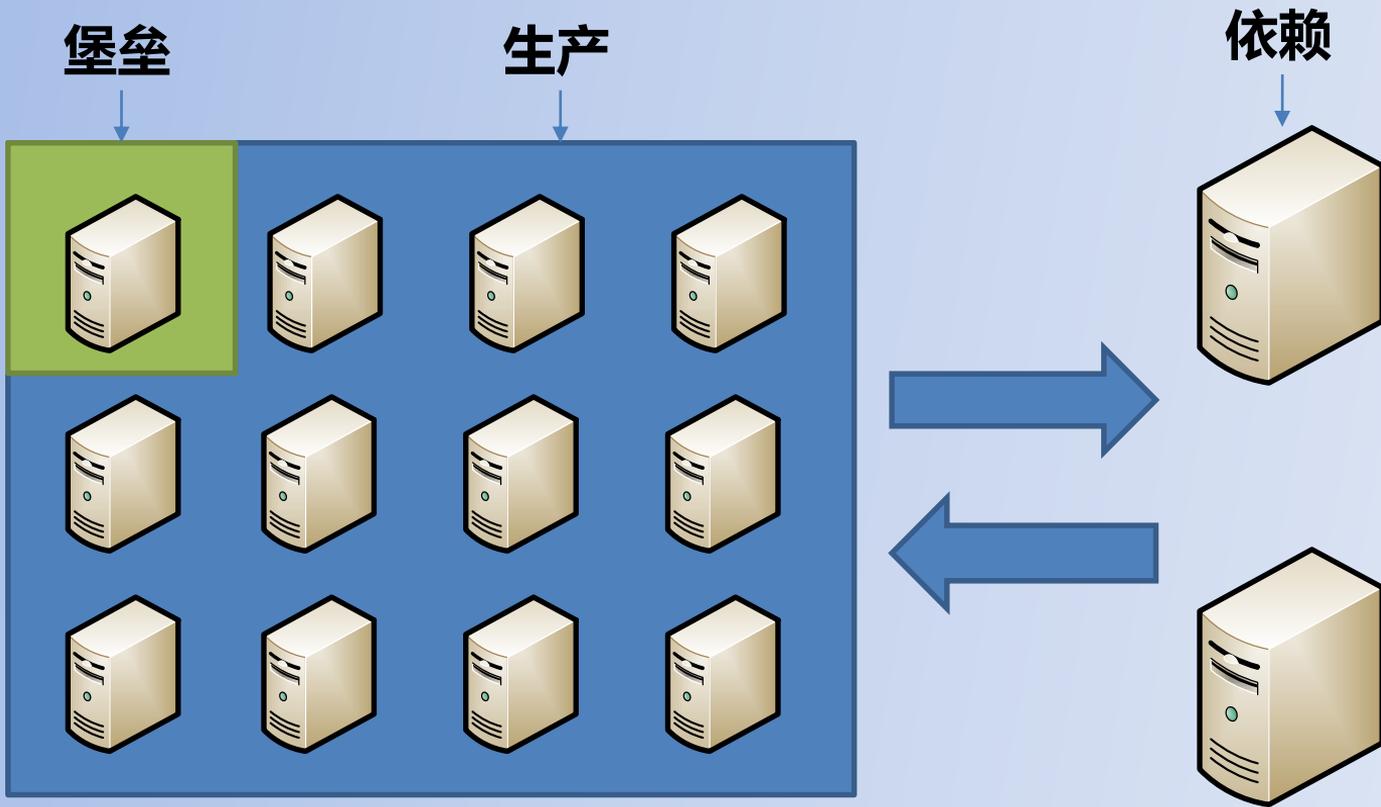
# 智能回放的验证场景

## 大流量、做监控



## 自动化做对比





## 发现问题了？

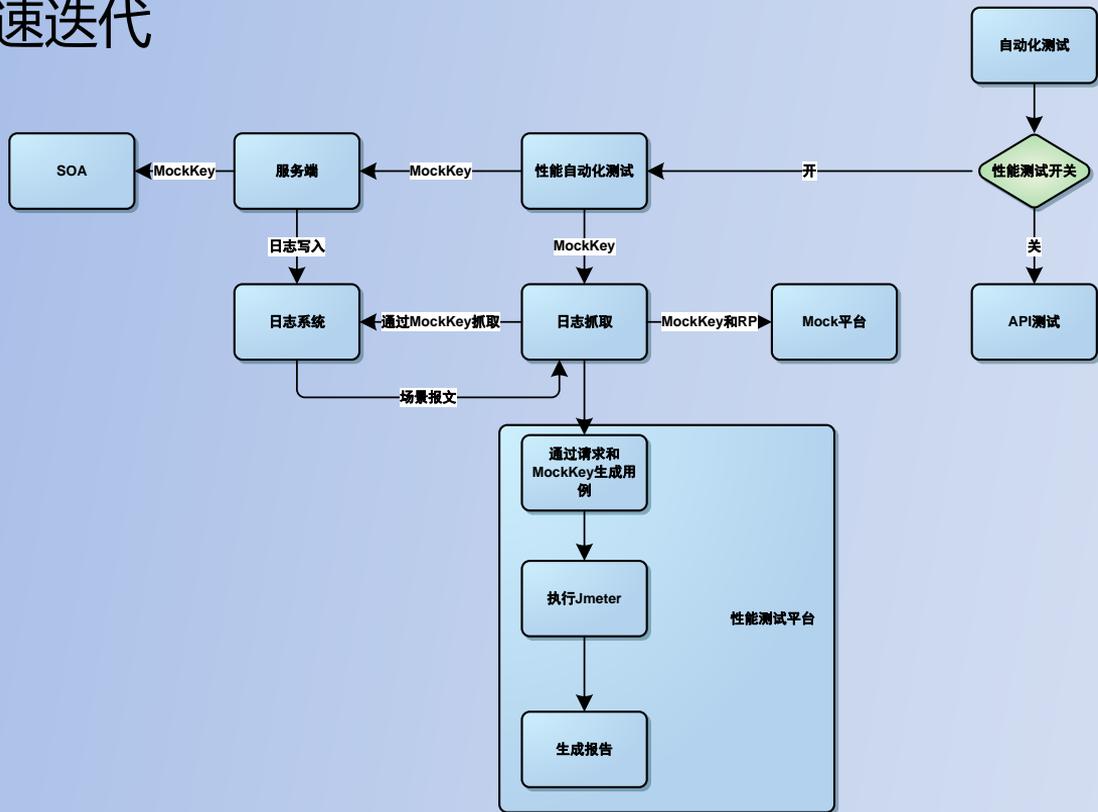
Type	Total	Status	Count
<span style="color: red;">■</span> error	1,451	<a href="#">[:: show ::]</a>	1,417
<a href="#">[:: show ::]</a>		<a href="#">[:: show ::]</a>	13
			10
			5
			5
			1

5	"CurrentPosotion": {	<>	5	"CurrentPosotion": {
6	"PositionRemark": "市中心",	<>	6	"PositionRemark": "",
7	"CoordinateInfo": {	=	7	"CoordinateInfo": {
8	"CoordinateEType": "GD",	<>	8	"CoordinateEType": "GG",
9	"Latitude": 9.5133012,		9	"Latitude": "-1.0",
10	"Longitude": 99.9688636		10	"Longitude": "-1.0"
11	},	=	11	},
12	"Radius": 50,	<>	12	"Radius": "0",
13	"KeywordType": 0,	=	13	"KeywordType": 0,
14	"KeywordValue": 0,		14	"KeywordValue": 0,
15	"KeywordFlagBitMap": 0		15	"KeywordFlagBitMap": 0
16	},		16	},

