



QCon 全球软件开发大会
INTERNATIONAL SOFTWARE
DEVELOPMENT CONFERENCE

BEIJING 2017

超大规模性能测试的云端方案及案例分享

SPEAKER / 金发华



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

ArchSummit
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

QCon
全球软件开发大会 [上海站]

2017年10月19-21日
咨询热线: 010-64738142

自我介绍

- 2016/4: XMeter创始人
- IBM中国开发实验室：10+年工作经验，Rational架构师，Eclipse committer

关于XMeter

- 专注于性能测试，<https://www.xmeter.net>
- 2016/9月份平台上线，实施了包括实施EMQ百万量级规模的性能测试等众多案例

应用程序性能的挑战

新技术发展带来的挑战

- 移动、物联网时代，接入的人与设备数目大幅增加
- 系统架构的复杂度增加（云，大数据，微服务等）

使用场景的碎片化

持续交付和敏捷开发模式

- 快速迭代和交付要求
- 计算资源难以快速、有效整合用于性能测试

可选解决方案

运营代替测试：灰度发布（快速上线、验证业务）

偏重验证业务，而非技术
发现问题晚，代价高

在线流量复制回放：TCPCopy（真实数据导入）

生产环境，风险高
接口更改，无法复制流量

传统性能测试工具 / 方案：LoadRunner，JMeter等（小规模模拟比较方便）

小规模：最多模拟几千虚拟用户
