

# 手Q性能专项实践2.0

从小数据到大数据

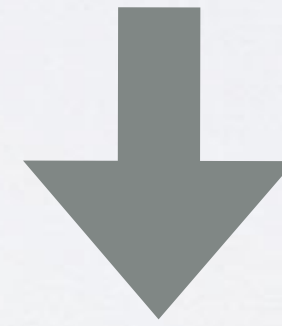
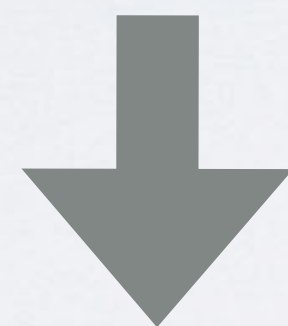
腾讯SNG社交产品质量部 victorhuang



# 背景：说一下1.0

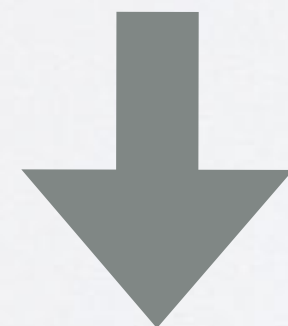
- 手Q专项测试最佳实践（QCON 2015）

阶段1: 核心专项自动化测试    自动化执行+监控



阶段2: 性能分析云