3

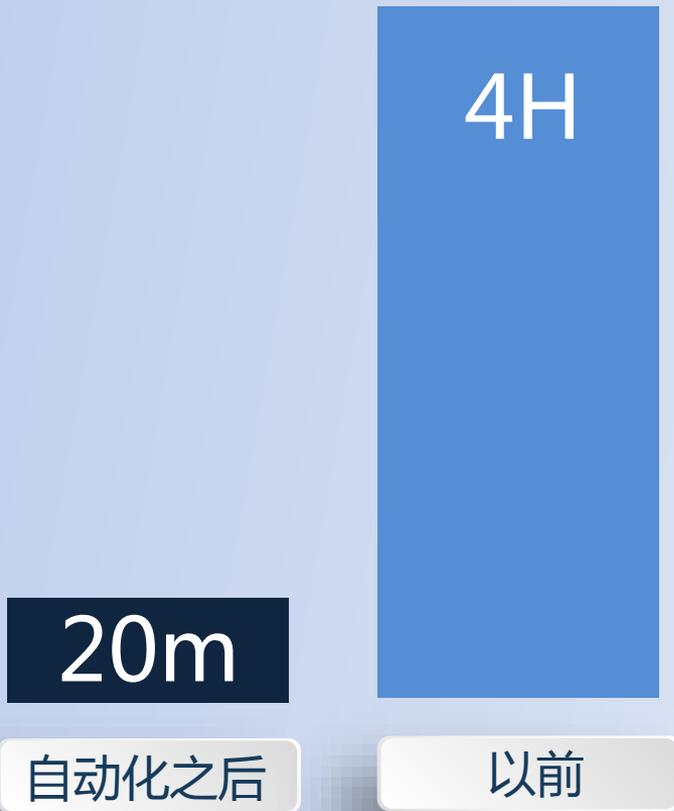
功能/性能  
自动化交互

我们在自动化测试框架中设计了一个小开关，以小博大，应对互联网的快速迭代

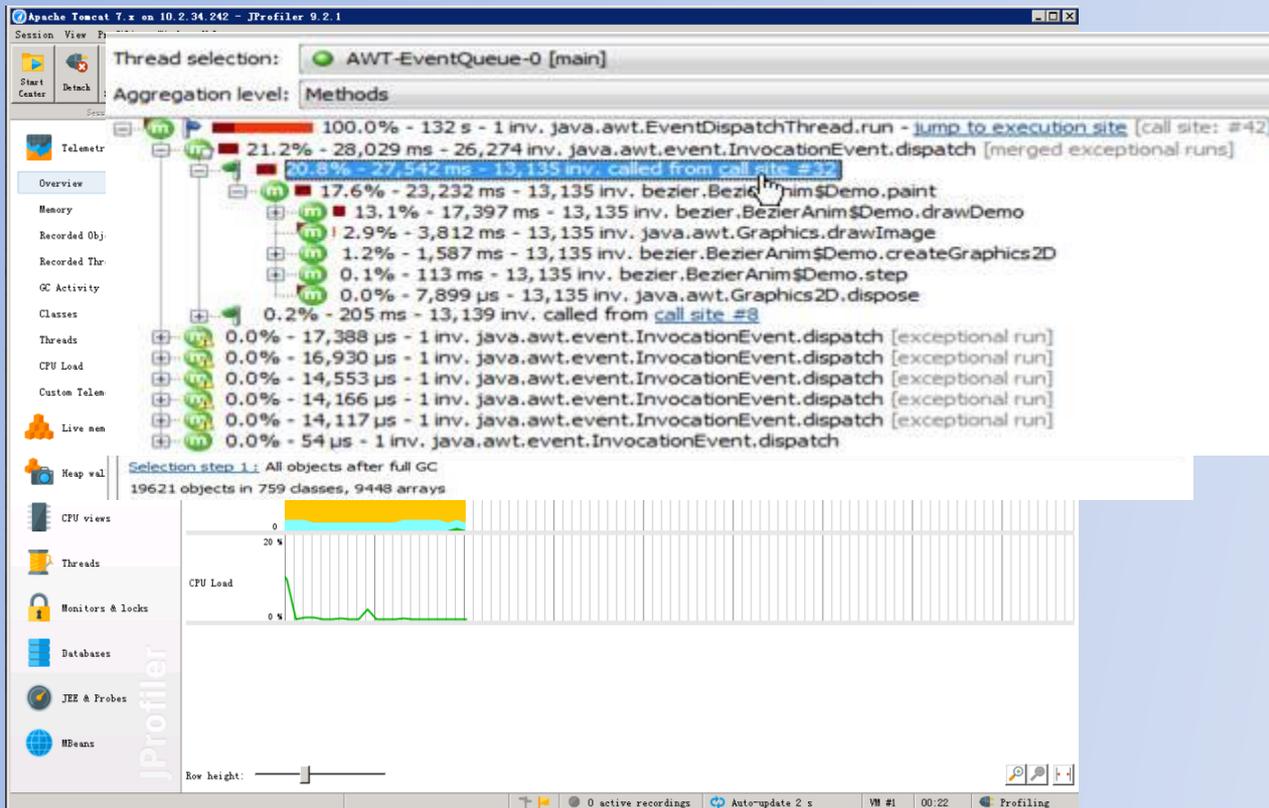


### 简化的性能测试

将【自动构造场景->生成执行脚本->执行->报告分析】自动结合在一起，人工干预时间大幅度缩小