低效率：手工准备和维护测试环境，数天、甚至几周完成性能测试
高成本：独占，人工管理计算资源；收费高

新方案需要考虑的问题

- 易学习，为社区所接受
- 开放平台，高度可扩展性
- 支持大并发，功能完善，支持持续集成测试
- 易于维护、使用
- 高性价比
- 技术发展趋势



基于开源的方案

考虑下列因素

- 在社区中具有一定影响力
- 良好的架构、具有丰富的可扩展性
- License友好



方案：Apache JMeter

- 最流行的社区性能测试工具
- 可扩展性的架构
- Apache license



JMeter的问题

不支持大规模并发

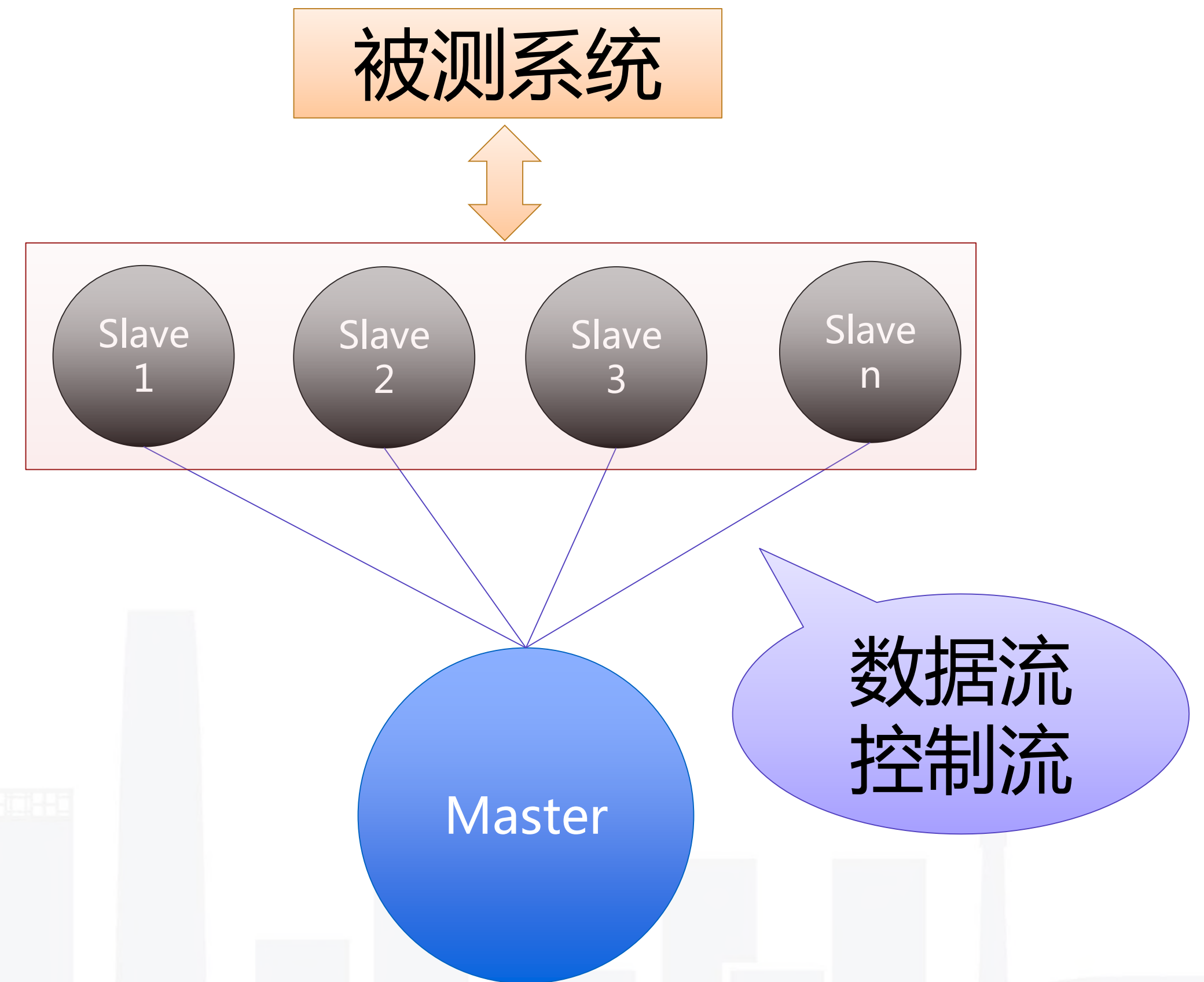
- 典型Master/Slave单点结构，无法扩展
- 测试结果数据处理瓶颈

测试报告/被测系统监控功能不完善

无法方便地做持续集成

只是工具、非平台、非解决方案

- 大量时间做测试环境准备
- 无法管理测试脚本、测试结果



改造JMeter，去中心化

数据流

分布式、容错、水平扩展

- ZK：任务分发、服务发现、分布式锁
- RMQ：测试状态、控制消息转发
- Slave(JMeter)：发送状态和测试数据，接受控制指令

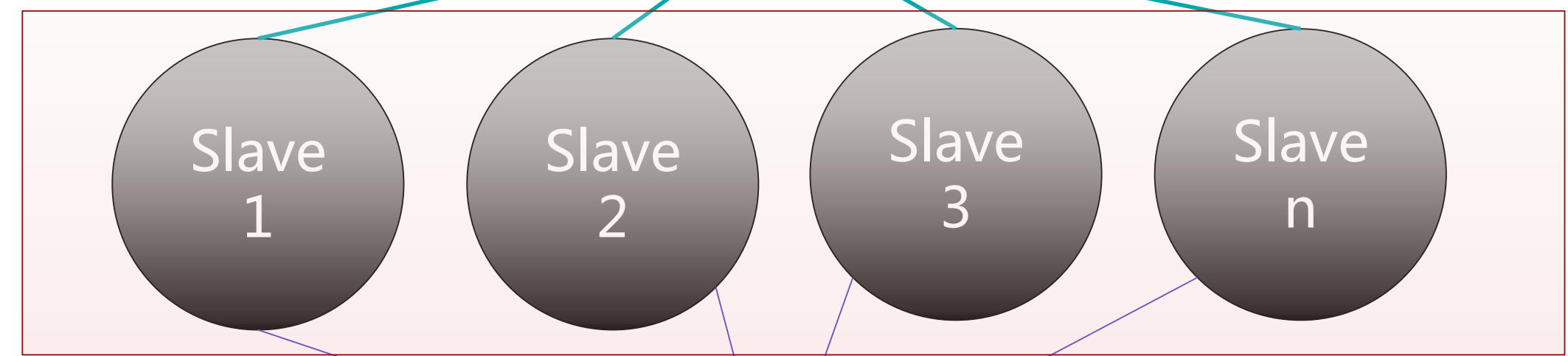
流式实时测试结果处理

- Kafka：测试数据转发、持久化
- Storm：测试数据处理、分析、存储最终结果

效果

- 完全水平扩展，支持超大规模
- 容错机制，系统更加健壮
 - 双Master (active-standby)，互为备份，对大规模性能测试很重要

流式大数据处理引擎(Kafka, Storm)