自动化分析



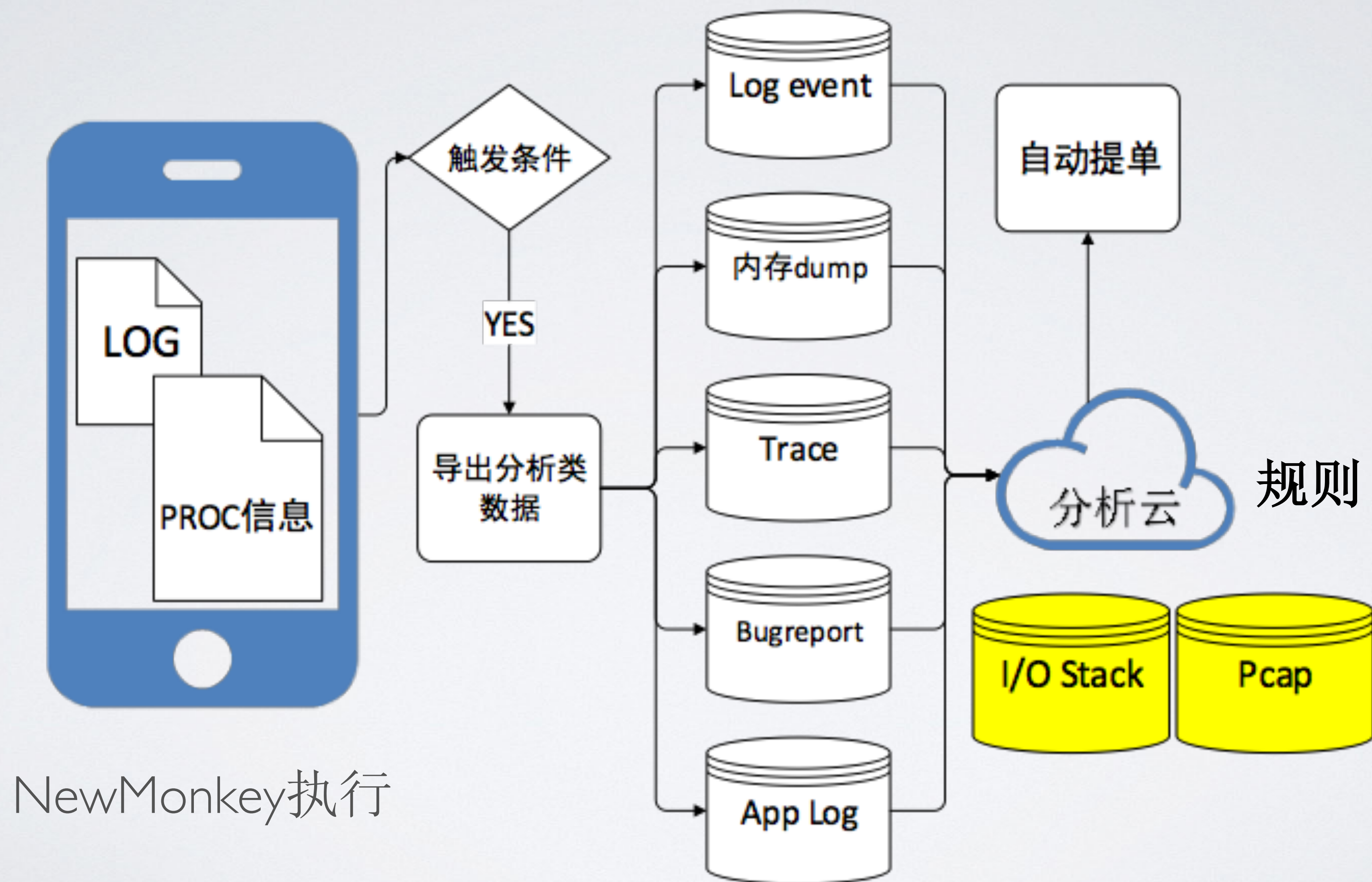
阶段3: 动态检查

自动化分析+自动化执行

} 自动化



# 动态检查



# 关键点

- 如何让随机测试尽可能遍历功能？
- 分析的规则都有什么？如何制定的？
- 乃至缺陷如何去重？最终效果如何？



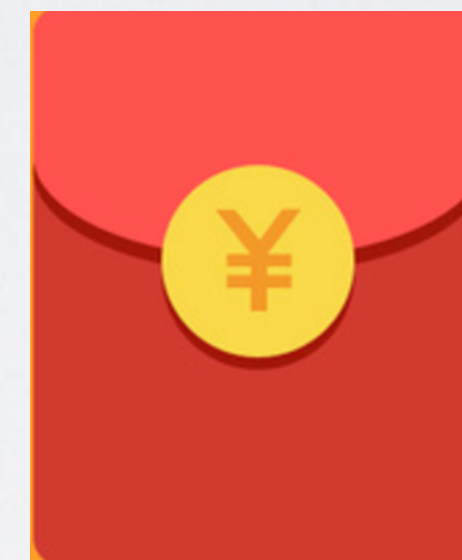
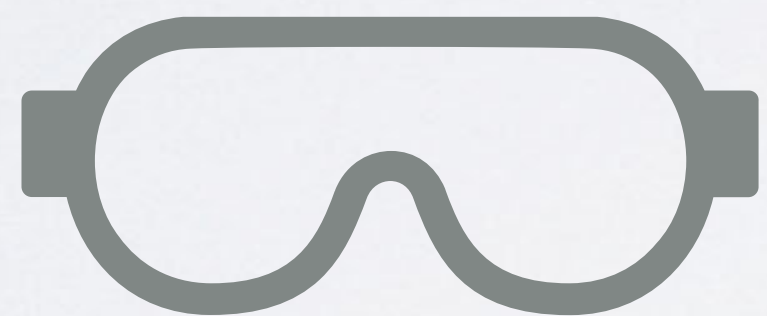
《手Q专项测试最佳实践》

