

# 全链路压测在滴滴的实践

分享人：杨明川/叶杰

2017-04-18







促进软件开发领域知识与创新的传播



关注InfoQ官方信息  
及时获取QCon软件开发者  
大会演讲视频信息



扫码，获取限时优惠



全球架构师峰会 2017 [深圳站]

2017年7月7-8日 深圳·华侨城洲际酒店

咨询热线: 010-89880682



全球软件开发大会 [上海站]

2017年10月19-21日

咨询热线: 010-64738142



## Agenda

- 为什么要做全链路压测？
- 滴滴如何做全链路压测
- 滴滴做全链路压测需要解决的问题
- 滴滴的解决方案
- 全链路压测的思考

# 为什么要做全链路压测

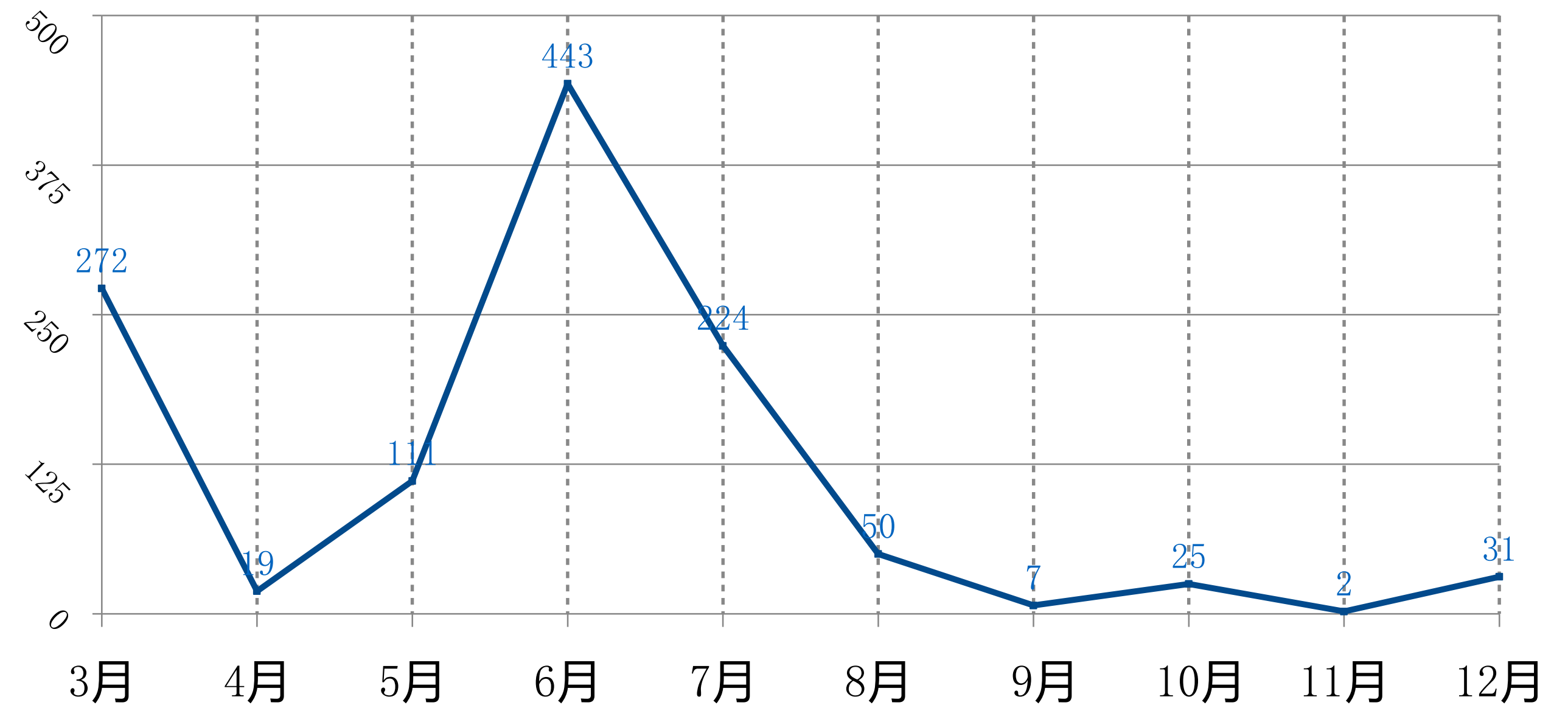
## • 背景

- 业务增长速度快，系统性风险增大
- 新模块不断上线，基础设施挑战严峻
- 架构快速迭代期，风险可控度降低

## • 达成的目标

- 摸清生产环境瓶颈点
- 提升各节点容量规划的准确性，降低成本
- 推动全链路架构review, 确保各环节的可靠性

2016年星辰花全平台事故时长月度曲线





# 滴滴如何做全链路压测

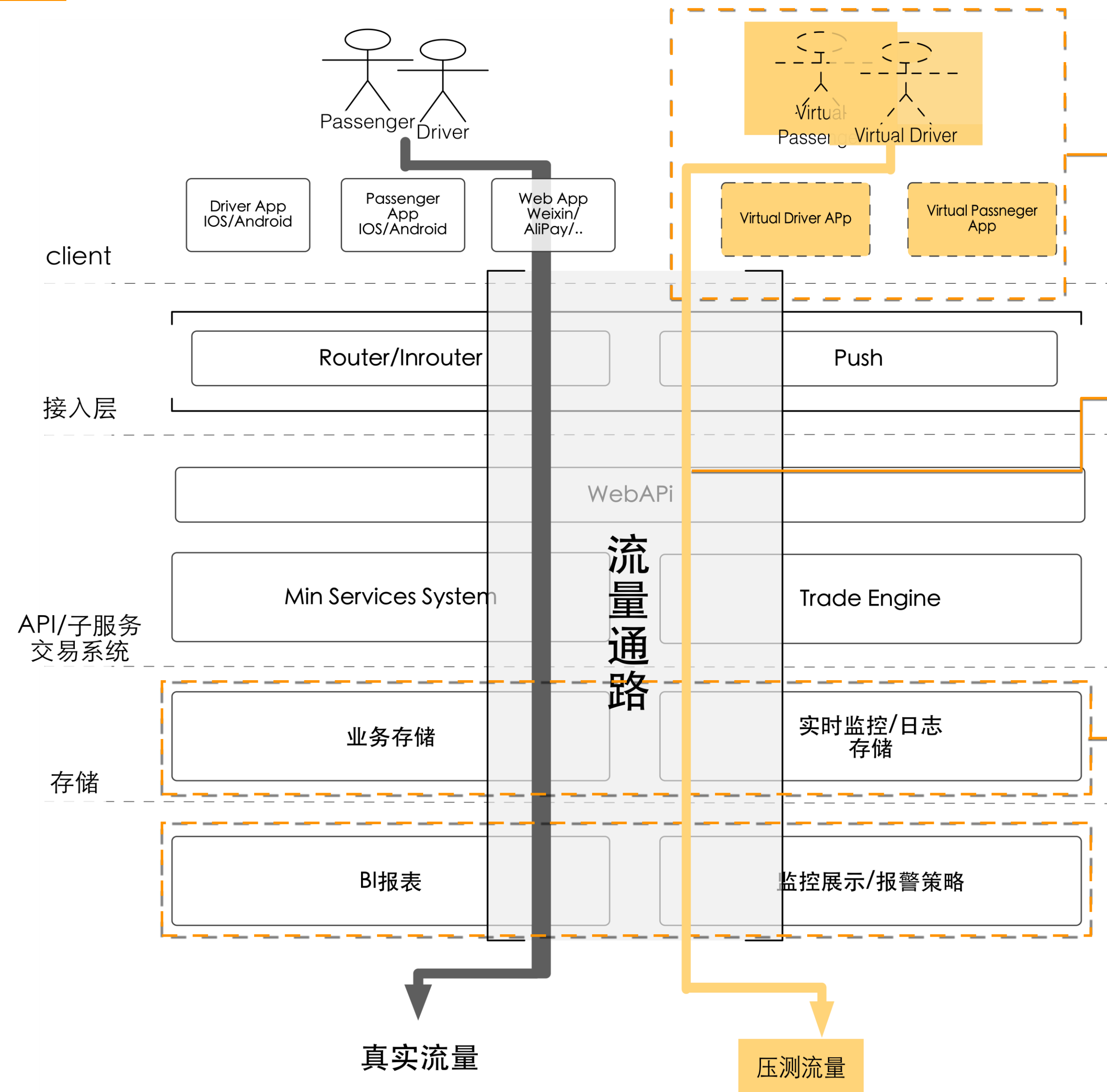
全链路压测的**全**是关键：**全流程**、**全平台**

“**虚拟国家**内，使用**虚拟乘客/司机**在**完整的线上环境**完成**交易流程**”