被测系统

ZK, RMQ



控制流

分布式组件的测试

复杂的系统交互：Zookeeper、RMQ

复杂的应用逻辑：分布式应用带来的不确定性：Slave断开、超时、RMQ消息无法转发等

带来的问题

- 即使手工测试也无法覆盖所有的情况
- 回归测试非常困难

解决办法：分而治之

- 需要与外部组件交互的功能进行接口抽象：单元测试的时候进行mockup，保证覆盖到所有分支
- 复杂逻辑的代码避免与外部系统进行交互，将需要的数据从方法参数传入，可以对这部分逻辑进行完整的单元测试

效果

- 近百个单元测试用例，对复杂的应用逻辑完全覆盖

测试环境管理 – 云化和容器化

利用云服务商提供的API，根据用户输入：

- 自动创建机器
- 自动创建网络（网络隔离）、GRE隧道等
- 自动绑定公网IP、分配带宽
- 分层架构：适配各种不同的云提供商，避免对上层应用产生太大影响
- 异构的云基础设施：消除计费方式不统一对应用产生的影响

实现方式从Shell替换成Python

- 控制更方便、功能更强（锁）
- 准备环境过程中需要通过RMQ报告状态

加快环境准备速度：环境预留

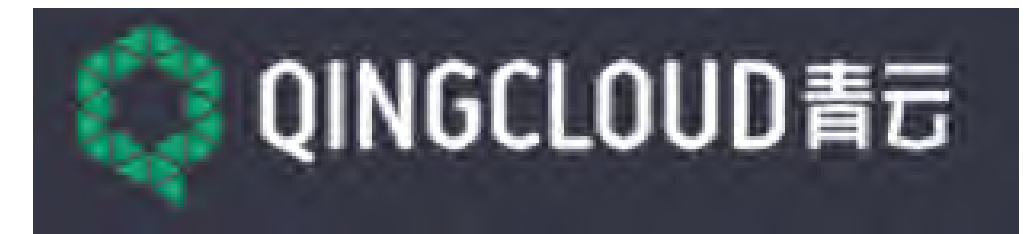
- 完全从头开始创建环境费时，采用环境预留缩短用户等待时间（等待时间缩短2/3）
- 环境是否回收、保留，利用ZK分布式锁