《你从来没有想过的新 Monkey 测试》



# 《ANDROID性能专项实战》

# 问题来了！实验室的代价

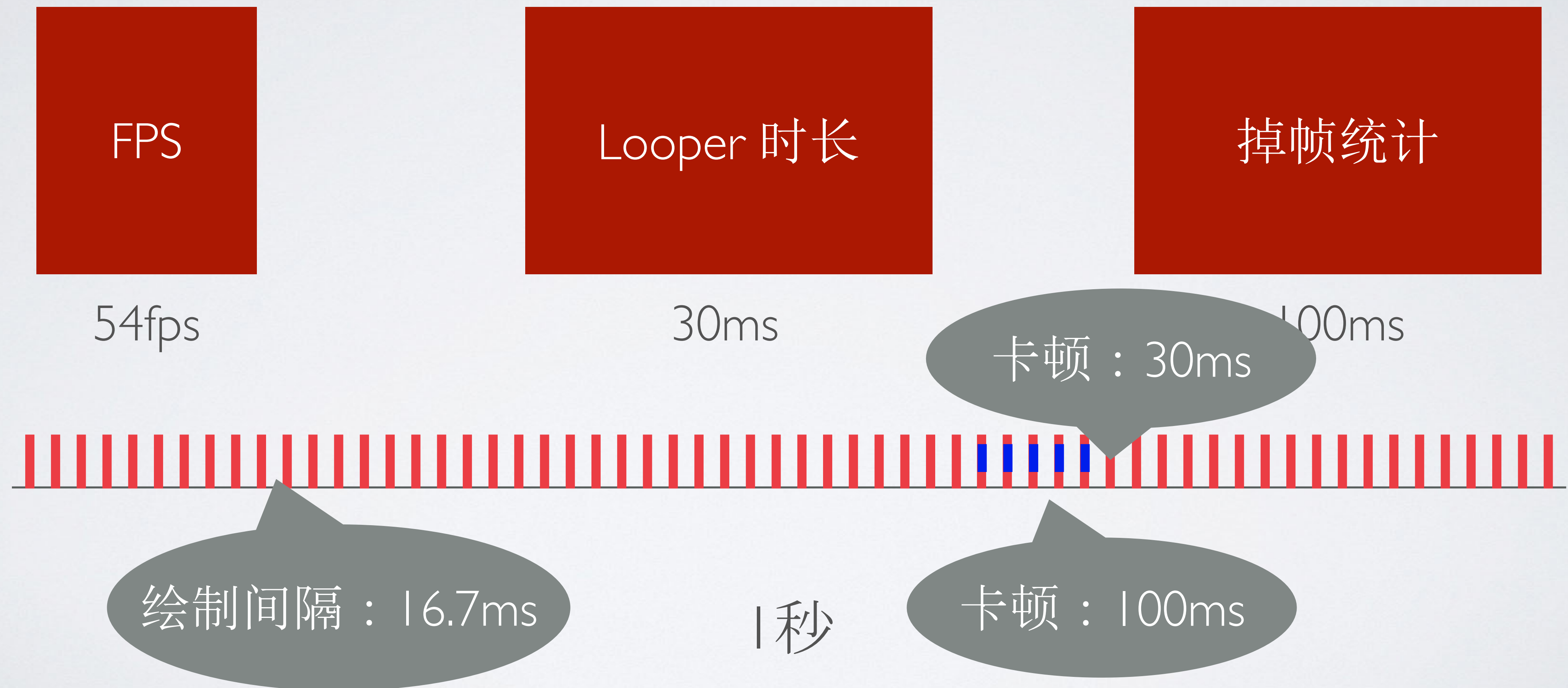




# “实验室的代价”的解决方案

- 大数据的优势：不用模拟了，用户会有这些场景
- 采集这些性能数据
- 关键点和难点：反映用户体验的性能指标？