---

- 为什么是在虚拟国家内？
- 为什么要用虚拟乘客/司机重新完成交易，而不是历史交易回放？
- 为什么一定是完整的线上环境？

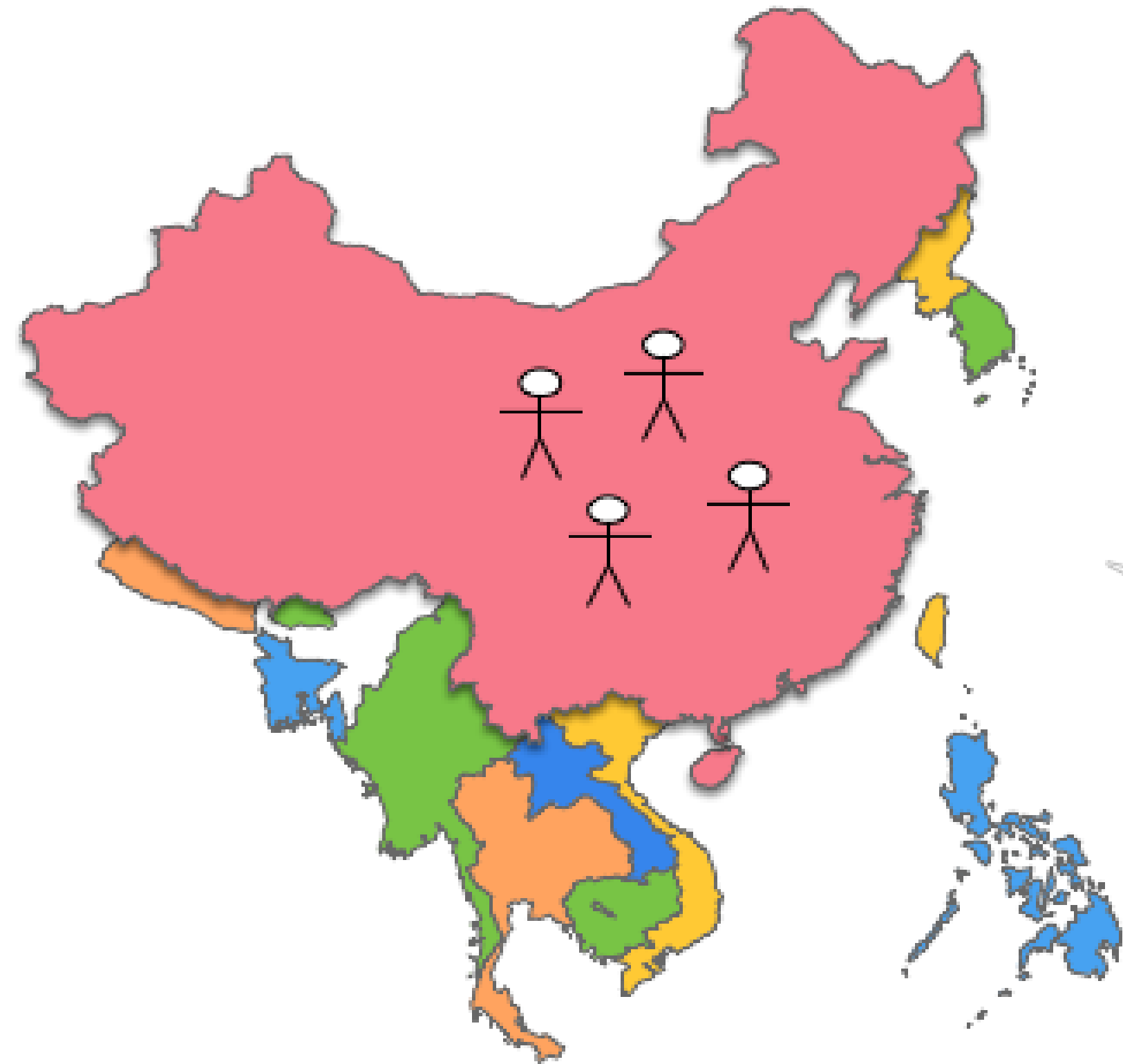
# 滴滴做全链路压测需要解决的问题



## 需要解决的问题？

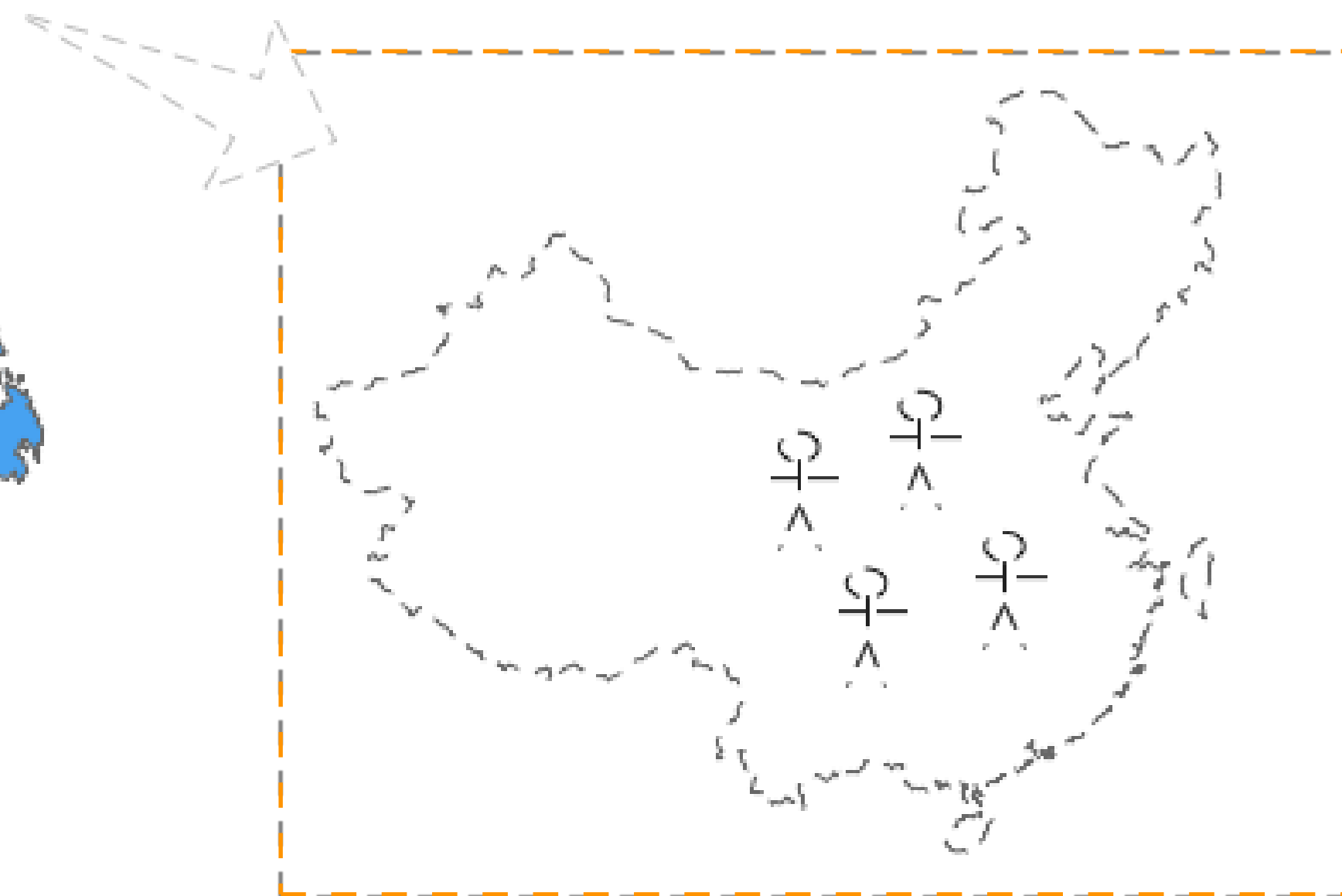
- 虚拟数据构造：虚拟乘客/司机，虚拟场景，虚拟App；
- 流量标记：http/thrift协议，mysql/codis/rockstable协议；
- 存储隔离/清理：持久化存储，缓存；

# 滴滴的解决方案—虚拟数据构造



## 虚拟数据构造

- **虚拟场景：**完全虚拟的国家，对所有地图数据同比偏移；
- **虚拟乘客/司机：**基础信息保持，关键信息偏移；
- **虚拟App：**历史数据初始化，依赖路径规划移动；