容器化

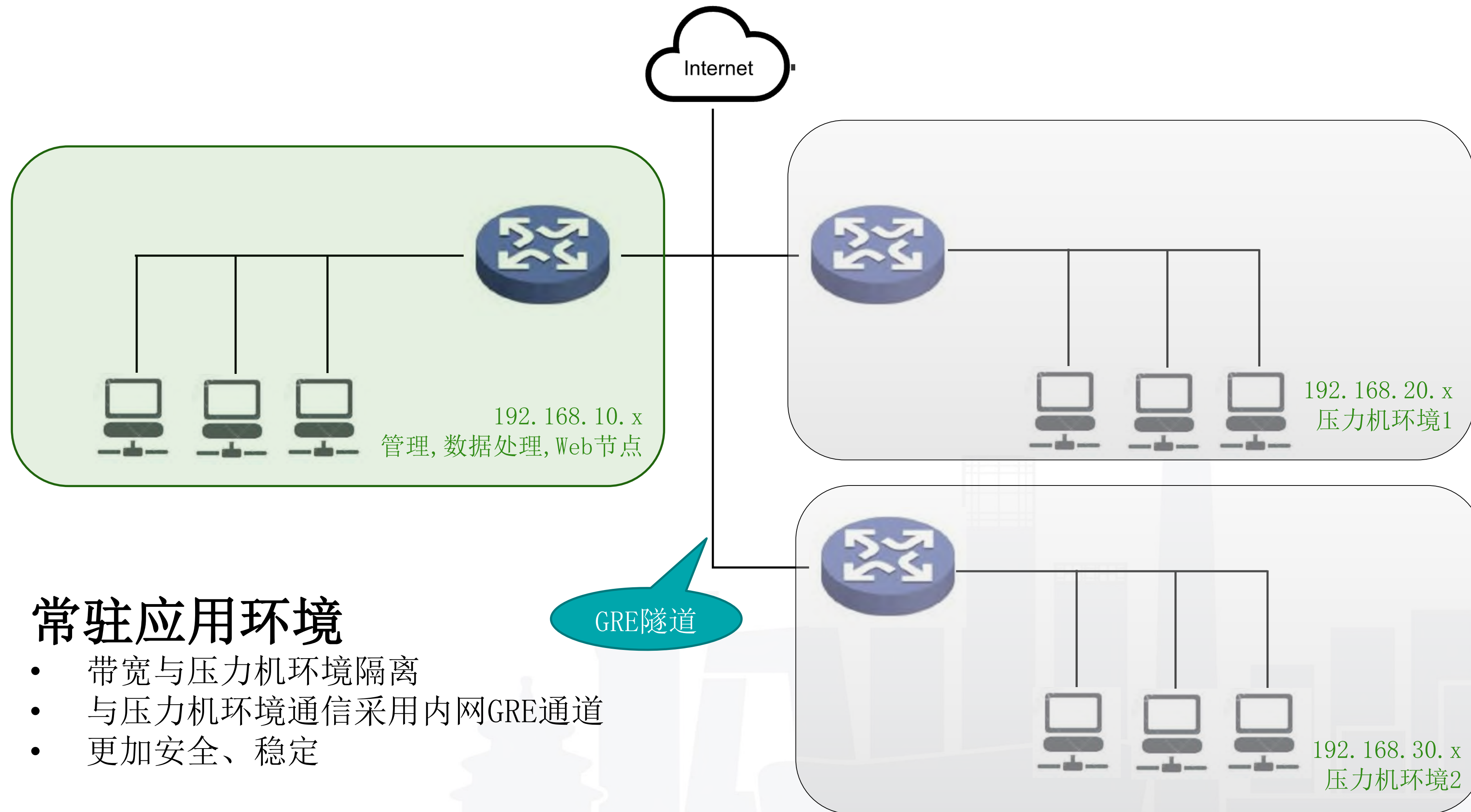
- 方便部署和维护，将JMeter相关进行了容器化（Docker）

效果

- 即需即用：无需管理测试机器、大幅节约使用成本
- 环境隔离、安全
- 分钟级创建测试环境：公有云
- 秒级创建测试环境：私有物理机器



云端多租户环境隔离



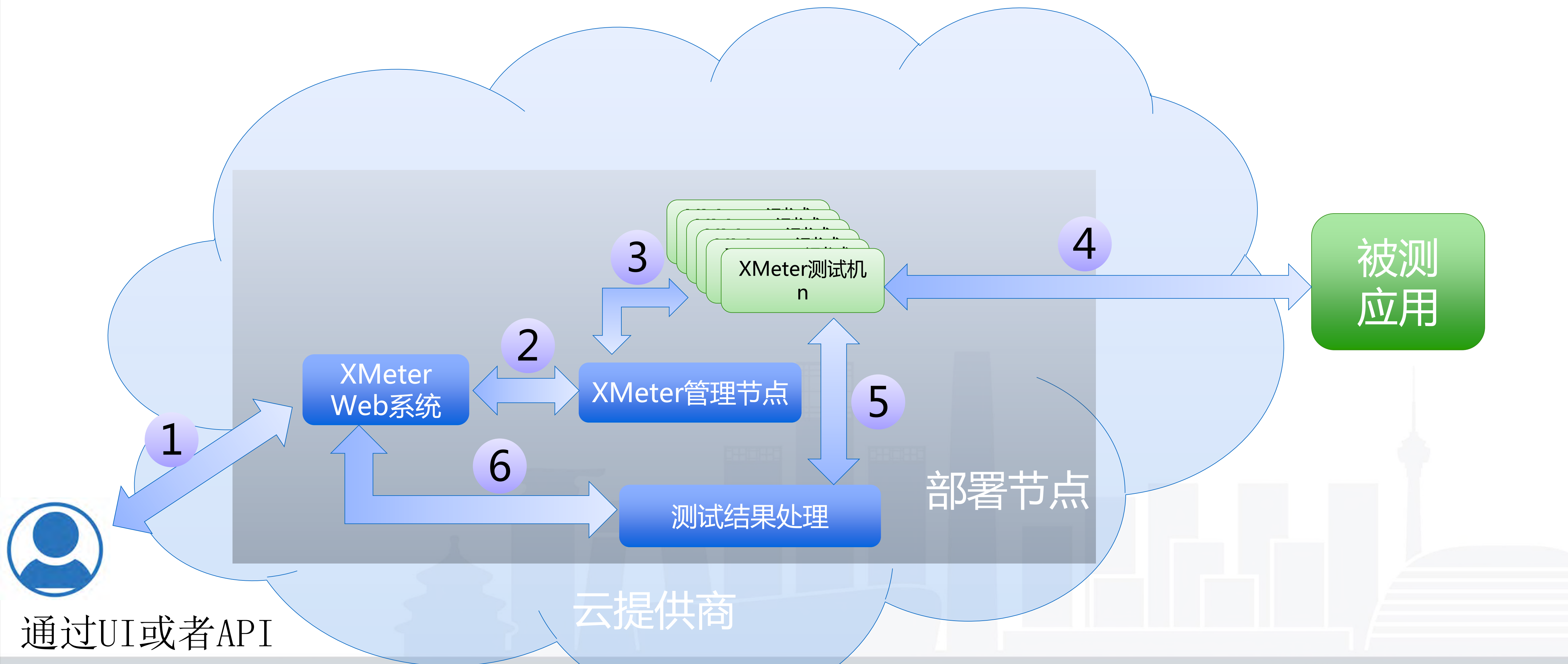
自动创建、销毁

- 虚机
- 子网络
- 路由器
- GRE通道
- 公网IP地址, 带宽

常驻应用环境

