



无人值守的性能测试

- 关于性能测试自动化的一些实践和探索

邱 鹏 Ricky Qiu

趋势科技中国研发中心

about Ricky

2005年加入趋势科技中国研发中心，现为企业级安全产品高级测试工程师。有丰富的软件性能测试和系统稳定性测试工作经验，并负责部门相关测试工具的开发、评估和选型。毕业于华中科技大学，获计算机硕士学位。

Contact me:

Blog: <http://blog.csdn.net/superqa>

MSN: qiupeng2002@hotmail.com

Email: rickyqiu1024@gmail.com

Survey

- 有多少人做过性能测试？
- 有多少人自己开发过性能测试的工具？
- 通常做一次完整的性能测试需要多少时间？

不包含:

- 性能测试的基本技术和方法
 - 真实用户行为的提取和模拟
 - 样本数据的提取和构造
 - 产生测试流量的工具使用
 - 数据的分析
- 性能测试的调优

专注于：

- 最大化现有测试工具和方法的价值
- 最小化测试的工作量
- 提高性能测试的场景覆盖和频度
- 缩短性能测试的周期
- 及时发现性能问题

如何记录一个人的肖像？



但是，如果：

- 需要全公司1000人的肖像
- 反复，比如模特穿不同衣服的肖像
- 需要快速的获取，比如5分钟
- ...

只需按下快门!



- 换胶卷
- 手动测光
- 手动对焦
- 底片冲印



普及

- ✓ 立即浏览照片
- ✓ 高速连拍
- ✓ 直出JPG
- ✓ 出去3天拍1000张
- ...

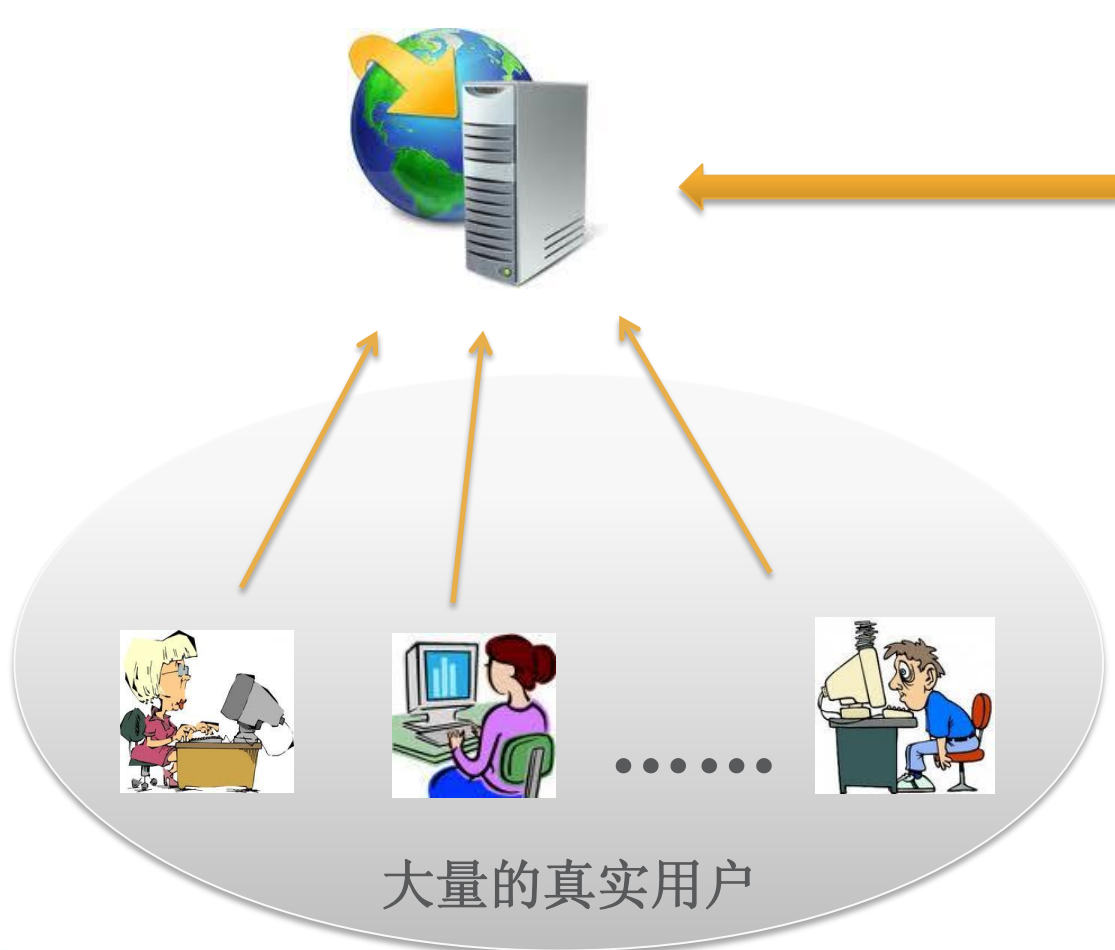
- 手动换片
- 手动测光
- 手动对焦
- 底片冲印



少数家庭用户
+ 旅游或者聚会

少数专业人士
正式场合

如何衡量一个系统的性能？



测试工具产生的
虚拟用户

新的需求1: 不同的场景

Description	Server Type	Mode	Message Size (KBytes)	Throughput (messages per hour)
Peak Capacity <i>no active policies</i> 配置	Single CPU	Standalone <i>Scanner, Policy and Database on same server</i>	1	313
			5	623
			15	364
			28	801
			28 (Mixed)	216
			53	117
			53 (Mixed)	888
			100	449
			500	690
			1000	186
Antivirus Only <i>Only AV active</i>	Single CPU	Standalone <i>Scanner, Policy and Database on same server</i> 部署	1	186
			5	568
			15	441
			28	1024
			28 (Mixed)	539
			53	1059
			53 (Mixed)	110
			100	1059
			500	206
			1000	400
Anti-Spam Only <i>Only AS active</i>	Single CPU	Standalone <i>Scanner, Policy and Database on same server</i> 样本	1	110
			5	934
			15	898
			28	154
			28 (Mixed)	169
			53	533
			53 (Mixed)	603
			100	365
			500	808
			1000	969
Content Filter Only <i>Only CF active</i>	Dual CPU	Distribution <i>Scanner, Policy and Database on Different Server</i>	1	603
			5	579
			15	659
			28	929
			28 (Mixed)	568
			53	287
			53 (Mixed)	287
			100	262
			500	942
			1000	559
Default Configuration <i>AV, AS and CF active</i>	Dual CPU	Distribution <i>Scanner, Policy and Database on Different Server</i>	1	937
			5	525
			15	476
			28	251
			28 (Mixed)	888
			53	
			53 (Mixed)	
			100	
			500	
			1000	