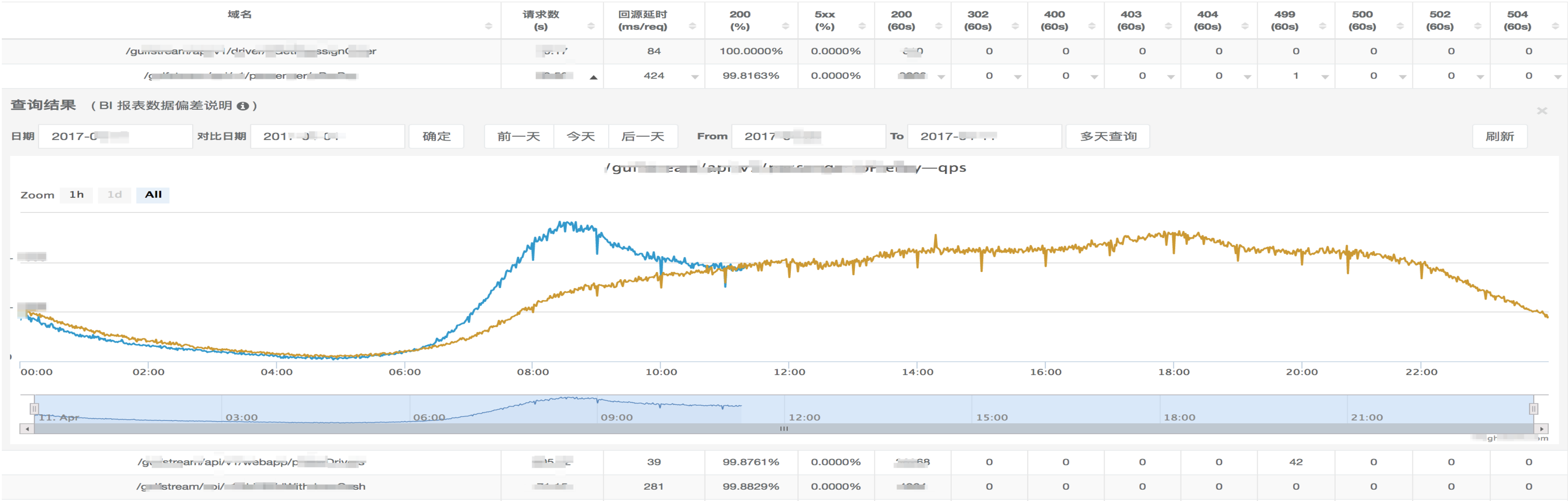
# 滴滴的解决方案—虚拟环境评估

“虚拟环境的真实性，用关键指标及接口同比放大的方式评估”

## 真实环境关键指标采样

订单	指标	呼叫量	未成交量	播单量	成交量	行程中量	订单分布城市数	订单平均到几个司机	成交订单几个司机抢	平均成交率
	值	x ( 峰值 )	Y ( 峰值 )	Zw(峰值)	H(峰值)					0.J
司机	指标	Push在线量	抢单量	叉单量	成交后取消量	支付发起量	现金支付量	听抢率	在线司机量	
	值	Kw(峰值)	L ( 峰值 )	U(峰值)	O ( 峰值 )	M ( 峰值 )		0.0N ( 平均 )	Ww(峰值)	
乘客	指标	调价接受率	成交前取消量	成交后取消量	成交前平均等待时间					
	值	0.A ( 平均 )	B ( 峰值 )	C ( 峰值 )						

## 各接口QPS采样



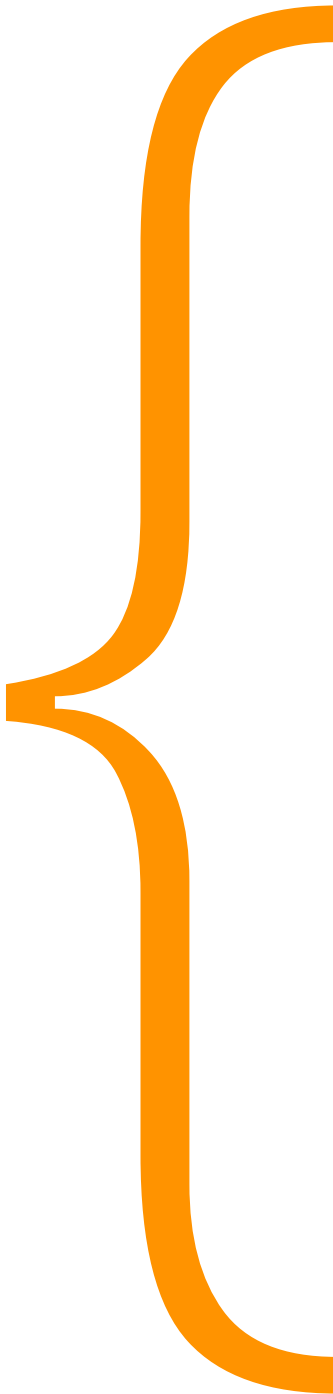




# 滴滴的解决方案—压测流量标记

## 压测流量标识方案

流量如何标识？



- ① 使用现有业务id
- ② 走trace通路，利用traceId编码
- ③ 走trace通路，增加字段标识流量来源

协议	举例	方案①	方案②	方案③
http	APP端到服务器调用。 服务器各模块（API）调用。	优：无需额外工作  劣：接口中不一定包含这些id，各模块流量识别需要进行id间映射转化	优：有统一准则识别流量。  劣：需要进行traceid反解。跨模块通信协议上需要注入traceid。	优：有统一标识识别流量。  劣：跨模块通信协议上需要注入结构体（trace信息，请求tag等）
thrift	服务器跨模块RPC调用。			
基于tcp/socket的协议	mysql redis memcache等开源服务。 自研服务（push）。	不能识别	mysql，redis。有标注需求可以考虑注释方式，或者是在协议代理层引入。 push等自研服务，header	

# 滴滴的解决方案—压测流量标记

## http通信

```
didi-header-rid: $traceId;  
didi-header-spanid: $spanId;  
didi-header--hint-code:  
$hintCode;  
didi-header-hint-content:  
'$hintContent';
```

## thrift通信

```
struct Trace {  
    1: required string logId;  
    2: required string caller;  
    3: optional string spanId;  
    4: optional string  
srcMethod  
    5: optional i64 hintCode;  
    6: optional string  
hintContent; }
```

## 部分业务流量识别

各业务id:  
cityid,  
passengerId,  
driverId,  
phone,  
orderId

## 长连接

- ①利用push包header结构体的logId传输
- ②传递(traceid + hintCode)。js被序列化后以二进制保存在header
- ③dispatch模块将tcp中header信息解出后。在转出的http请求的header中以key-value方式进行设置