- 带宽与压力机环境隔离
- 与压力机环境通信采用内网GRE通道
- 更加安全、稳定

系统总体架构拓扑图



平台其它功能的完善

在线管理测试用例和结果

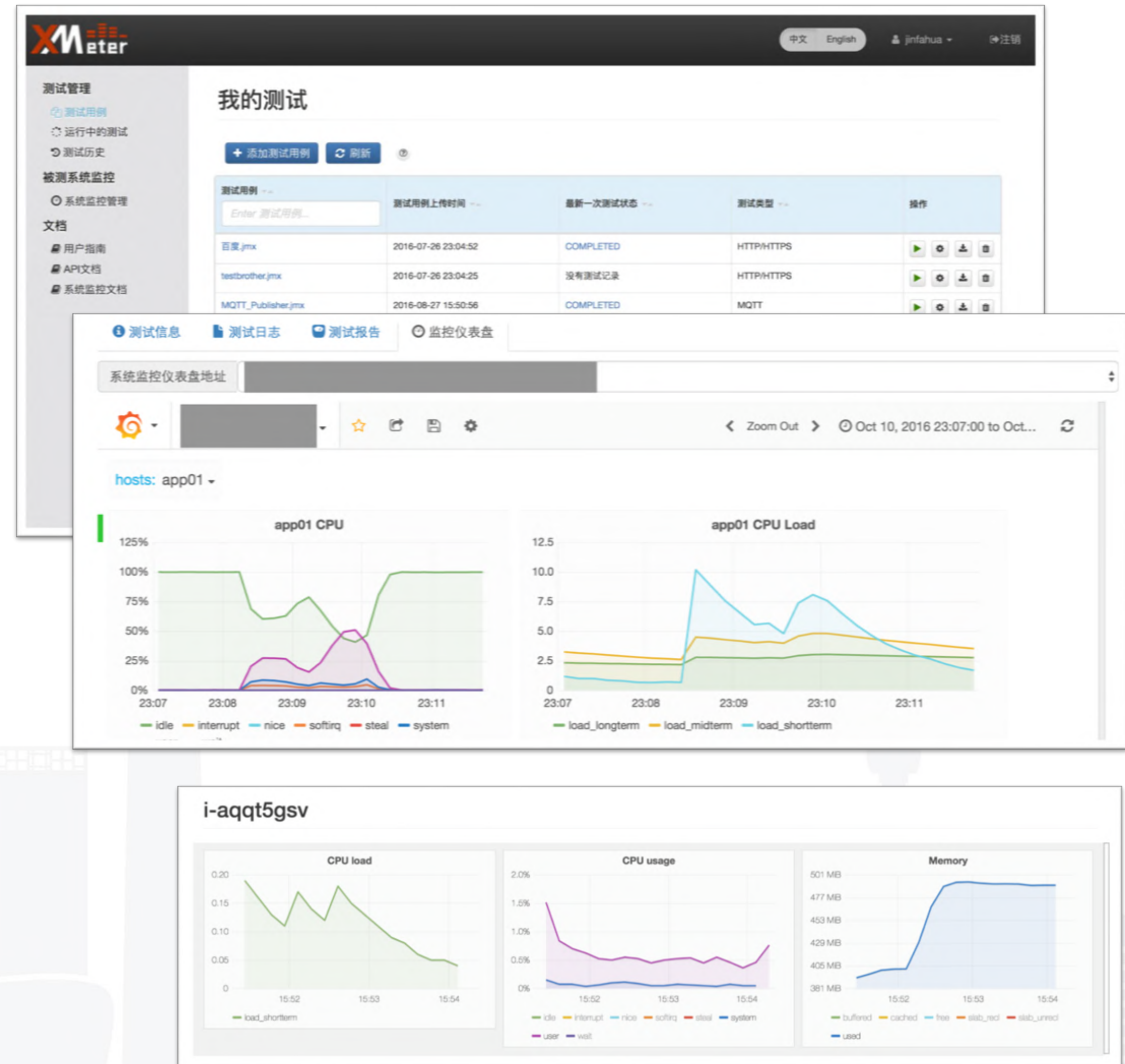
- 接口测试 (REST API)
- ReactJS

完善监控系统

- 被测系统监控
利用Grafana、Collectd和InfluxDB对被测系统实施监控
提供扩展机制允许加入第三方中间件的监控
- 保障测试质量，对测试发起机器监控
避免压力发起机器工作负载过高导致对测试结果有影响