# 反映用户的真相：流畅度

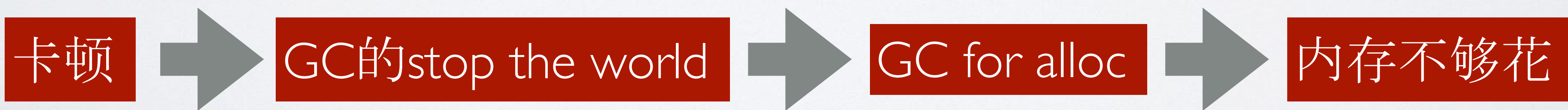




FPS本身就很好用户体验？！

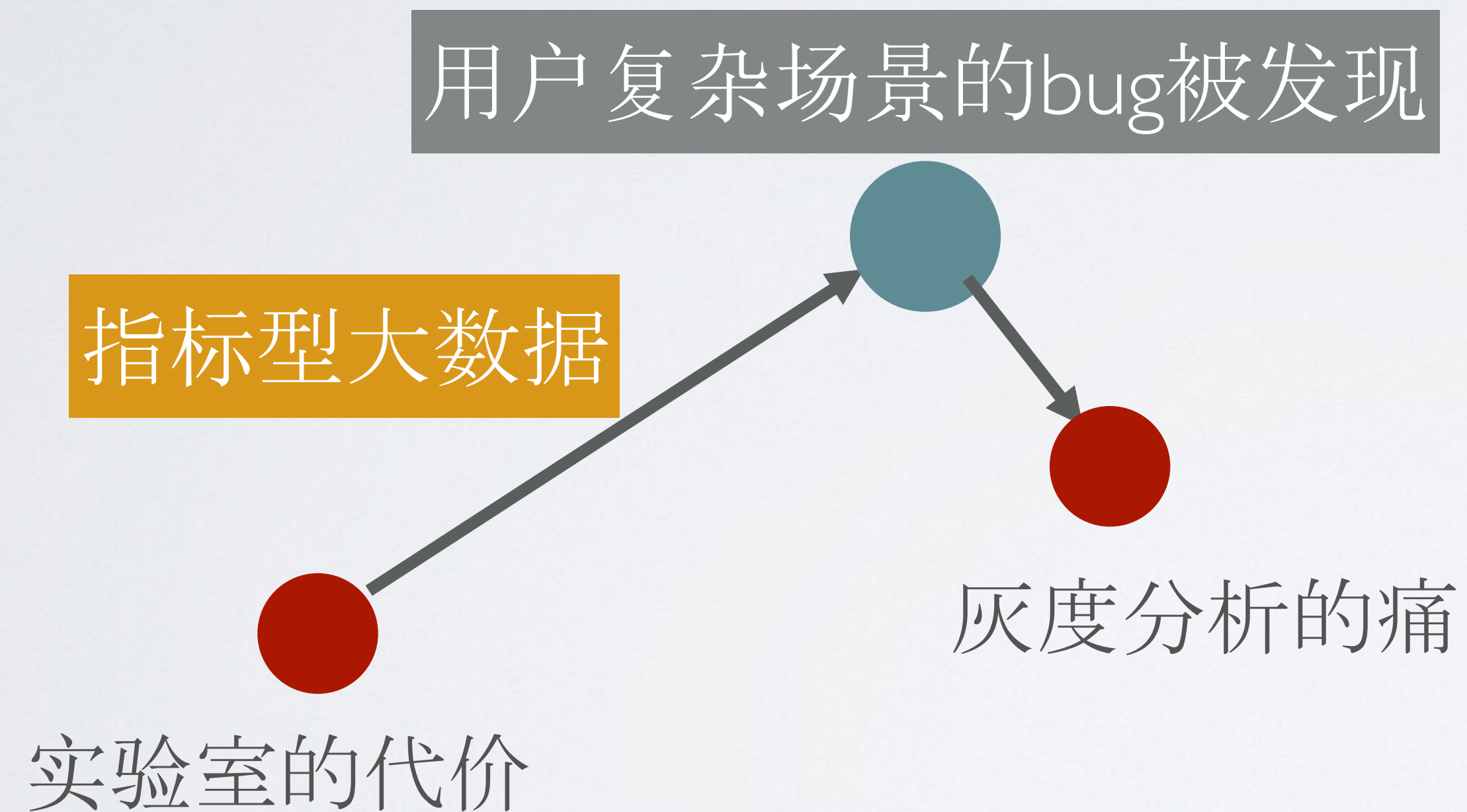
# 反映用户的真相：资源类的指标

- 内存问题对用户体验有什么影响？ { OOM, Crash  
卡顿
- OOM的crash占比
- 触顶率 (80% MaxMemoryHeap)