配置 x 5

硬件 x 3

部署 x 2

样本 x 9

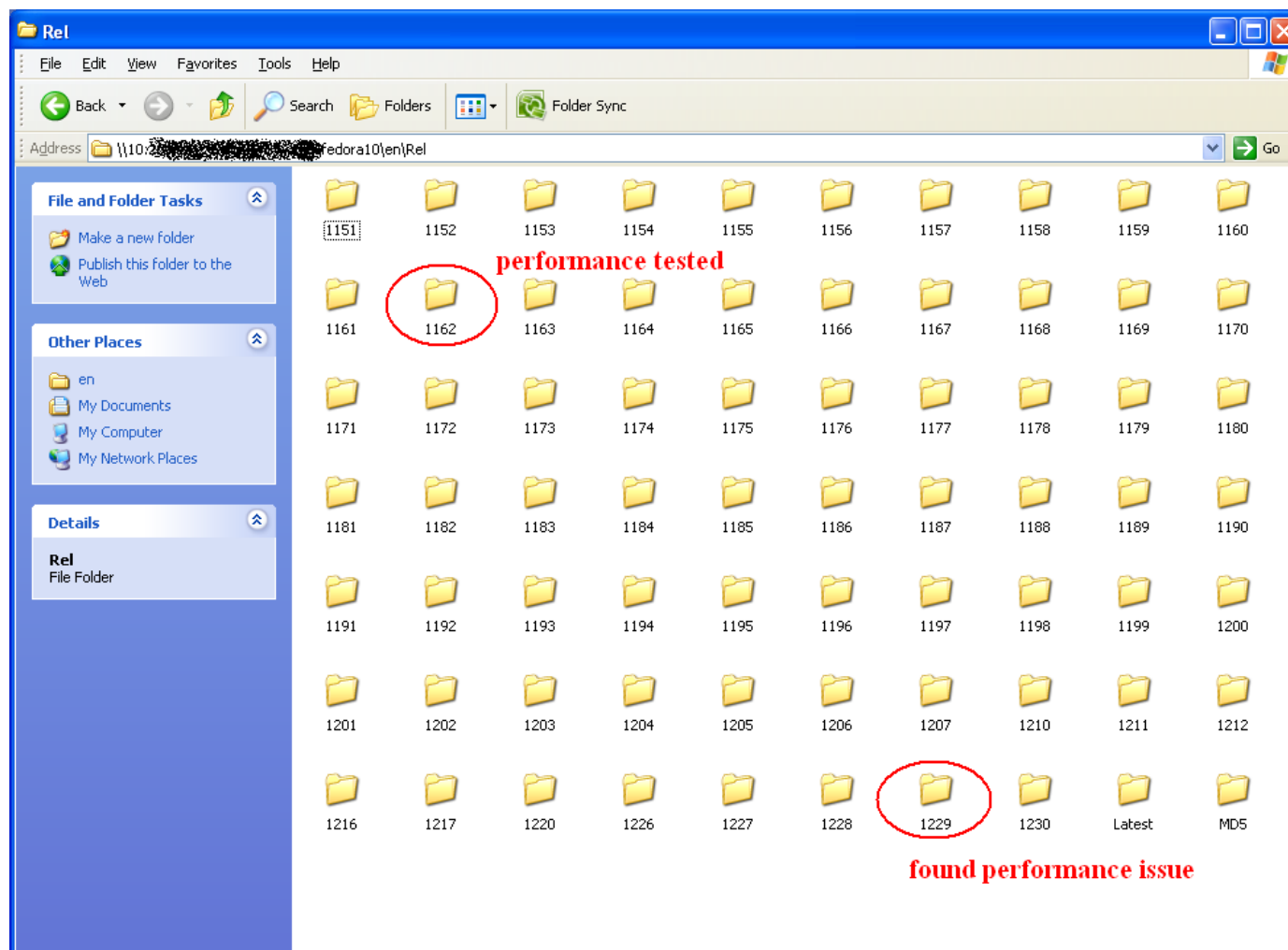
270 次测试!

类似的需求:

- 在不同的压力情况下系统的资源使用率?
- 不同数据量的情况下系统的响应时间?
- 不同并发用户数目时的响应时间?
- 不同产品设置时的性能?
 - Cache? Keep alive? buffer size?

.....