API支持

- 常用功能暴露REST API，提供性能测试持续集成



方案前后对比

	前	后
大规模虚拟用户支持 (>10k)	否	是
自动管理测试环境	否	是
实时测试报告	是*	是
在线测试管理与协作	否	是
易于持续集成支持 (CI)	否	是

* 在大规模用户的情况下不支持

系统实际效果图

请输入百度.jmx的运行参数

本次测试预计花费(元): 87.00

>> 选择多站点运行模式

青云北京

10000

3 分钟

测试描述(150字以内)

提交测试

测试数据明细 (100%数据)

Time	page	最大响应时间	最小响应时间	平均响应时间	平均吞吐量	平均请求大小	请求响应成功率	平均标准差
10-08	59 /tabTestpaperExercises_selectExercisesFromRedis	47 s	68.00 ms	3.90 s	39.30	12.86 KiB	100.00%	3.49 K
10-08	79 /user_logout	54 s	60.00 ms	1.21 s	37.48	2.03 KiB	100.00%	1.62 K
10-08	11 /user_login	27 s	68.00 ms	1.18 s	40.96	2.44 KiB	100.00%	1.36 K

测试信息 测试日志 测试报告 APM仪表盘

平均吞吐量	最大活跃虚拟用户数	平均响应时间	最大响应时间
438	1000	1 s	53.9 s
最小响应时间	请求响应成功率	验证点成功率	平均请求大小
29 ms	100.0%	100%	3 KiB