## 性能测试结果的分析



## 流量回放及对比

将【智能抓取数据->分离读写->智能回放】结合在一起，智能增加自动化及性能场景，让发布风险大幅缩小

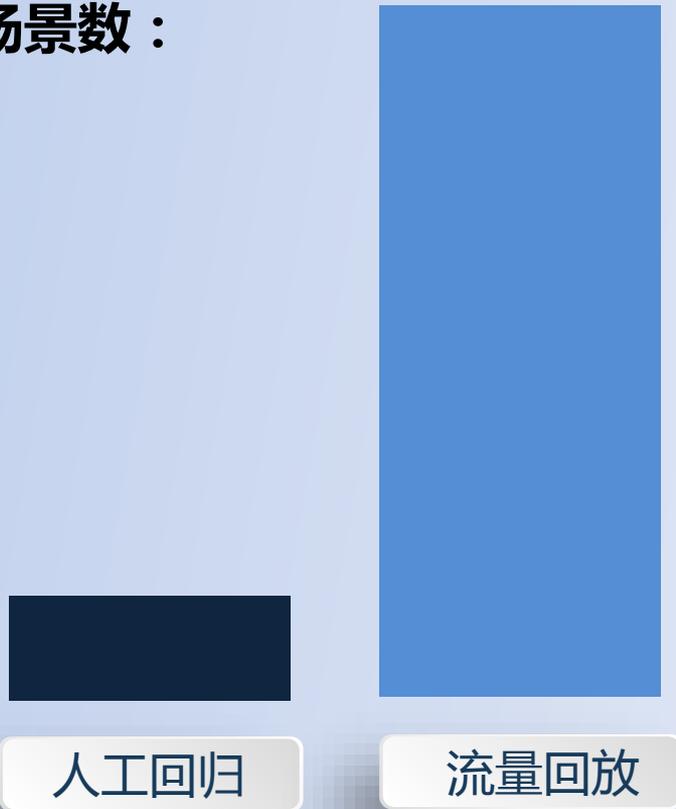
### 带来的好处：

大幅增加性能测试和功能回归的场景

对比线上版本，保证发布版本的稳定性