新的需求2：及时发现性能缺陷



- 需求很合理
- 工作很重复
- 周期很漫长
- 效率很低

• 无人值守的性能测试之5大境界

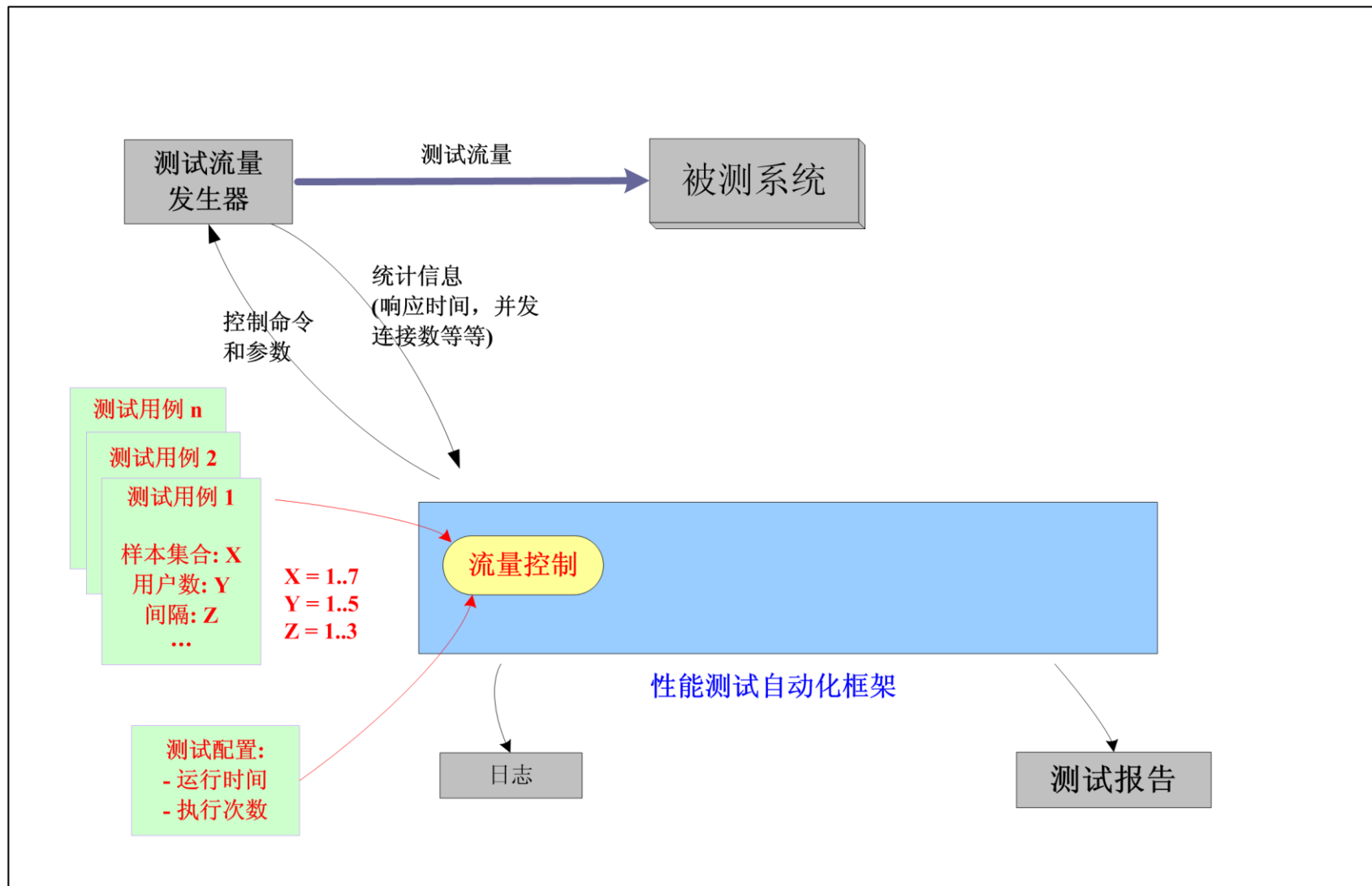
境界一： 让不同用例的执行自动化

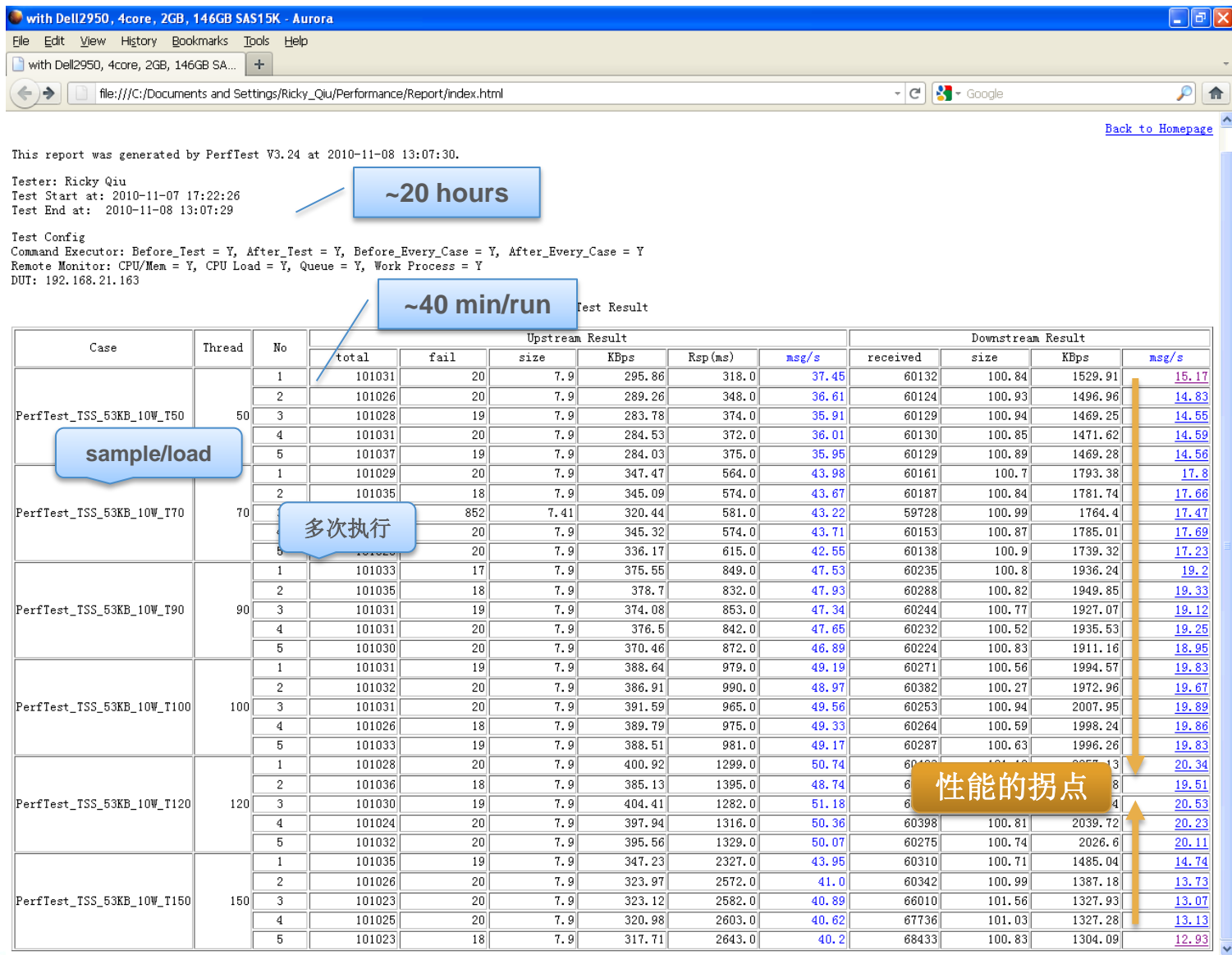
➤ 有大量的重复执行的工作

- 不同的样本(sample)
- 不同的压力
 - 并发用户数
 - 访问的时间间隔 (think time)
- 不同的持续执行时间
- 重复执行多次(减少偶发错误)
-