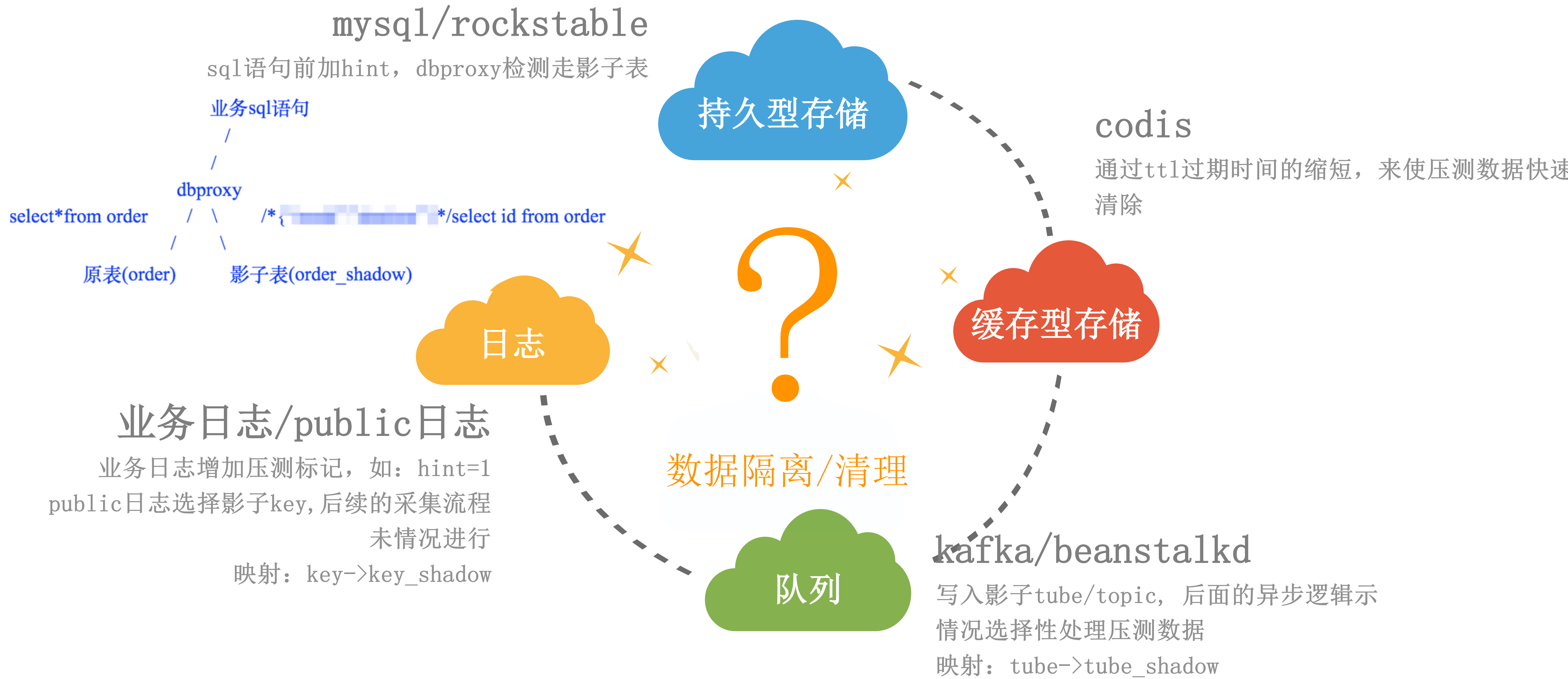
## 数据库

sql语句前置hint标记：

“流量标识方案：  
方案① + 方案 ③”



# 滴滴的解决方案—数据隔离及清理



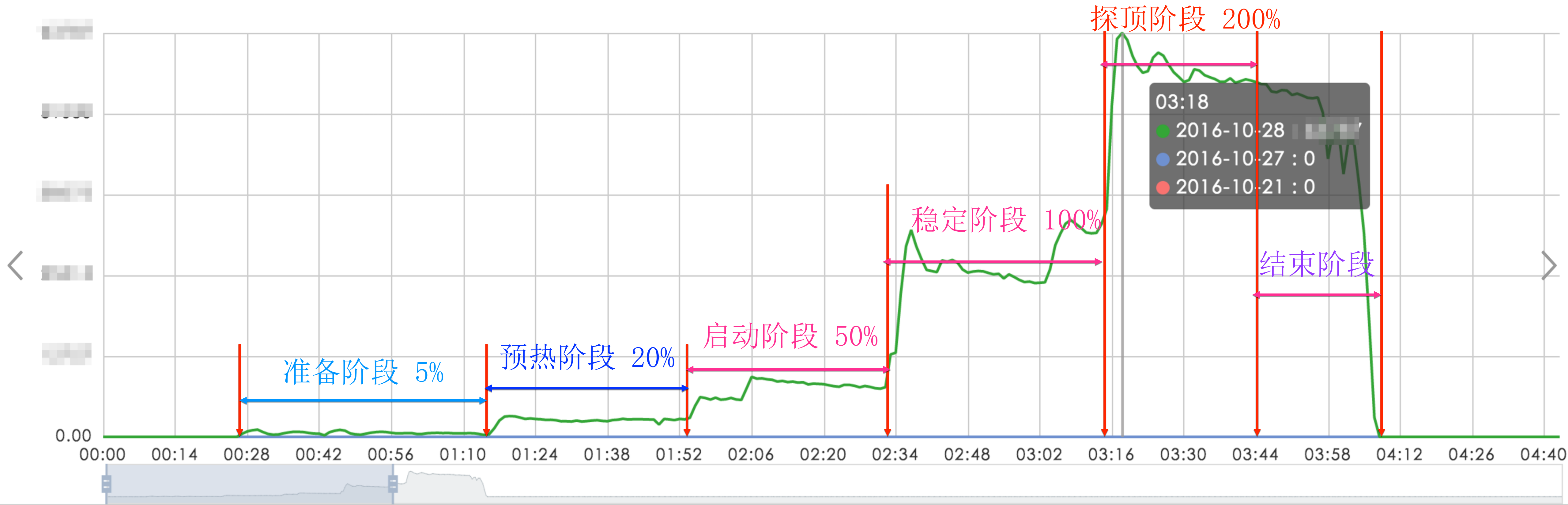




# 滴滴的解决方案—压测阶段

重要指标概况实时订单呼叫量

● 2016-10-28 ● 2016-10-27 ● 2016-10-21



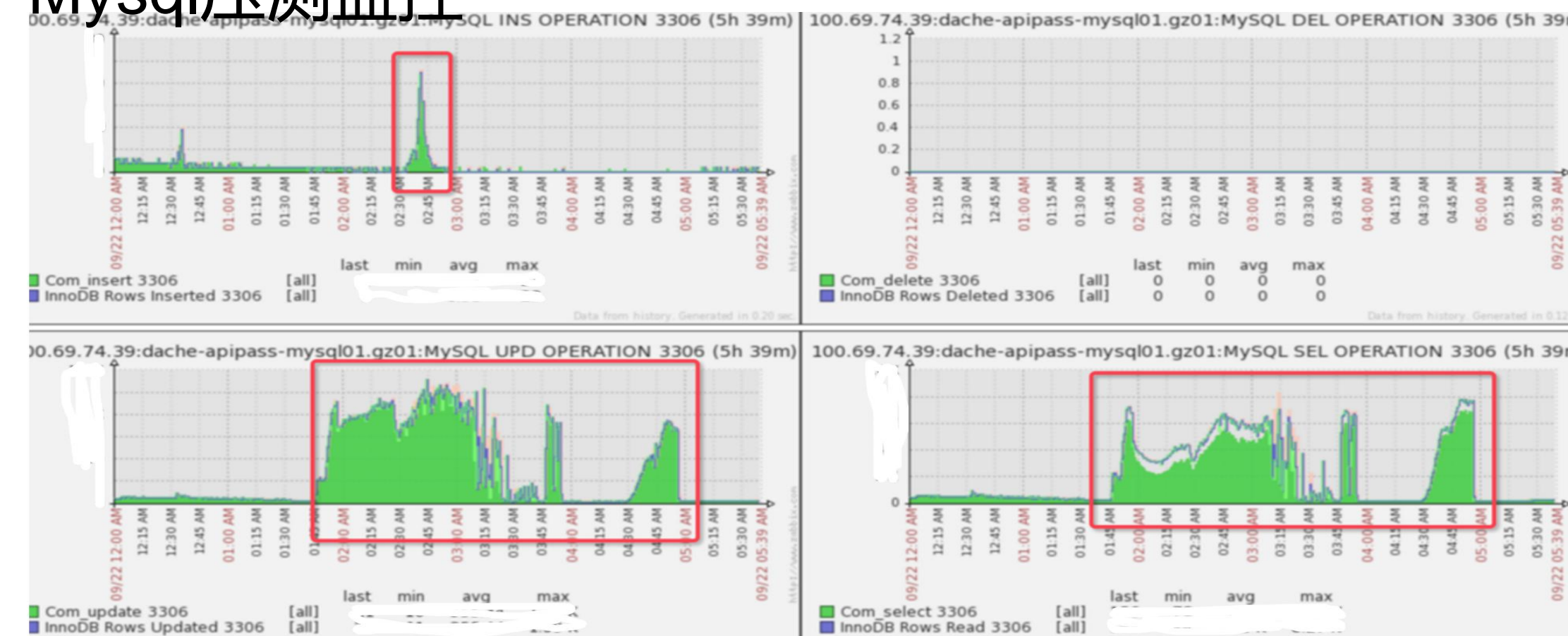
日期: 2016-10-28    最大值: 357    最小值:    平均值: 1554    总计: 1554