请求响应时间

吞吐量

测试信息 测试日志 测试报告 APM仪表盘

下载全部日志文件

56ec0dad5625_jmeter.log

549ceb0f76f4_jmeter.log

549ceb0f76f4_sampler_debug.log

56ec0dad5625_sampler_debug.log

2016-10-08 11:37:04.026:

```
http://[redacted]user_validat?uuid=0.017173854858121396
Request headers:
Connection: keep-alive
Referer: http://[redacted]index.html
Accept-Language: en-US,zh-CN;q=0.8,en;q=0.5,zh;q=0.3
X-Requested-With: XMLHttpRequest
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: application/json, text/javascript, */*; q=0.01
Host: [redacted]

Response headers:

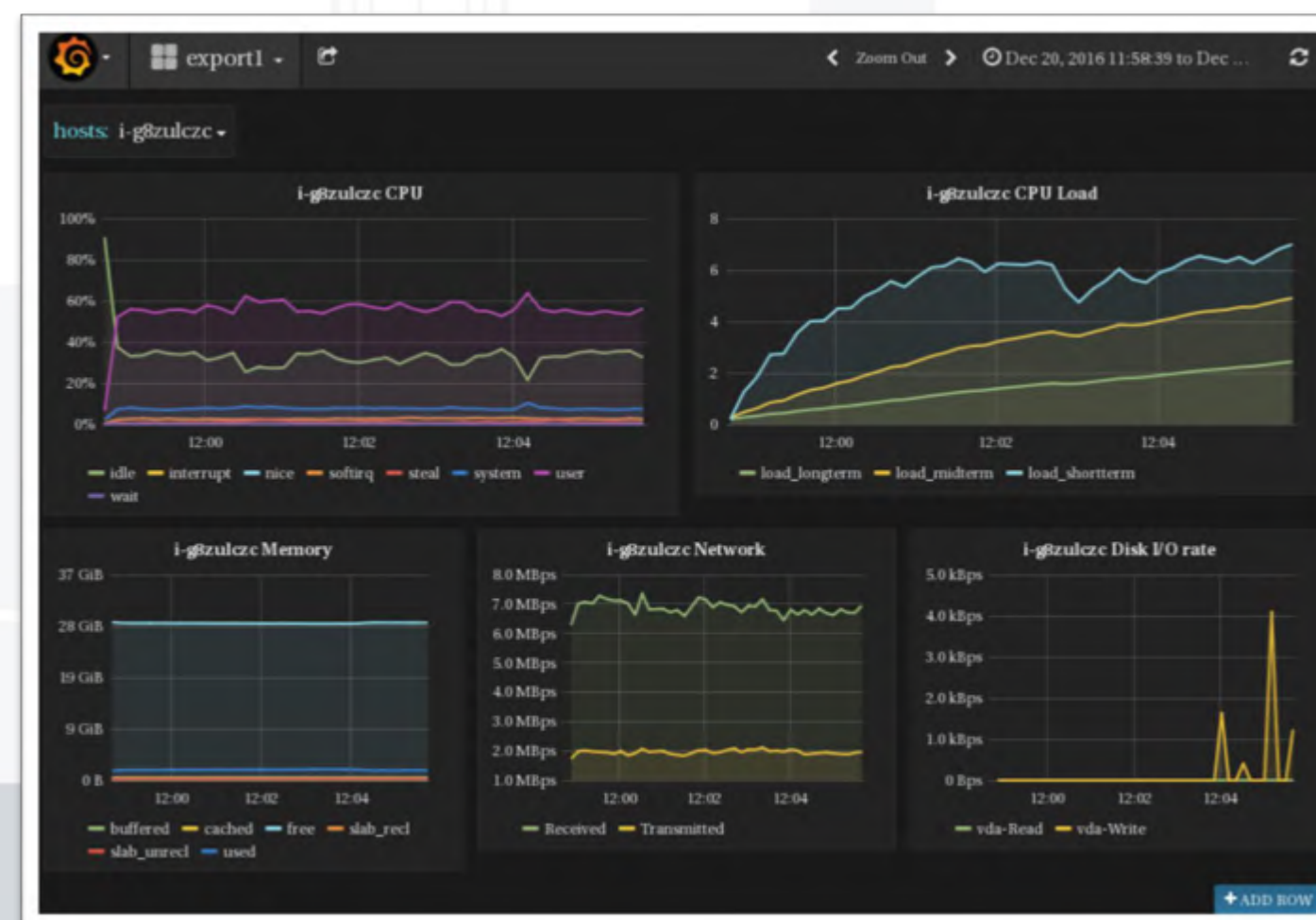
Assertion results:

Contents:
java.net.SocketException: Connection reset
    at java.net.SocketInputStream.read(SocketInputStream.java:196)
    at java.net.SocketInputStream.read(SocketInputStream.java:122)
    at org.apache.http.impl.io.AbstractSessionInputBuffer.fillBuffer(AbstractSessionInputBuffer.java:160)
    at org.apache.http.impl.io.SocketInputBuffer.fillBuffer(SocketInputBuffer.java:84)
    at org.apache.http.impl.io.AbstractSessionInputBuffer.readLine(AbstractSessionInputBuffer.java:273)
```


案例分享 - 百万MQTT连接测试

EMQ - <http://emqtt.io/> 开源社区最流行的MQTT中间件

- 实施EMQ基准性能测试，共计将近30个测试场景，20pd完成所有工作
- 开放性：定制开发开源的JMeter的MQTT插件 <https://github.com/emqtt/mqtt-jmeter/>
- 节约成本：测试时间从2周下降到1个小时，大幅降低测试所花成本
- 持续集成：XMeter测试服务作为将来版本发布的质量保证
- 易于定位问题：定制后端报告模版，实时监控EMQ服务器表现
- 样例测试报告
 - 中文入口 - <https://github.com/emqtt/emq-xmeter-cn>
 - 英文入口 - <https://github.com/emqtt/emq-xmeter-en>
 - 连接测试报告 - <https://github.com/emqtt/emq-xmeter-cn/blob/master/connection.rst>
 - 吞吐量测试报告 - <https://github.com/emqtt/emq-xmeter-cn/blob/master/throughput.rst>



案例分享 - MQTT测试实践

JMeter插件消息时延计算（异步系统时延计算）

- Pub / Sub机制，由于异步转发消息，Sub端无法得到消息发送时间，无法计算时延
解决方法：Pub在发送消息的时候带上所在服务器的发送时间戳
- 服务器时间的同步