➤ 需要人工干预

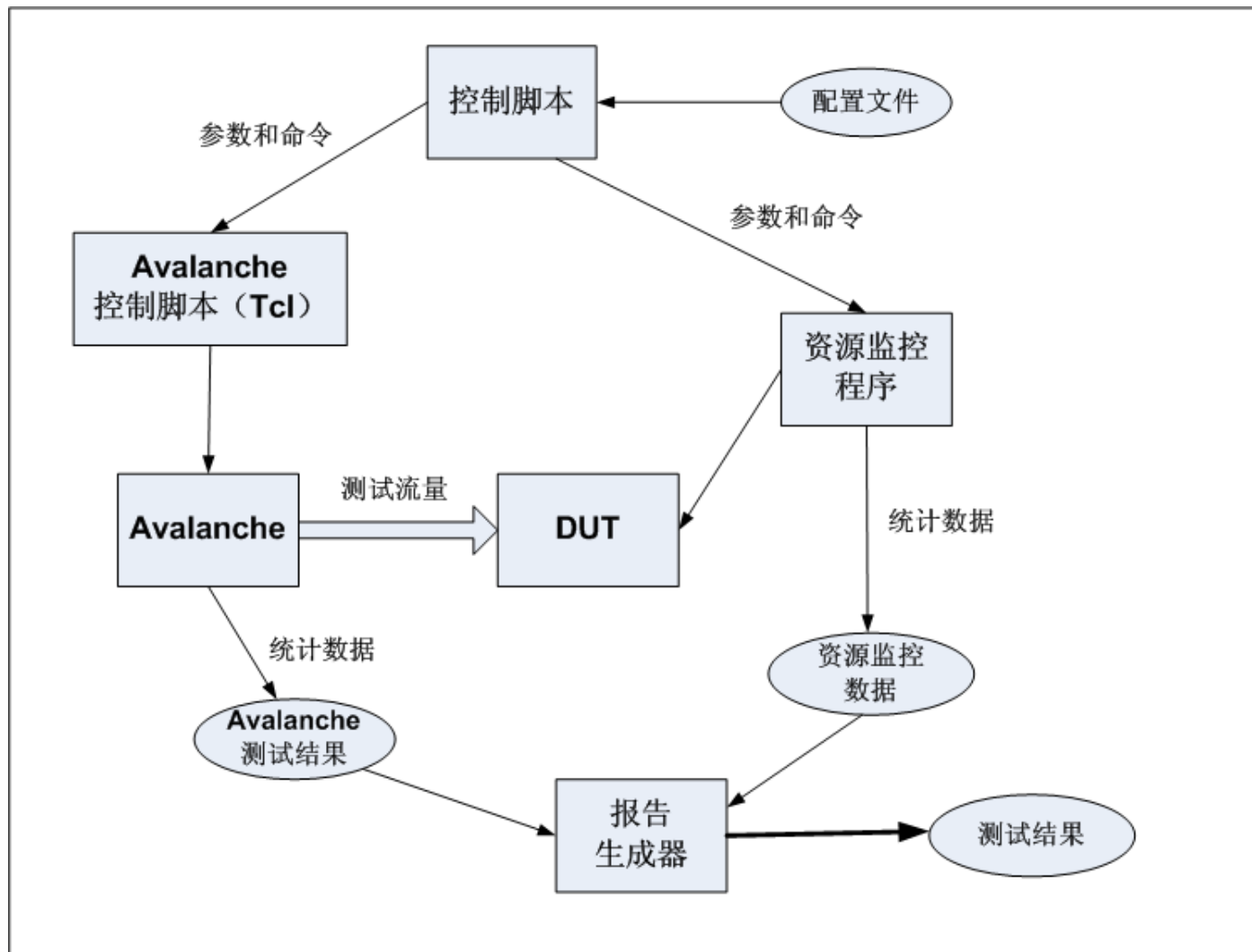
- 配置工具
- 发起测试
- 记录测试结果





Report demo

基于已有工具的二次开发



境界二：让资源监控自动化

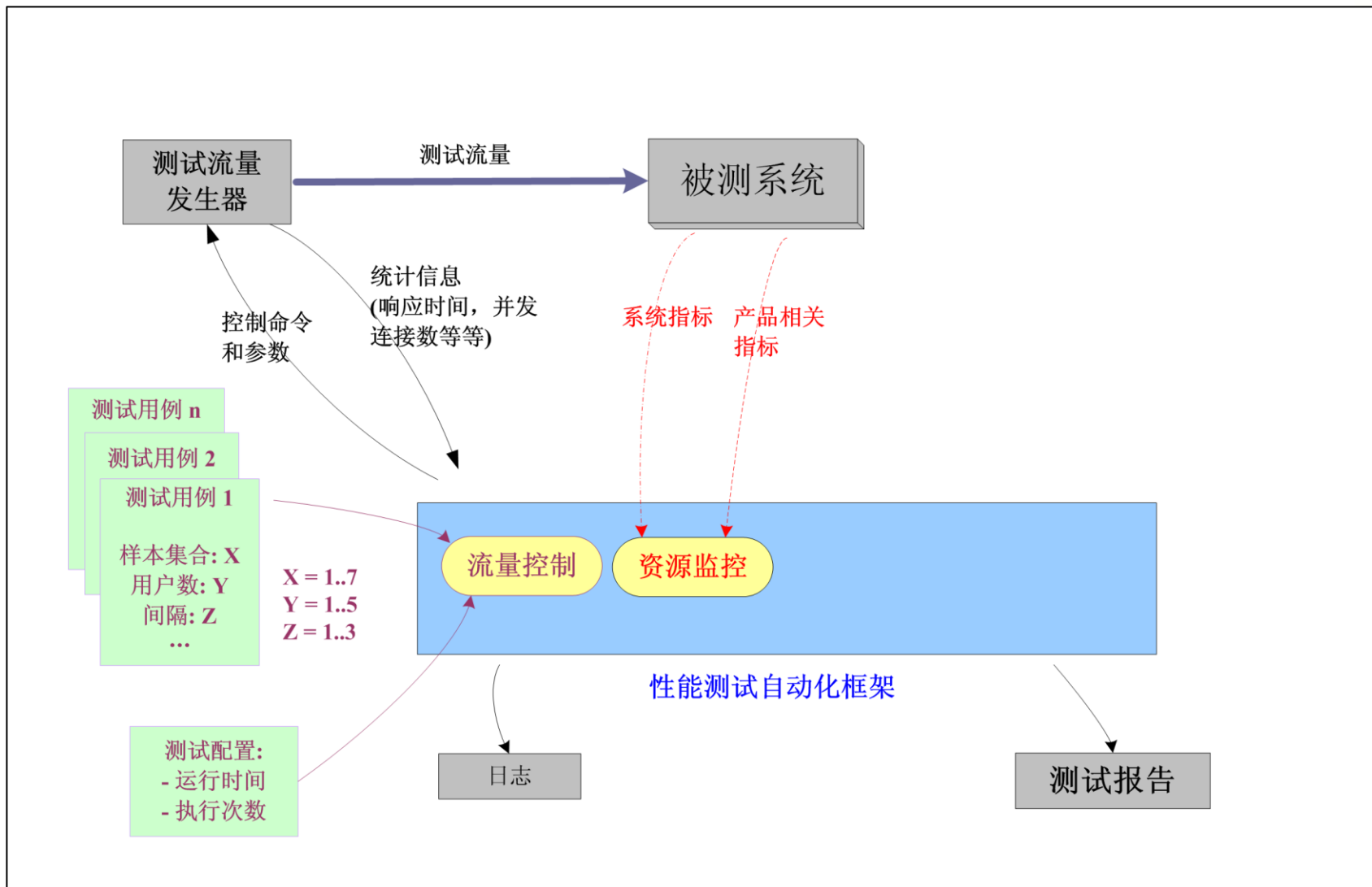