# 滴滴的解决方案—监控隔离

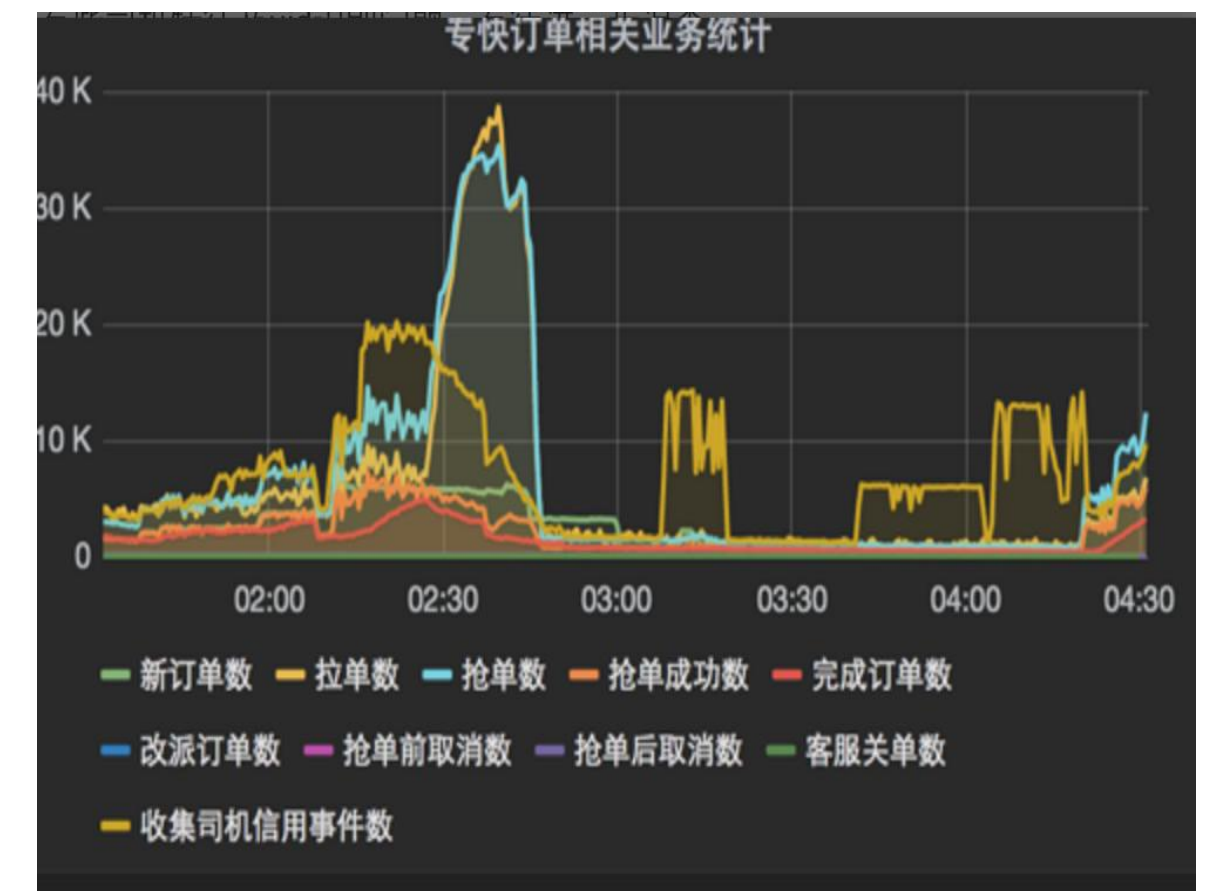
## 各接口压测监控



## Mysql压测监控



## 引擎自研监控



## 滴滴的解决方案—压测收获

### 完成全平台主业务线压测

- △ 出租车一期全链路
- △ 专快车及支付一期全链路压测
- △ 顺风车一期全链路压测
- △ 出/专快+支付/顺风车联合压测

- △ 出租车二期全链路压测
- △ 专快车及支付二期全链路压测
- △ 顺风车二期全链路压测

### 其它？

公交/代驾/地图/商业 等其它业务

60%

#### 链路中的隐患

- 长连接参数问题导致到达瓶颈
- sug资源不足，当时cpu idle到0
- 分单codis资源不足

40%

#### 附加价值

- 完整的、隔离的线上环境，支持策略算法等功能的仿真环境
- TraceID覆盖更多模块
- dbproxy支持多表的查询、写入改造方案



## 正在努力解决的问题

- ①流量通路搭建，对业务侵入！
- ②数据隔离需要业务逻辑处理！
- ③数据通路维护成本高昂！

如何支持日常压测？

## Disf (didi service framework)

- 解决服务间的naming问题
- 架构RPC通路，解决流量的底层标识透传问题
- 收敛子服务调用SDK, 规范服务调用，进步收敛RPC通信

平台化压测

What's the  
next?

与卓越的人一起共事，欢迎自荐：  
[yangmingchuan@didichuxing.com](mailto:yangmingchuan@didichuxing.com)