# 问题又来了！灰度分析的痛



测试：请问你在什么场景卡的？

A用户：想不起来了

A用户：反正你们QQ就是太卡了

测试：请问你在什么场景卡的？

B用户：我卡过么？没有呀

测试：……

# “灰度分析的痛”的解决方案

- 前面提到的分析云的能力，外网能不能用？

不能用

200MB的内存dump

取堆栈影响用户体验

VS

能用

byte[]数组裁剪

20ms获取用户堆栈



# MINIDUMP

native hook



hprofAddU1ListToRecord

myHprofAddU1ListToRecord



# 新的问题又来了？！

bug可以不依赖用户描述  
进行分析定位

用户复杂场景的bug被发现

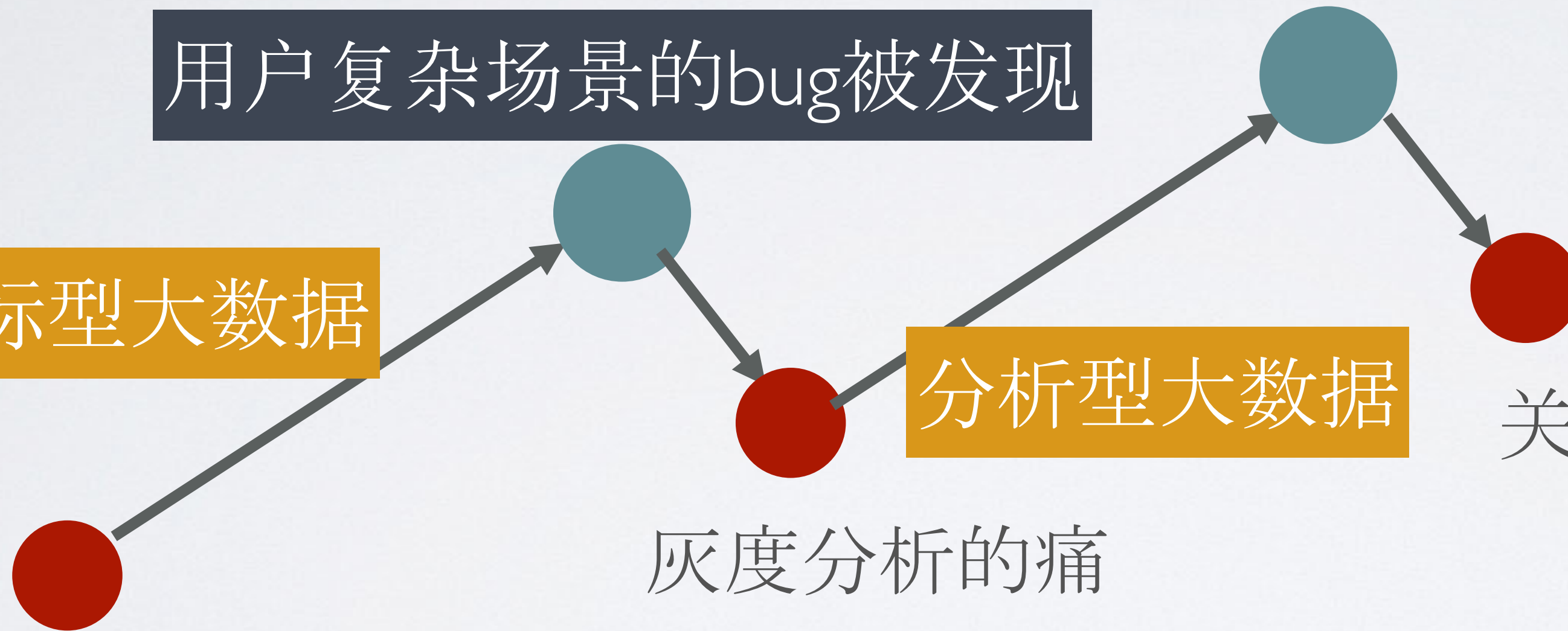
指标型大数据

分析型大数据

关键的BUG是哪个？

灰度分析的痛

实验室的代价

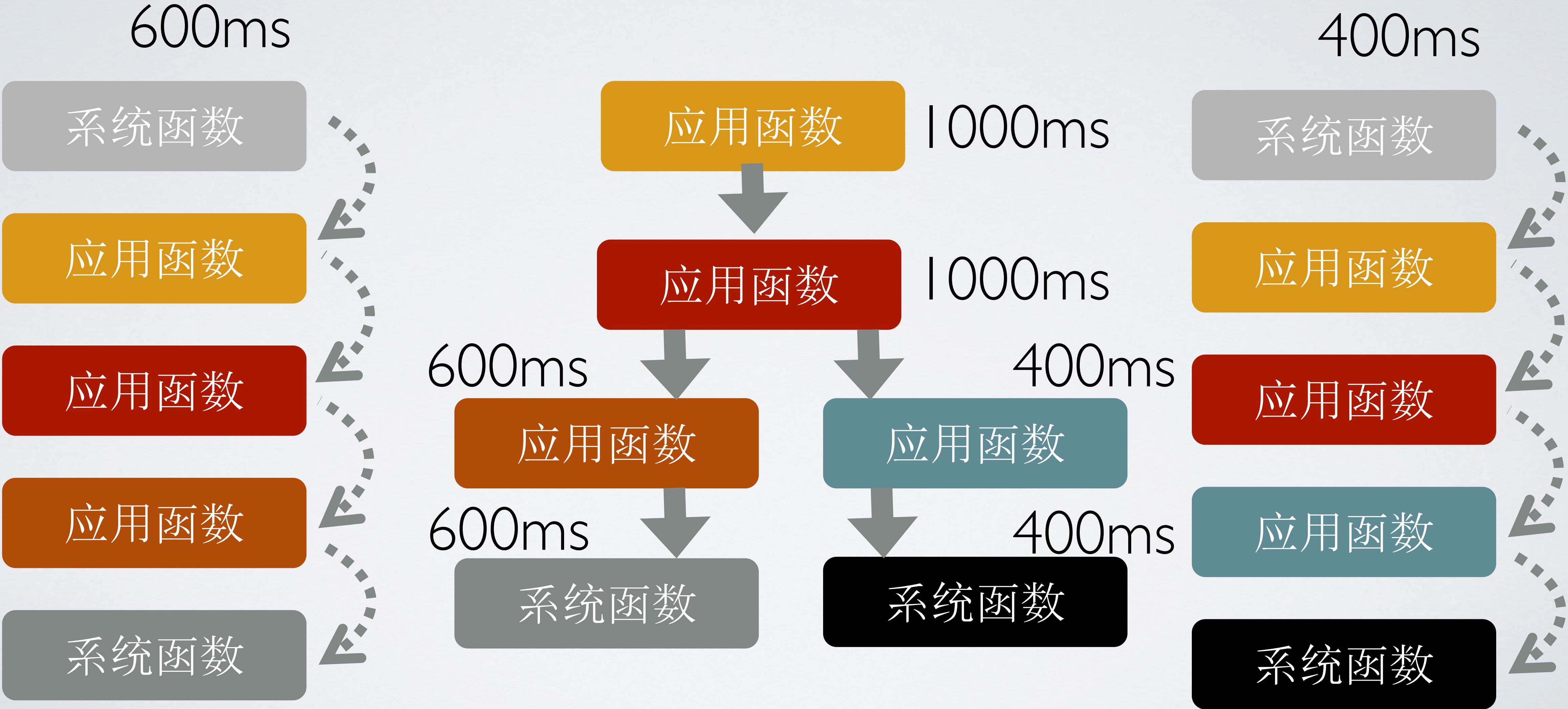