我们有办法采集到数据：

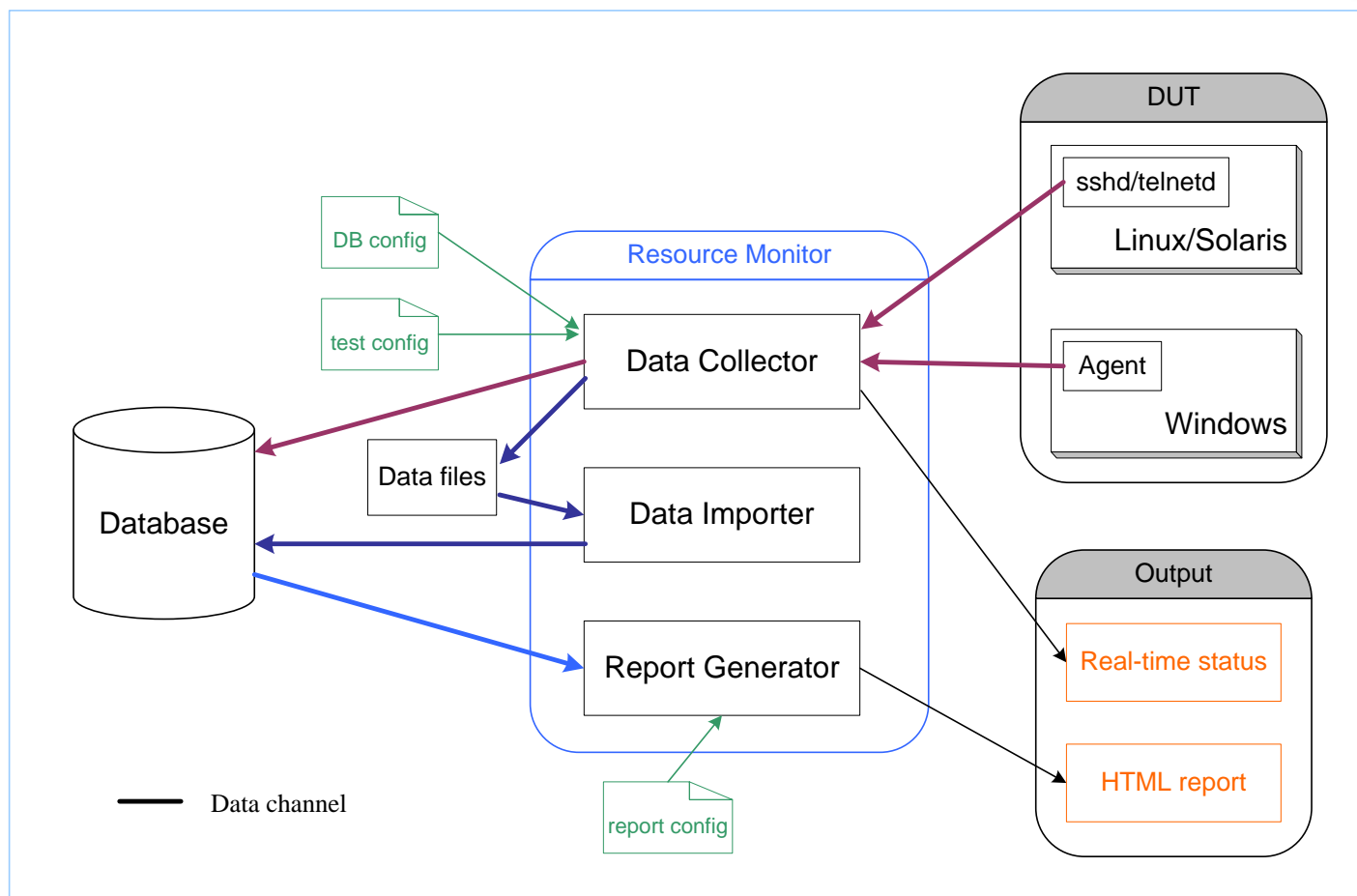
- Windows: Performance Monitor
- Linux: vmstat, iostat, sar, top, /proc,