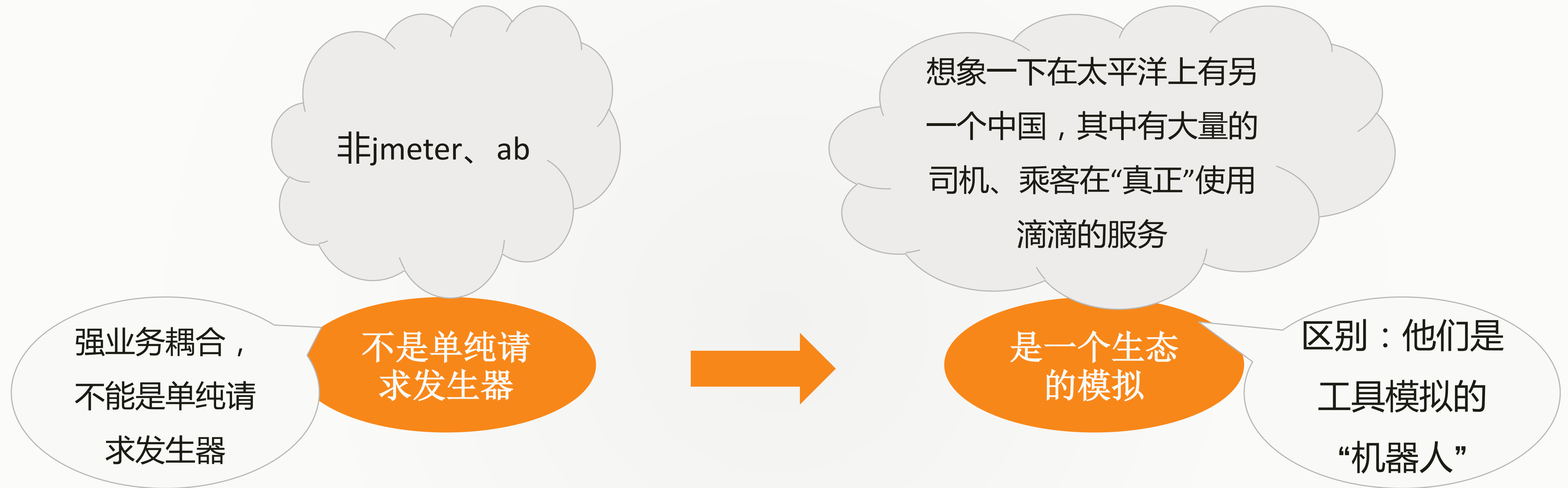
# 全链路压测工具端介绍

分享人：叶杰  
yejie@didichuxing.com



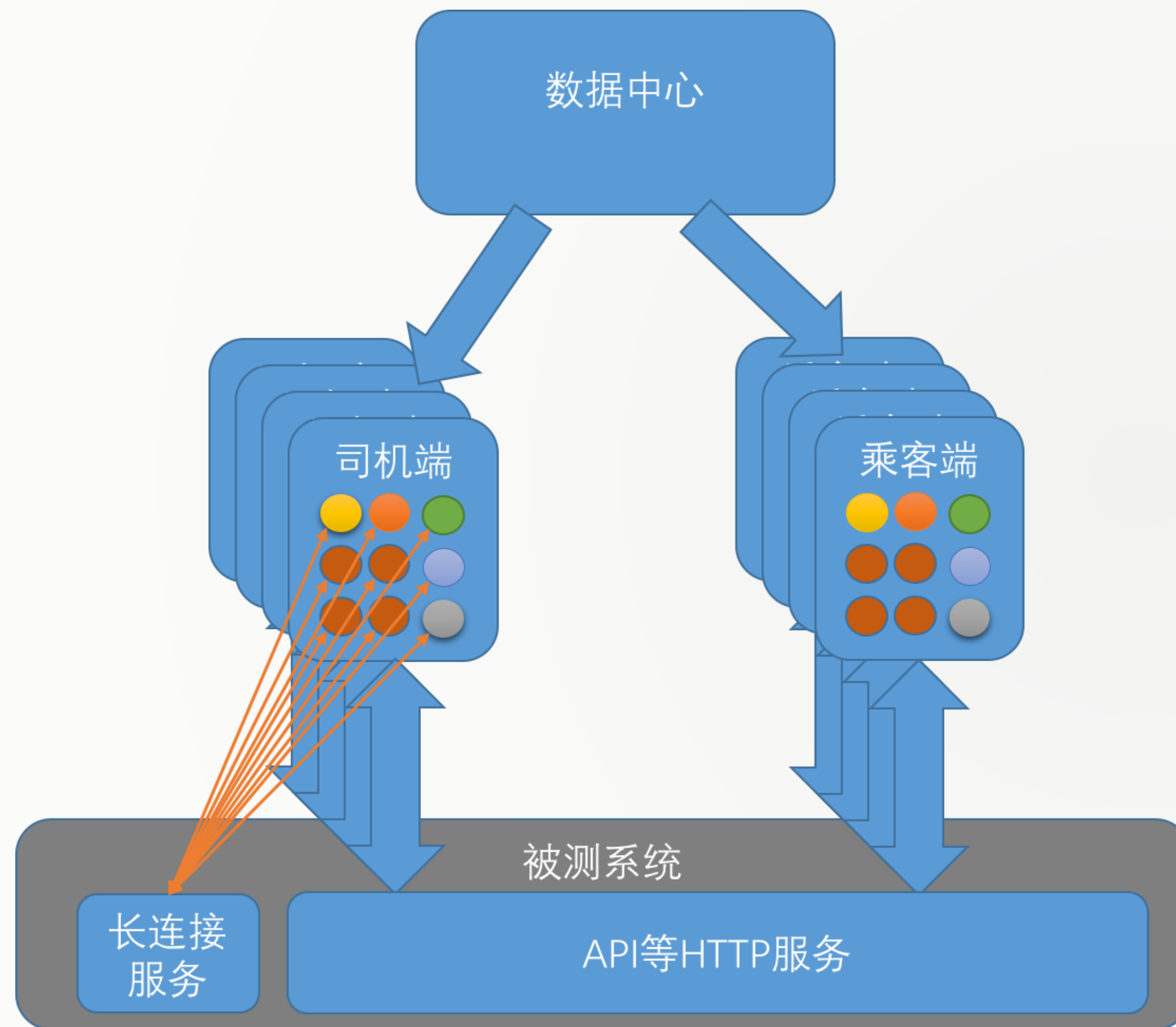


# 选择什么模拟方式？



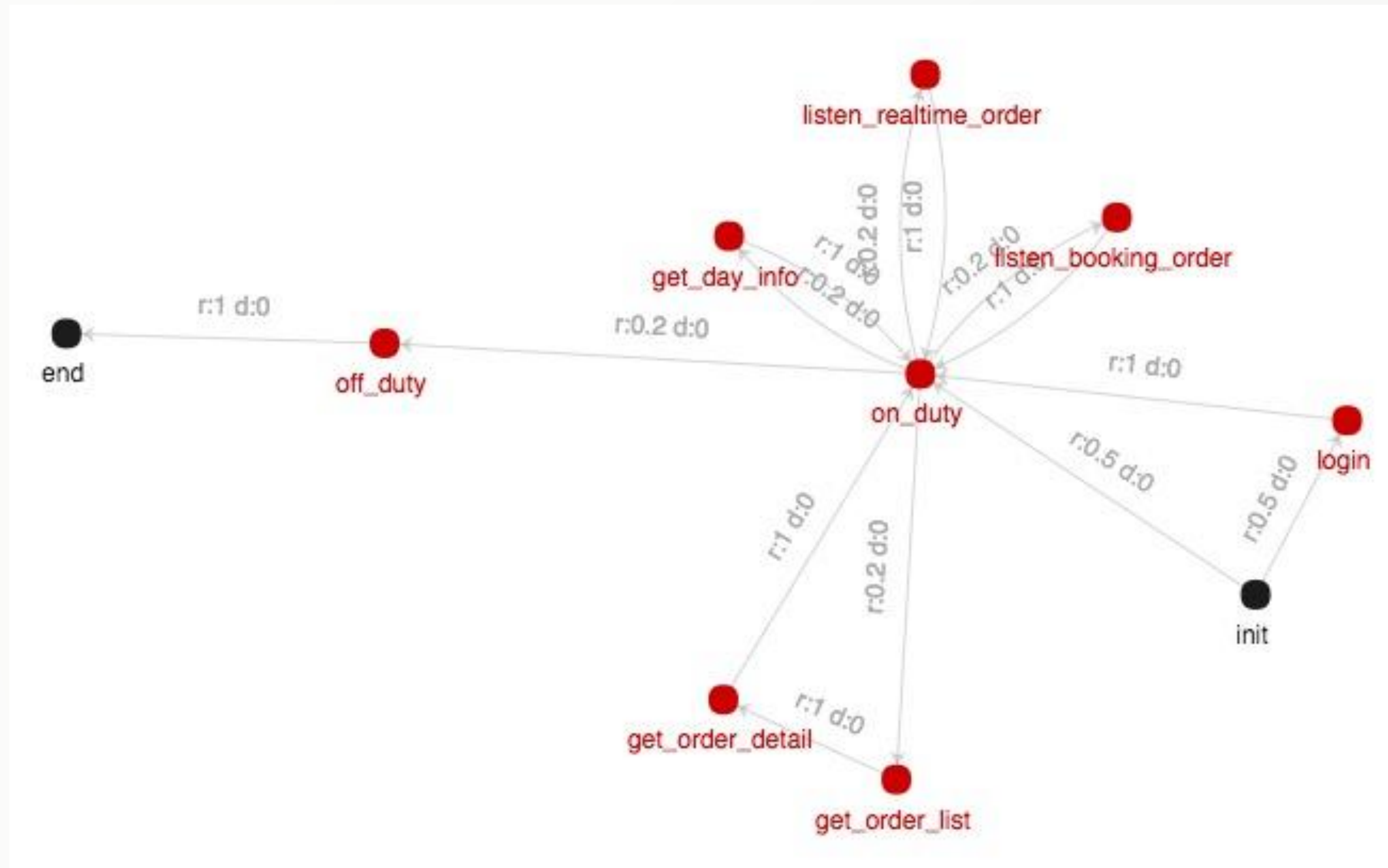
- 压测工具模拟千千万万个“机器人”，形成一个生态
- 压测流量是这个生态的一个宏观外在产出

## 宏观上长什么样？



- 数据中心：控制机器人的**数量投放、行为配置、位置数据等**
- 单个进程模拟**N**个机器人
- 线上服务接口分两大类：**长连接、http服务**
- 每个机器人：**独立**的长连接、**共享**的http长连接