# “关键BUG的纠结”的解决方案

- 聚合算法并量化效应的大小
- 内存对象聚合，耗电堆栈聚合（频繁日志，GPS，WakeLock）

# 卡顿堆栈聚合

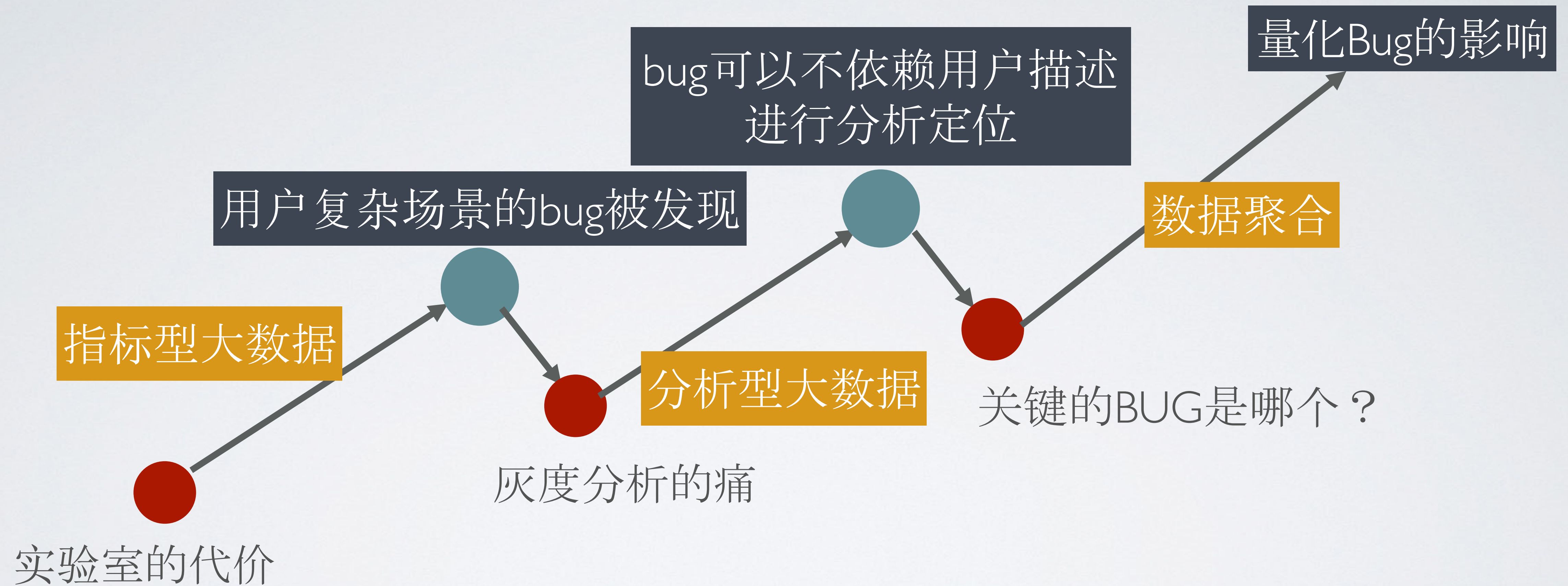




# 卡顿堆栈聚合

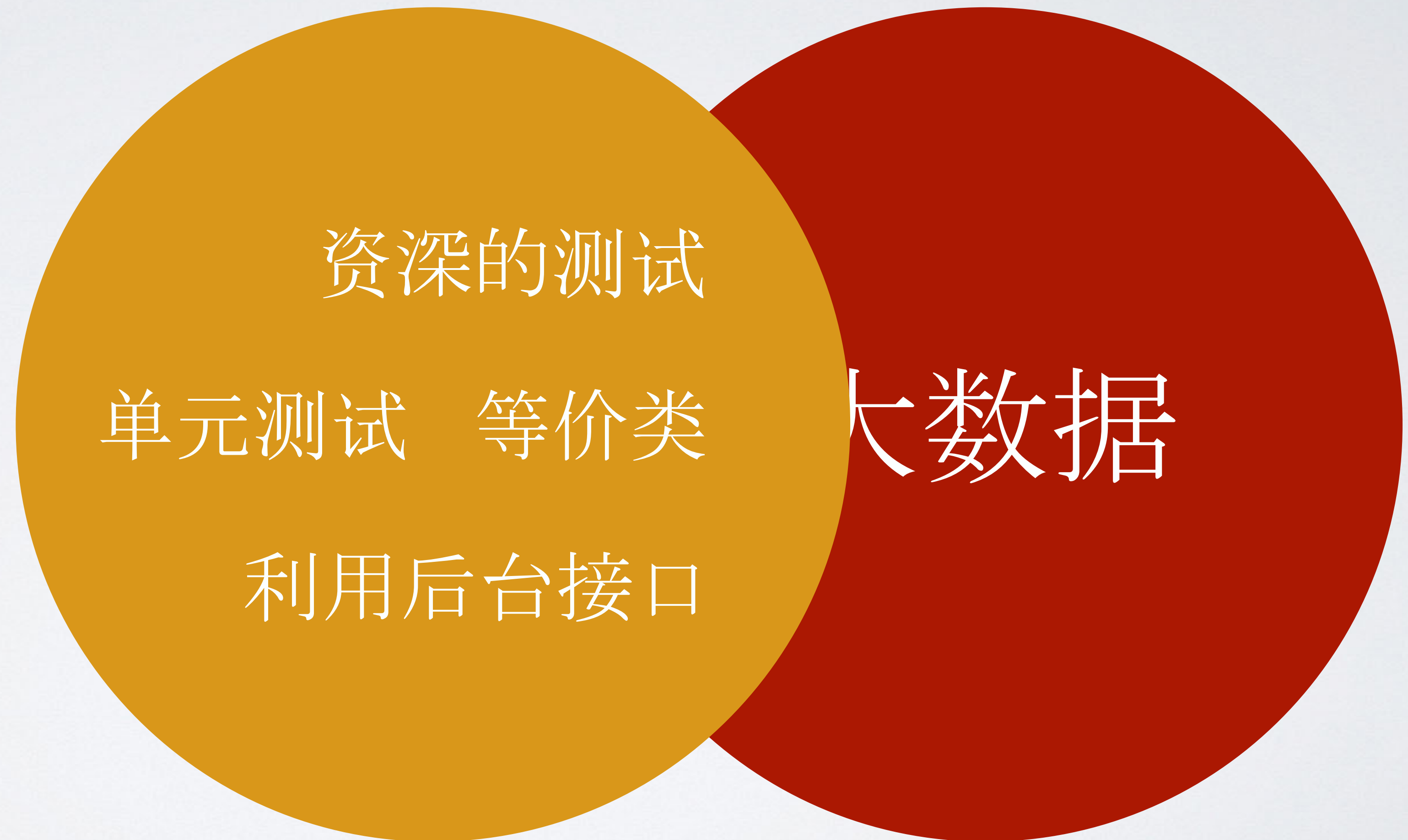
节点		辅助功能	版本占%	基准版本占%	版本占比变化	耗时(秒)
▼  com.tencent.wic		⚡ 展开红色	14.606	11.032	32.4 %	147
▼  com.tencent	:getView	⚡ 展开红色	9.637	7.150	34.8 %	97
▶  com.tenc	adapter1.getView	⚡ 展开红色	9.637	7.140	35.0 %	97
▶  com.tencent	ItemLayout.onMeasure	⚡ 展开红色	2.422	1.384	75.0 %	24
▶  com.tencent	motionAdapter.getView	⚡ 展开红色	0.780	0.215	262.1 %	8
▶  com.tencent.mobil	elViewPagerAdapter.instantiateItem	⚡ 展开红色	0.436	0.128	239.8 %	4
▶  com.tencent	el.AudioPanel.onMeasure	⚡ 展开红色	0.298	0.109	174.4 %	3
▶  com.tencent	:getItemViewType	⚡ 展开红色	0.177	0.105	69.2 %	2
▶  com.tencent	ayout	⚡ 展开红色	0.108	0.058	86.7 %	1
▶  com.tencent	.onLayout	⚡ 展开红色	0.070	0.000	19645.1 %	1
▶  mqq.app.AppAc		⚡ 展开红色	13.609	9.663	40.8 %	137