但是：

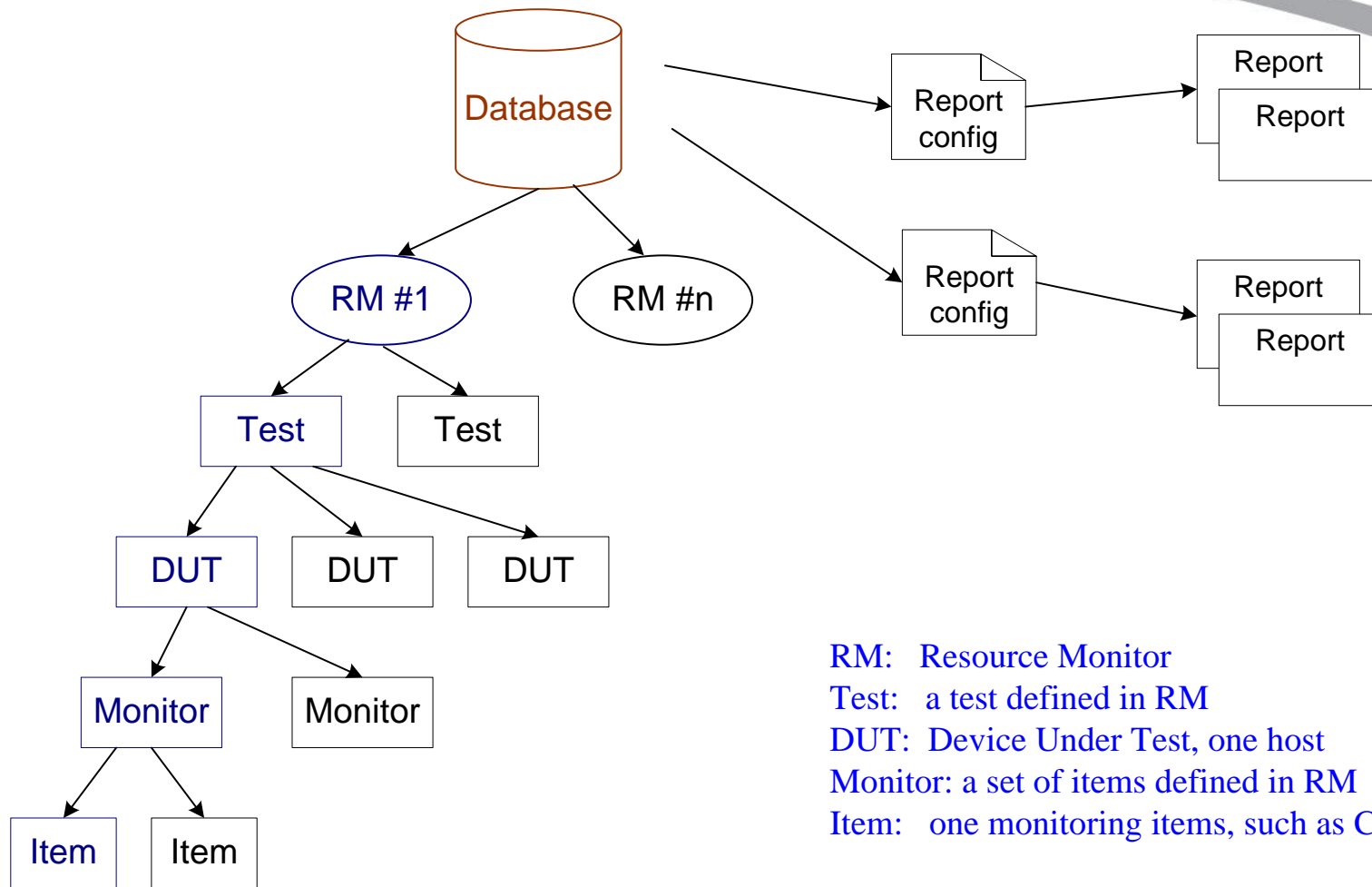
- 数据比较零散难以合并
- 需手动生成图表
- 产品相关的数据
- 同时监控多台机器

需要一些定制化的工具！





Resource Monitor



RM: Resource Monitor
Test: a test defined in RM
DUT: Device Under Test, one host
Monitor: a set of items defined in RM
Item: one monitoring items, such as CPU usage