# 单个“机器人”（乘客/司机）长什么样？



□ 机器人的“**大脑**”是状态流转图

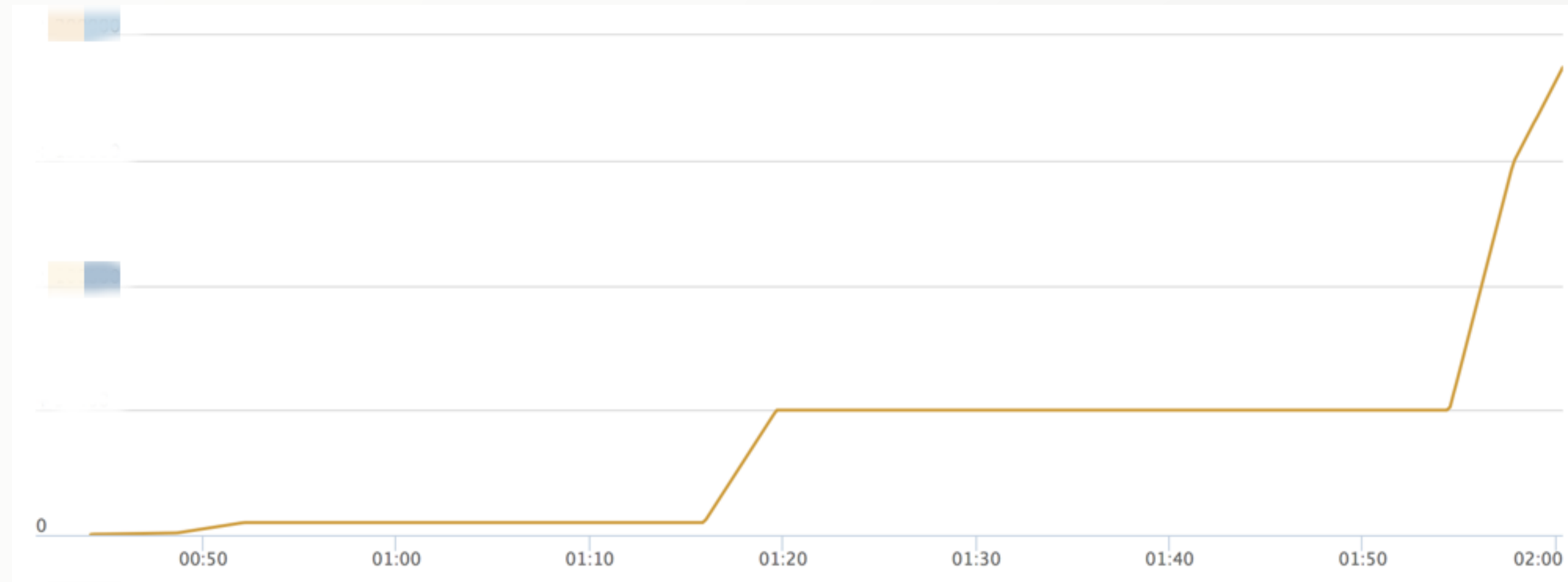
□ 决定了其按比例做什么、什么时间做

□ “机器人”的行为决定了整个生态的宏观流量

□ 可**动态修改**：跳转路径、比例、延时、节点  
流量上限

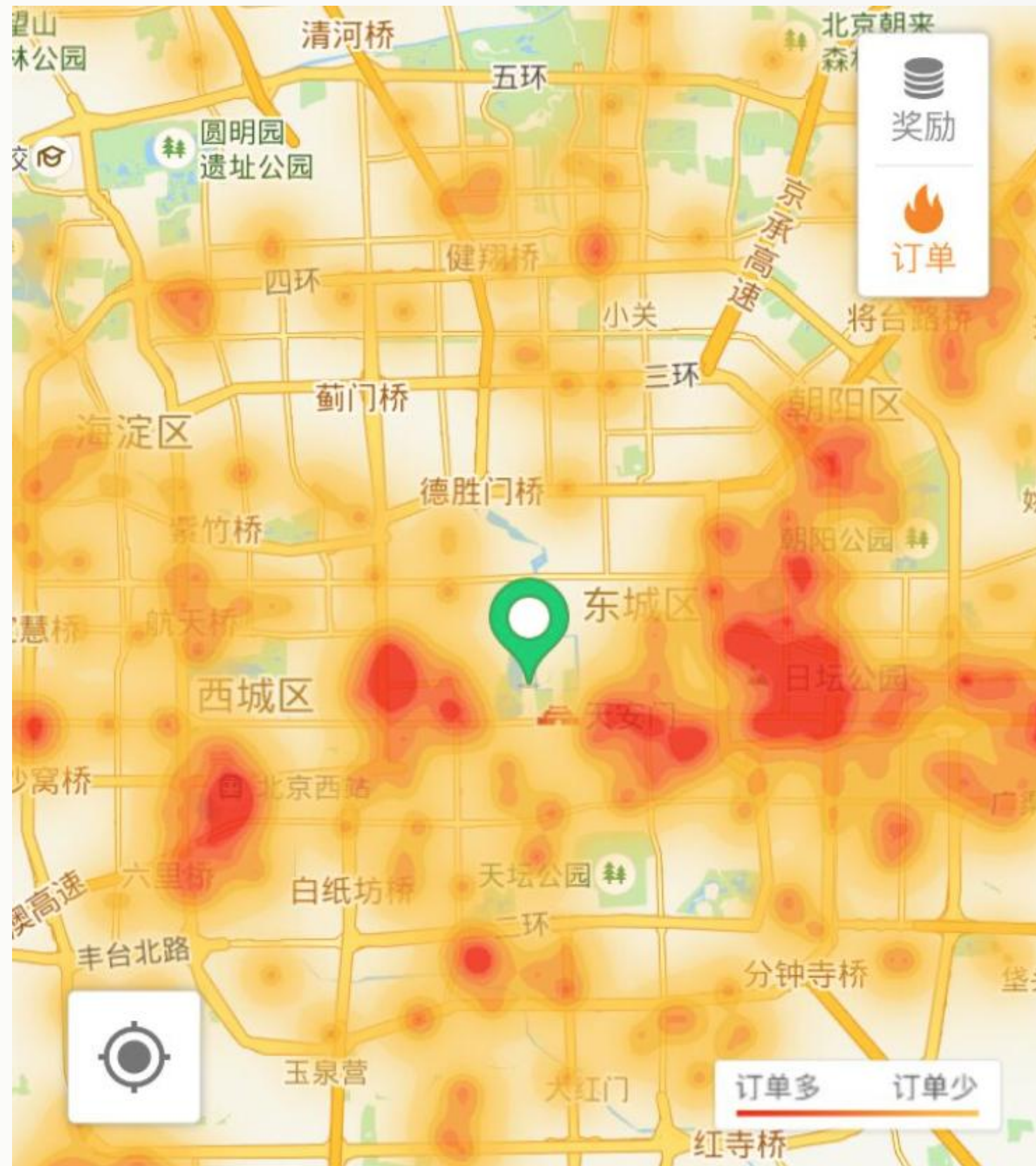


# 机器人是怎样投放的？



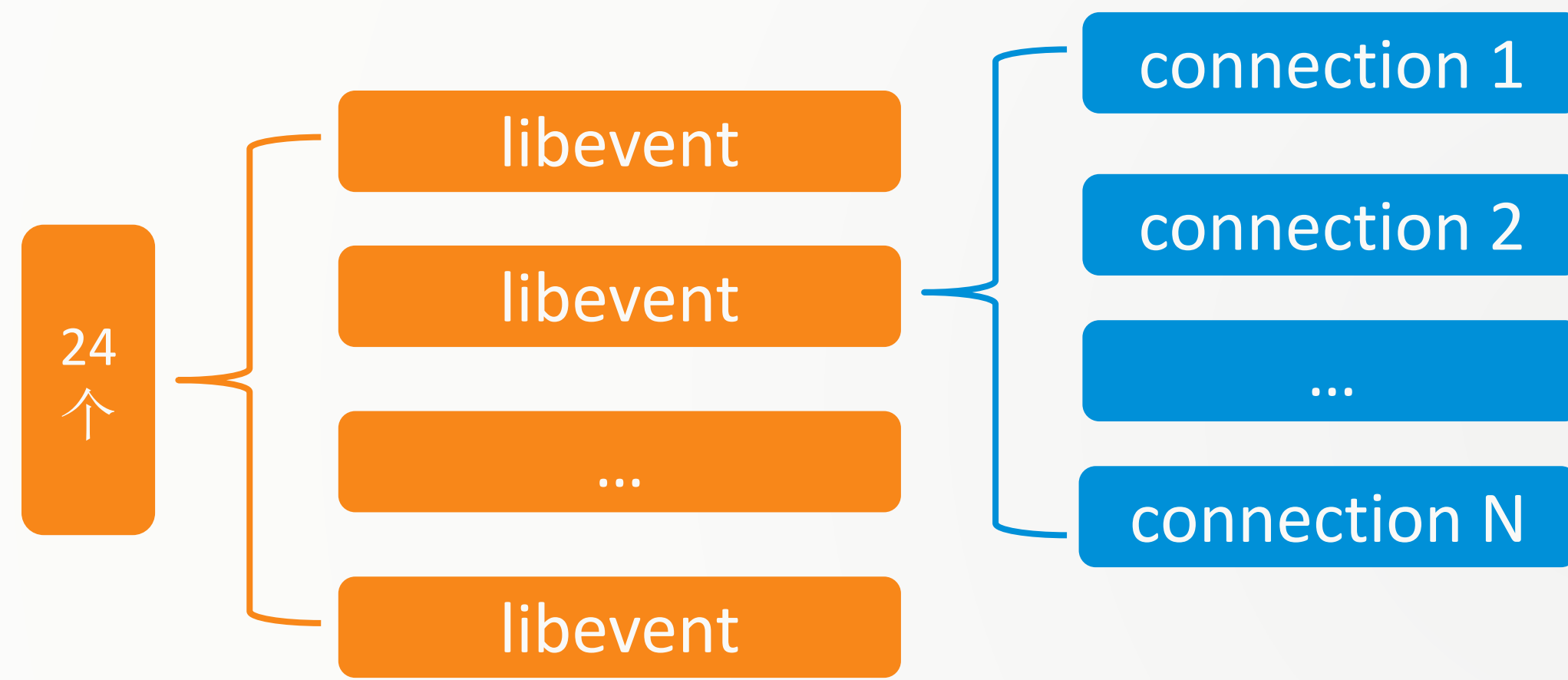
- 提前在数据中心准备好用户池
- 投放量及速度可**动态控制**：准备、预热、启动、稳定、探顶、结束
- 用户结束后回收再利用