系统配置更改

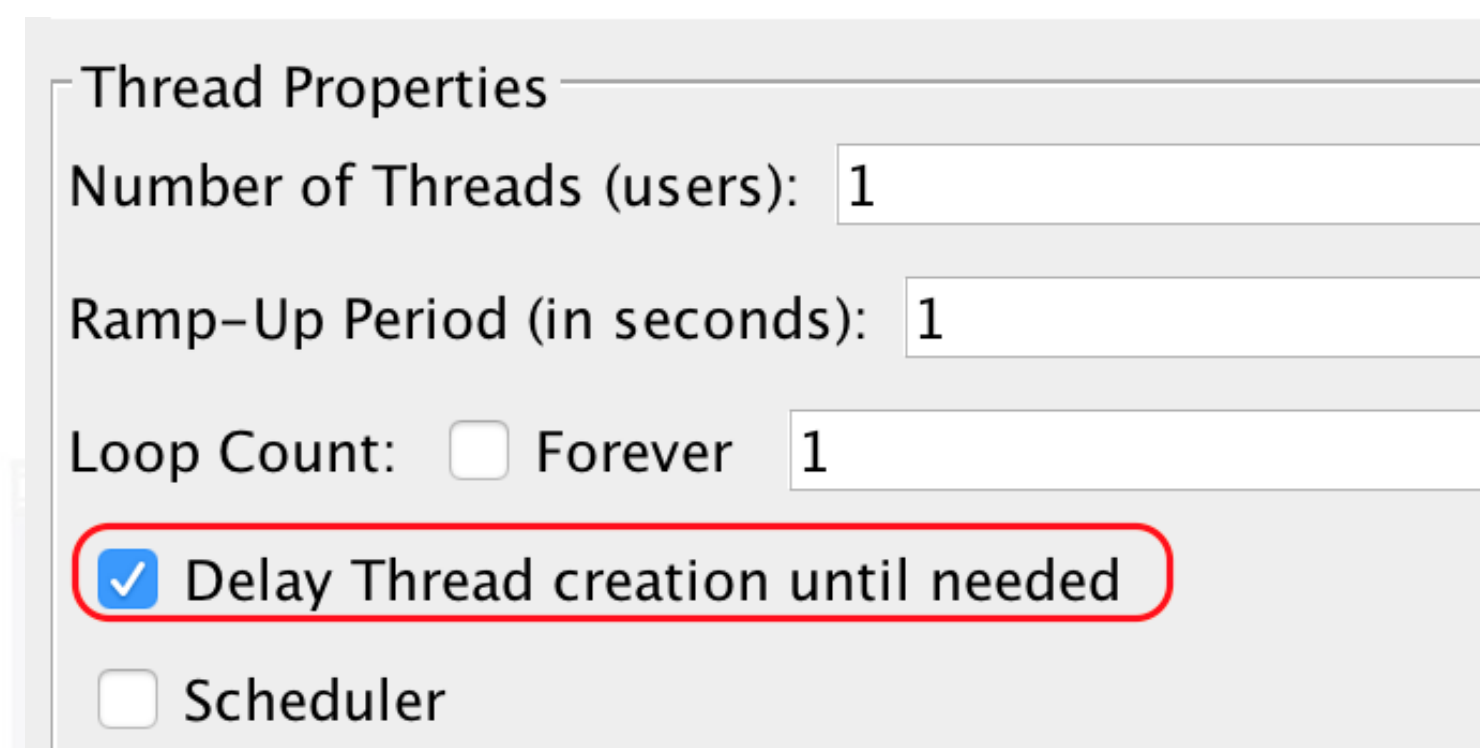
- 进程资源控制，文件、线程数等：ulimit
- 连接重用设置 - net.ipv4.tcp_tw_reuse

JMeter脚本设置

- 在必要的时候创建线程

服务器运行的虚拟用户数和带宽进行调整

- 可以针对脚本的情况对硬件配置和网络带宽进行调整



The image shows a screenshot of the 'Thread Properties' dialog box in JMeter. The dialog has several fields and checkboxes. The 'Number of Threads (users)' field is set to 1. The 'Ramp-Up Period (in seconds)' field is set to 1. The 'Loop Count' section has a radio button for 'Forever' and a text box set to 1. The 'Delay Thread creation until needed' checkbox is checked and highlighted with a red border. The 'Scheduler' checkbox is unchecked.

Thread Properties	
Number of Threads (users):	1
Ramp-Up Period (in seconds):	1
Loop Count:	<input type="checkbox"/> Forever 1
<input checked="" type="checkbox"/> Delay Thread creation until needed	
<input type="checkbox"/> Scheduler	



关注QCon微信公众号，
获得更多干货！

Thanks!



主办方 **Geekbang** > **InfoQ**
极客邦科技