Resource Monitor – Data Organization





产品相关的指标

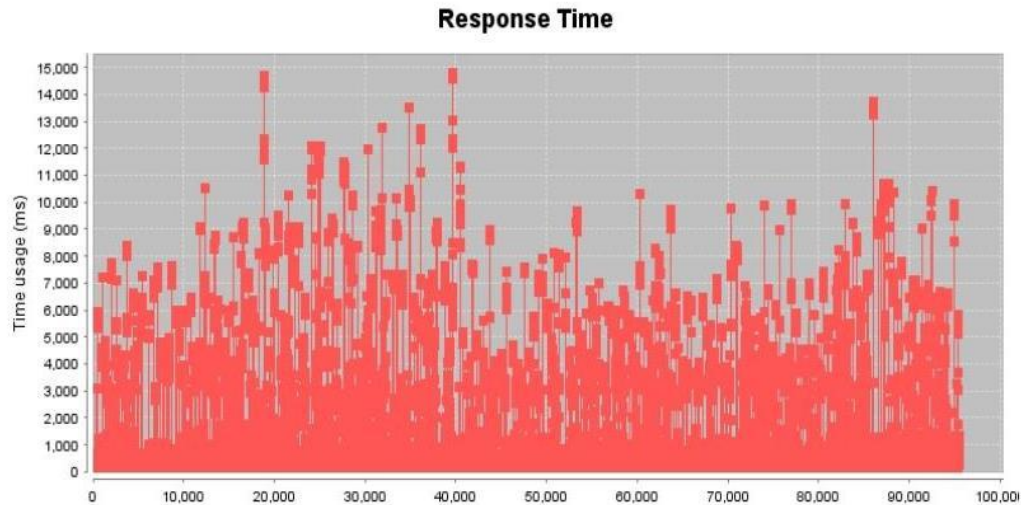
Example - 获取指定进程的内存使用状况

```
[root@localhost ~]# cat /proc/`ps -ef|grep hello | grep -v grep | awk '{print $2}'`/status
Name:   hello
State:  S (sleeping)
SleepAVG:      88%
Tgid:    3424
Pid:     3424
PPid:    3398
TracerPid:     0
Uid:      0      0      0      0
Gid:      0      0      0      0
FDSize: 256
Groups: 0 1 2 3 4 6 10
VmPeak:   3040 kB
VmSize:   3040 kB
VmLck:      0 kB
VmHWM:    984 kB
VmRSS:    984 kB
VmData:   320 kB
VmStk:     88 kB
VmExe:      4 kB
VmLib:   2572 kB
VmPTE:     32 kB
StaBrk: 0804a000 kB

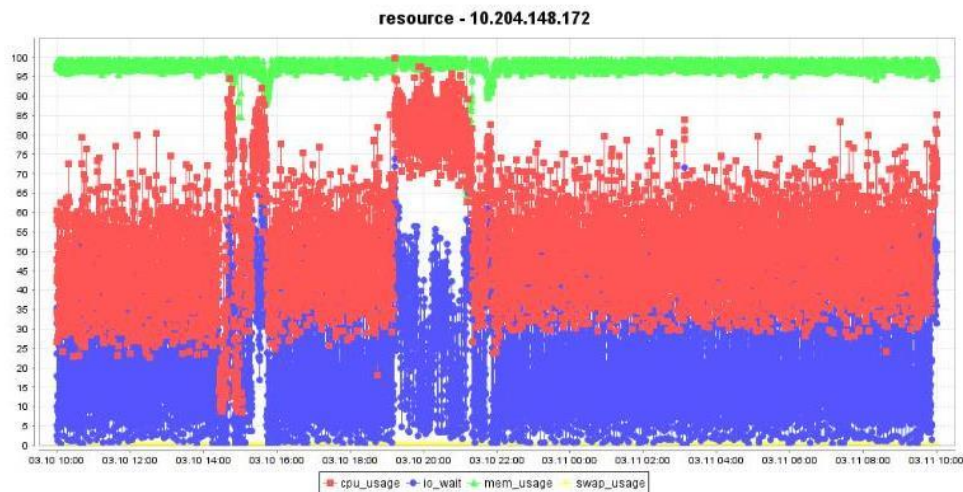
```

```
<command-parser>
  <name>AS4_Hello_Mem_Usage</name>
  <command>cat /proc/ ps -ef|grep hello | grep -v grep | awk '{print $2}'/status | grep -E
'VmSize|VmRSS|VmData'</command>
  <pattern>(\d+).\r\n.+\s+(\d+).\r\n.+\s+(\d+)</pattern>
  <type>VmSize:VmRSS:VmData</type>
  <expression>#1/1024:#2/1024:#3/1024</expression>
  <description>process mem usage (MB)</description>
</command-parser>
```

Time usage of one request: avg = 896.0, min = 0.0, max = 14782.0 (ms)



cpu_usage: avg=51.41, min=8.65, max=99.85
io_wait: avg=18.8, min=0.05, max=85.9
mem_usage: avg=97.62, min=62.79, max=99.53
swap_usage: avg=0.0, min=0.0, max=0.0