# 小结





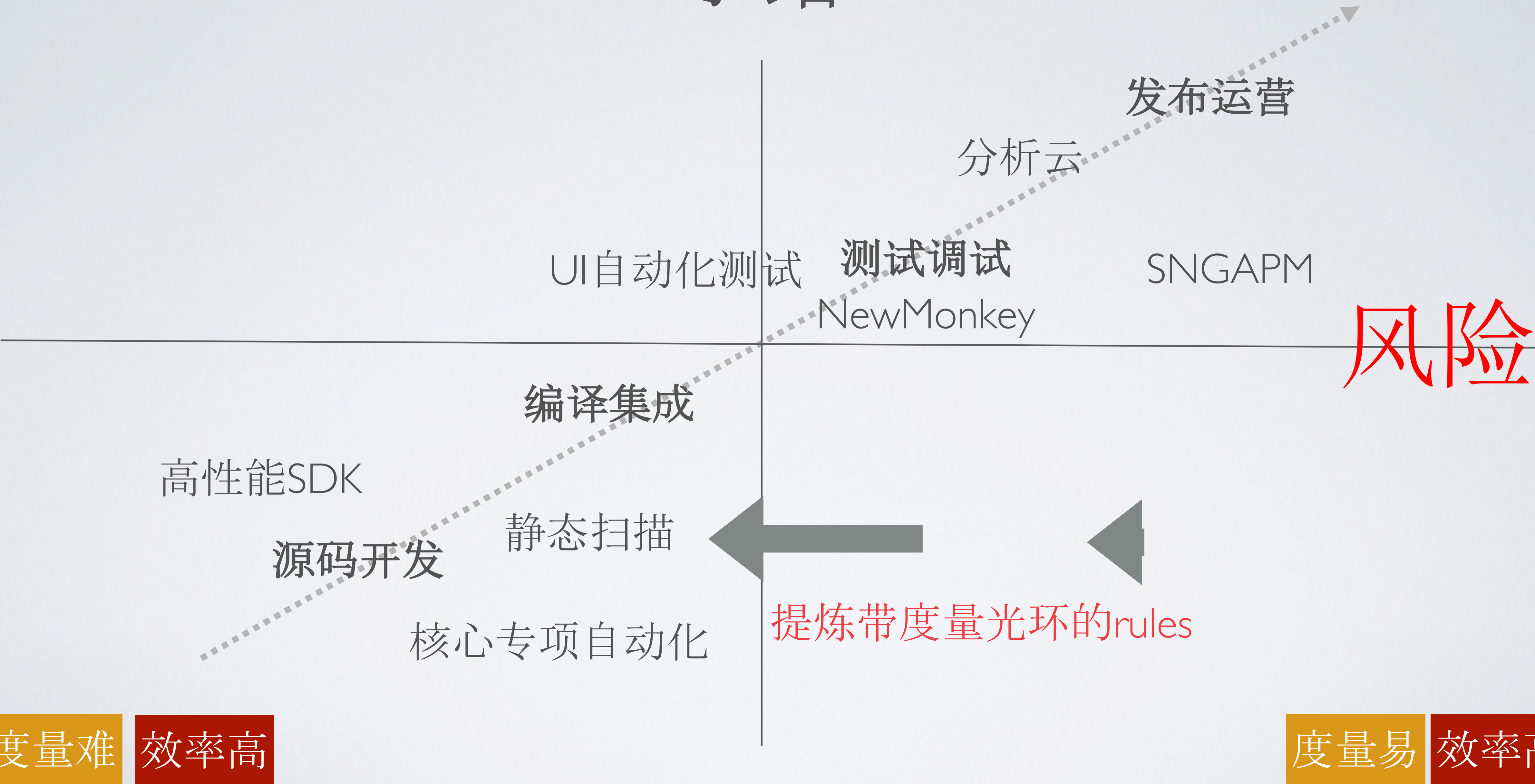
# 小结：解决思路



度量难 效率低

度量易 效率低

# 小结



度量难 效率高

度量易 效率高



# 未来：专项大数据

- 度量效应大小：聚合分析
- 发现反直觉结论：File.exist, ClassLoader很耗时
- 发现罕见案例的规律：机型的流量问题，特定用户行为

让我们“失业”的步伐来得更猛烈些！





微信



简书

广告



我们的书