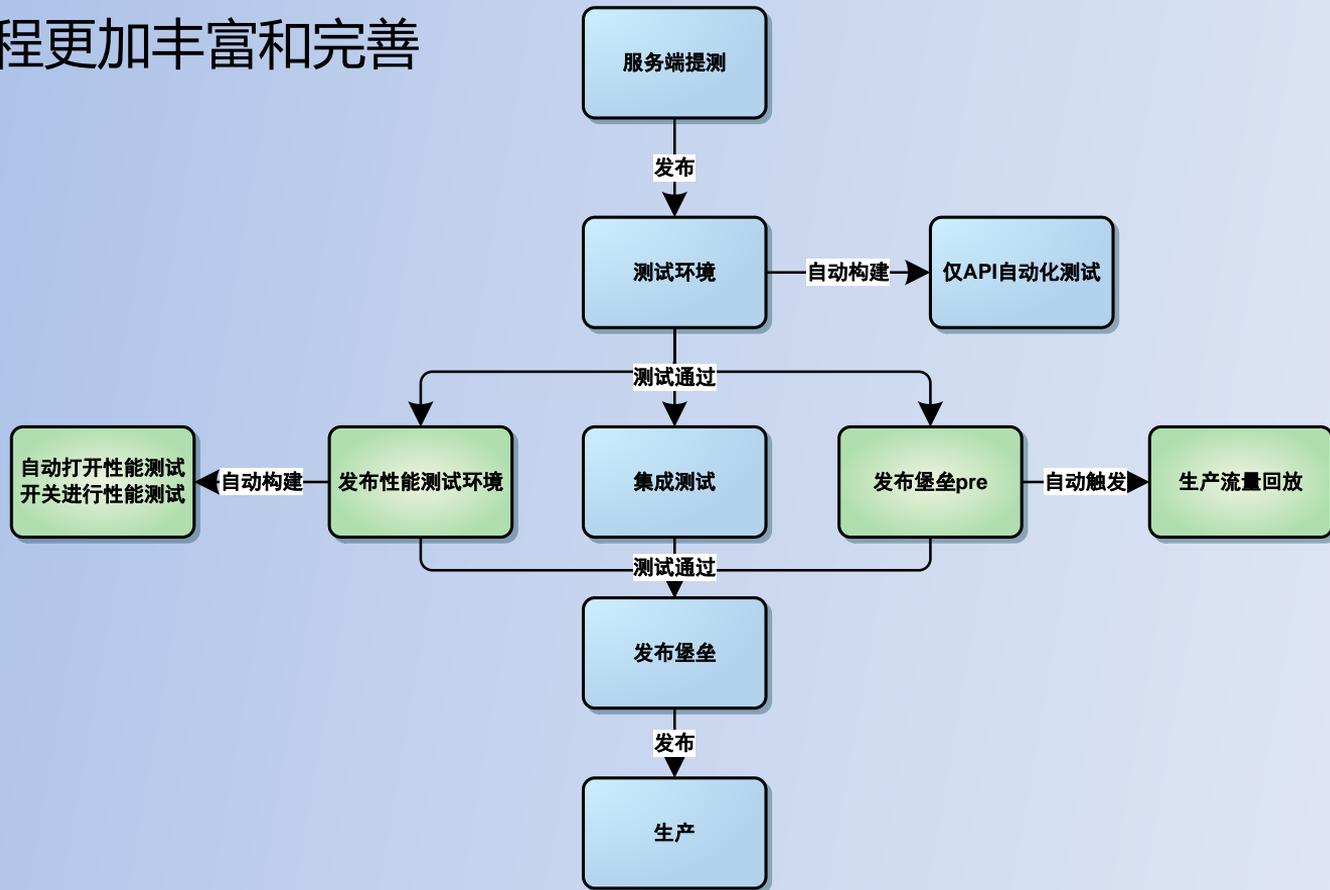
减少生产退回的次数

## 流量及场景数：



# 完整的API测试链路

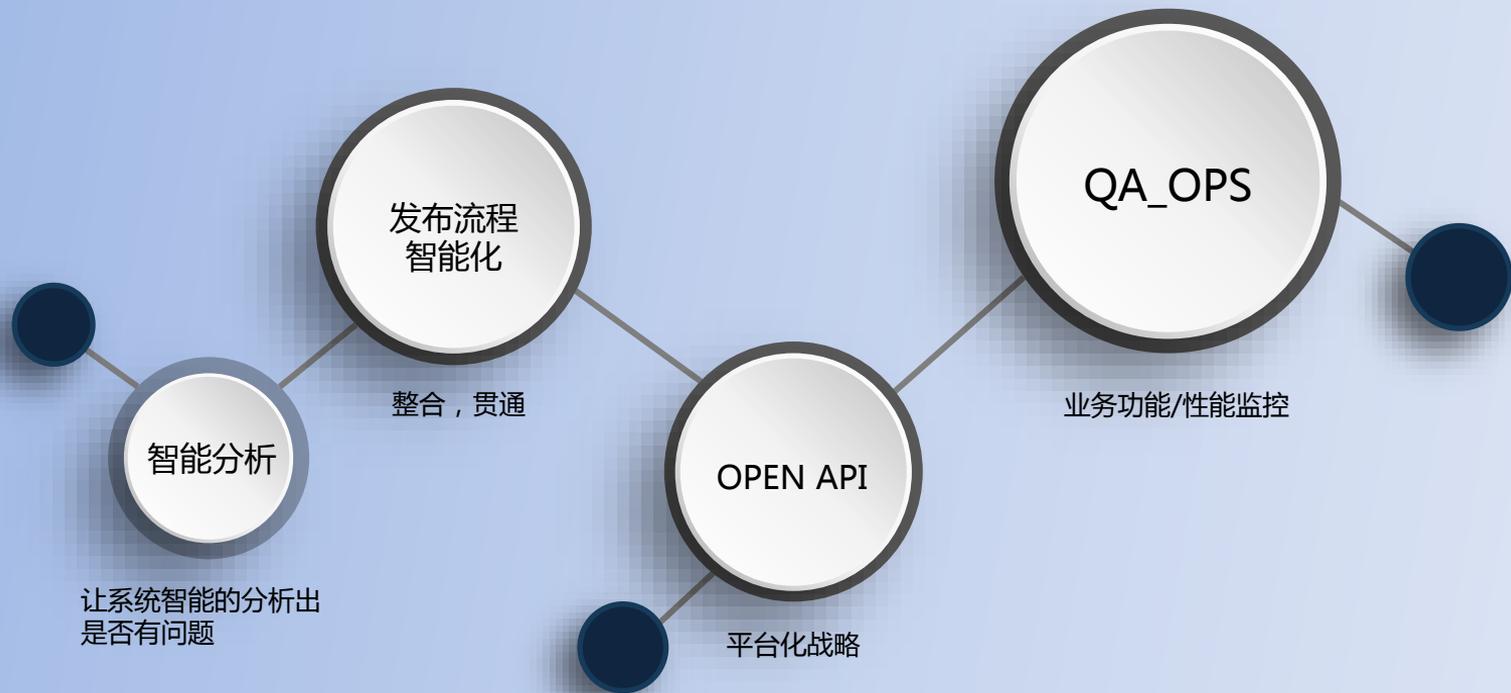
于是自动化流程更加丰富和完善



4

未来  
我们需要做的更好

# ● 将要实现





谢谢大家