# 司机和订单位位置是怎样分布的？



- 司机按照线上高峰期**切片位置**投放，并随机选点按路径规划运行
- 订单按照高峰期**历史订单数据**投放
- 接单行为和线上正常行为没有区别，如果用 **app**发单（经过特殊处理进入压测渠道），**会真的看到有个机器人师傅来接您**

# 关键技术细节：建立海量的长连接



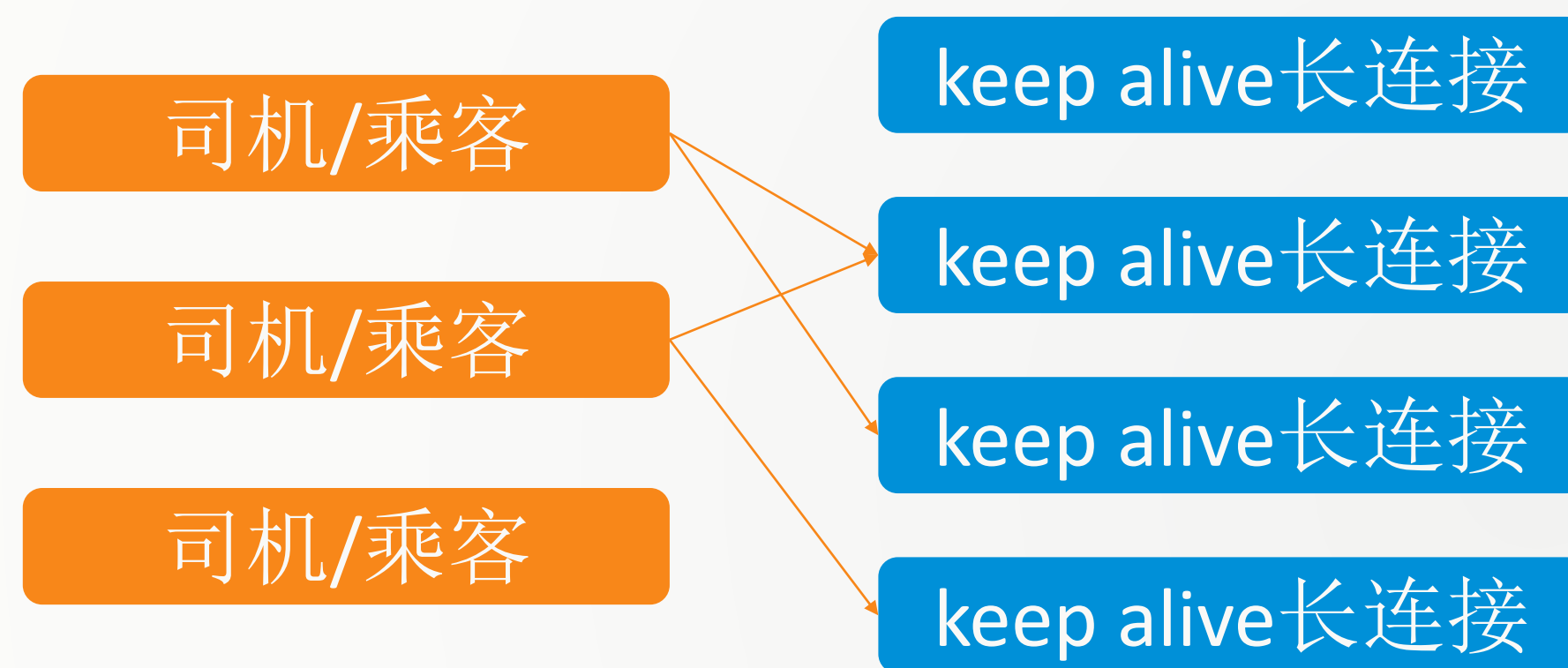
## □ 两级扩容：

- 单**libevent**实例管理单**connection** -> 单实例管理多个**connection**
- **libevent**实例数1个 -> 24个（48个CPU的一半）

□ **ip**扩容：一个**ip** -> 5个**ip**（同一个网卡），突破当前**linux**版本**ipv4**的**65536**端口上限

□ 达到单机**20w**长连接级别

## 关键技术细节：海量http请求



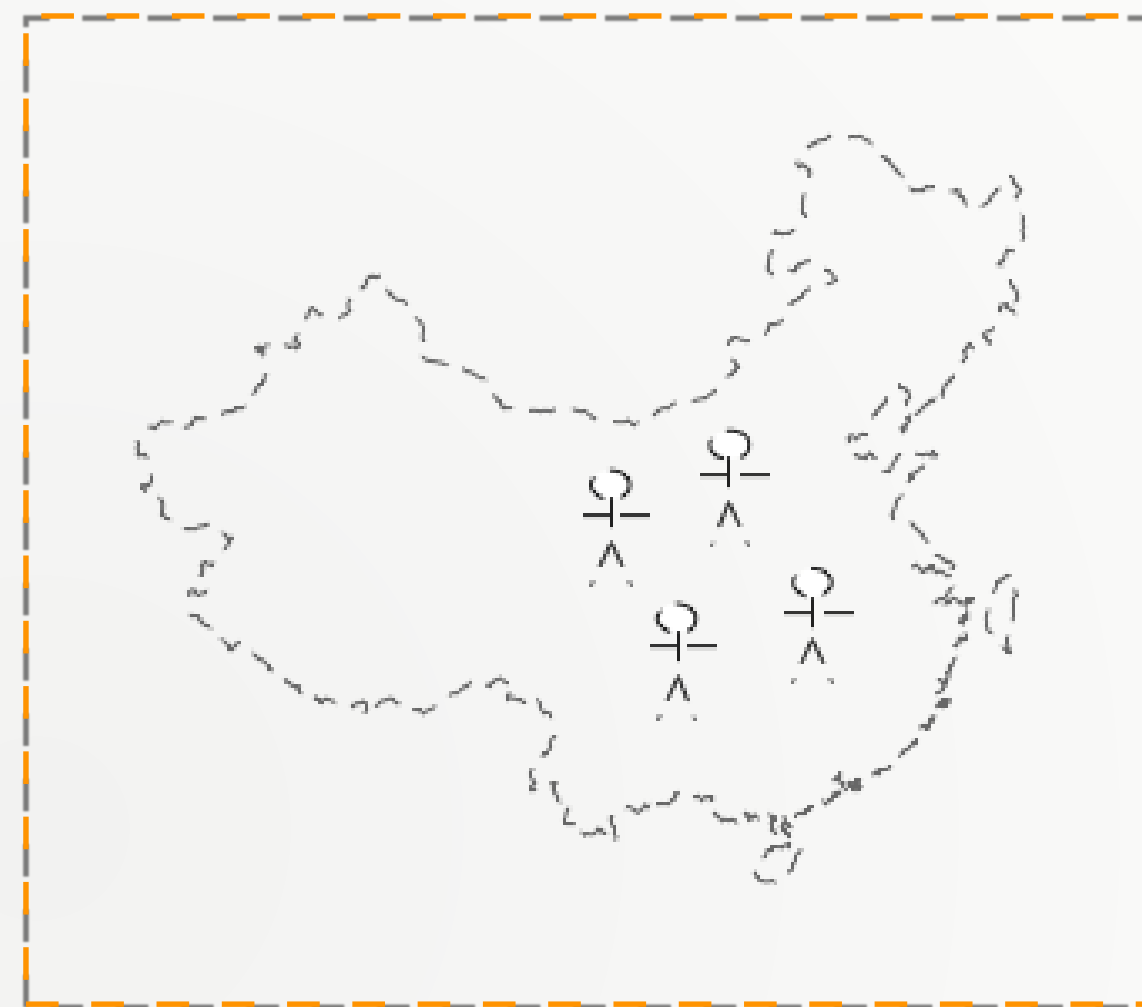
- ❑ **keep alive**长连接，降低创建链接的资源消耗
- ❑ 异步发送，防止线程**block**
- ❑ 达到单机**2w qps**级别



# 关键技术细节—状态机事件驱动模型



## 总结回顾



- 数据、流量、存储的隔离，使线上压测成为可能
- 千千万万司乘的模拟，在虚拟的中国形成一个生态，使线上压测实际落地

# THANK YOU



北京滴滴无限科技发展有限公司  
北京市海淀区东北旺路8号院尚东·数字山谷B1号楼