- 自动生成图表
- 同一个时间轴

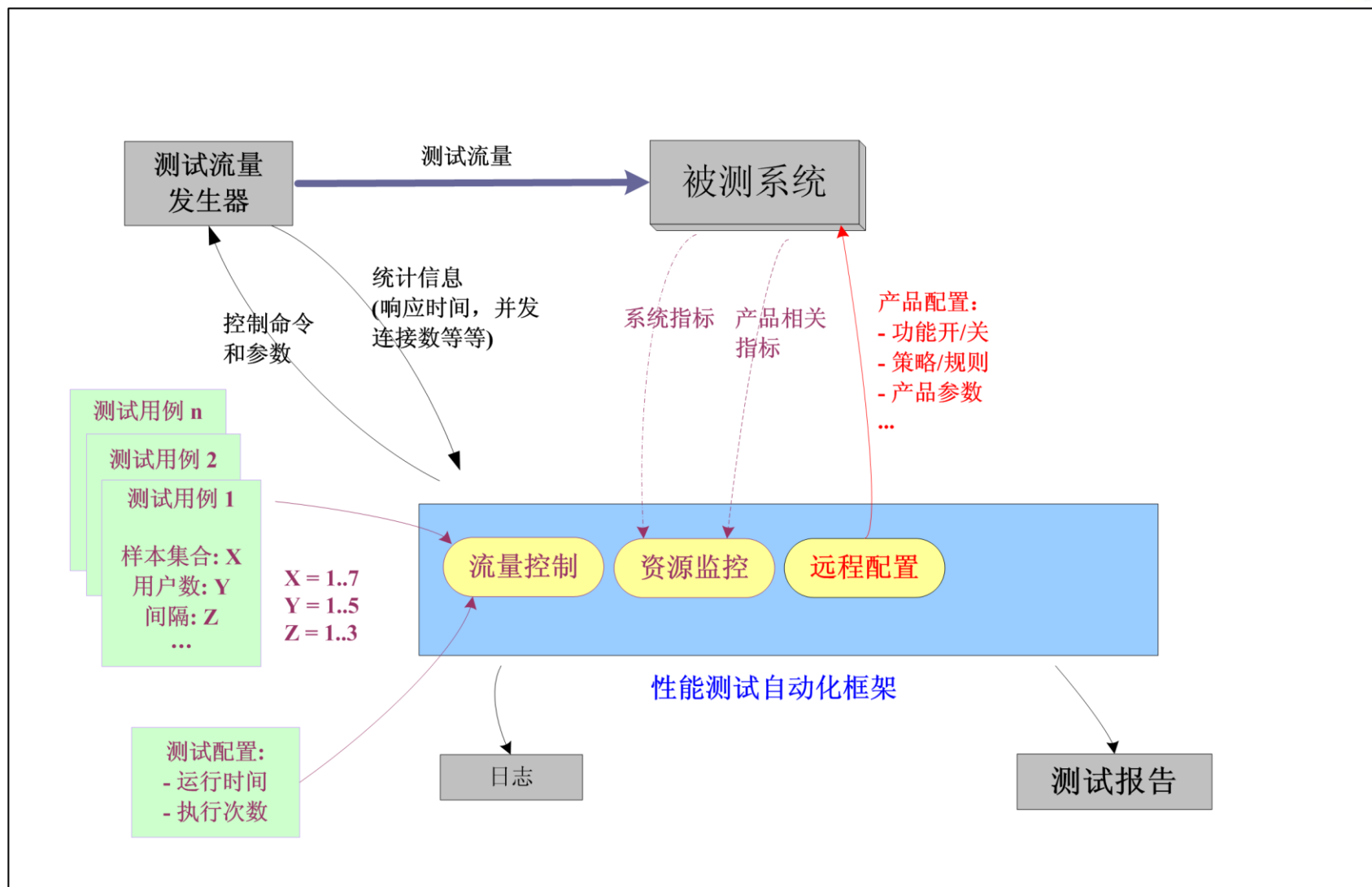
境界三： 让产品配置自动化

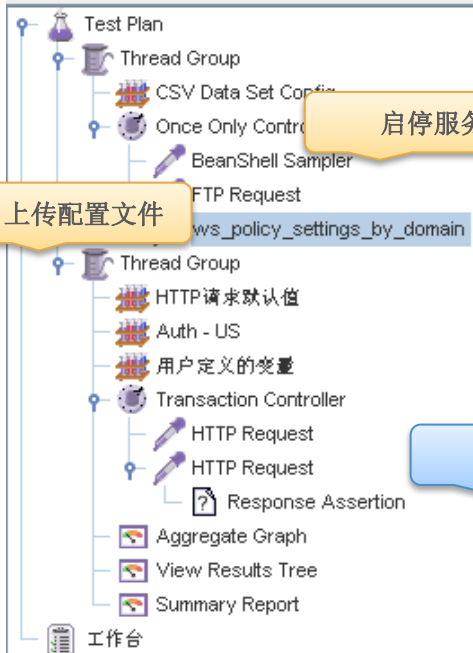
- 1. 产品自身配置
 - 功能的开关
 - 策略/规则
 - Cache的开关
 - ...

- 2. 系统的配置
 - Web服务器
 - 数据库
 - 系统资源
 - ...

可能的技术方案：

- ✓ 产品接口，比如Restful API
- ✓ Agent-based，比如STAF
- ✓ Agentless: SSH, telnet
- ✓ 虚拟化，API
- ✓ IPMI
- ✓ ...





启停服务

上传配置文件

HTTP请求

名称: ws_policy_settings_by_domain

注释:

Web服务器

Restful API

104.195.1

端口号

HTTP请求

协议: HTTP

方法: POST

Content encoding:

路径: /2.0/domains/\${domain_name}/settings.xml

测试流量

定向

☐ 跟随重定向

☒ Use KeepAlive

☐ Use multipart/form-data for HTTP POST

同请求一起发送参数:

名称:

值

境界四： 让产品的安装和部署自动化

可以完全借助于功能测试的自动安装部署模块

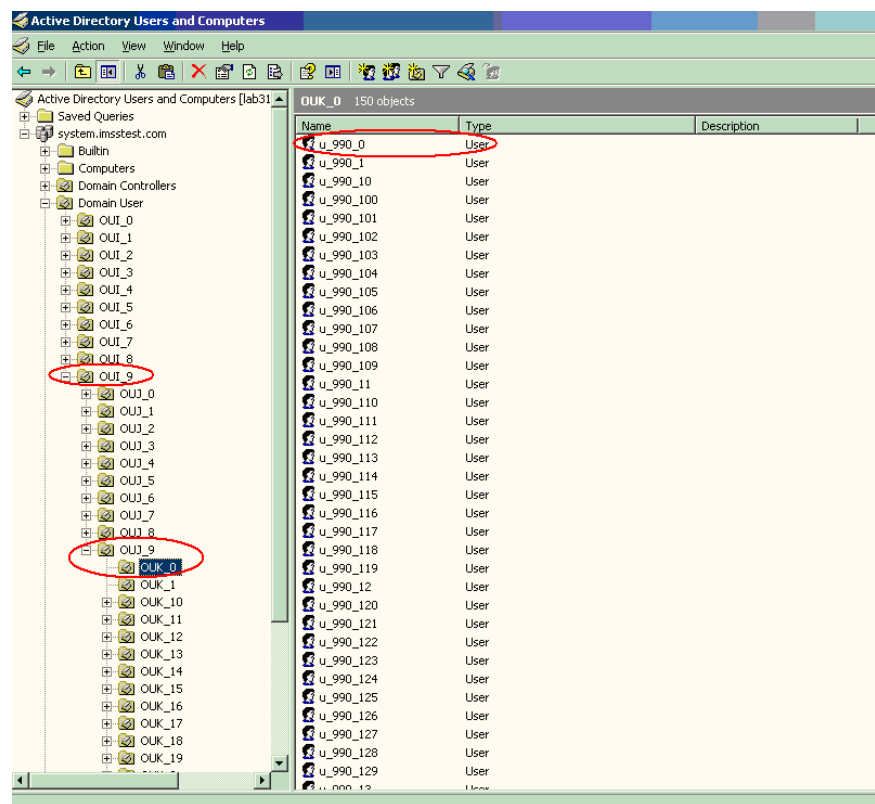
- daily build集成
- CI (静态检查 + 功能自动化 + 性能自动化)
- 自动的notification

Note:

- 在特定的阶段
- 针对特定的核心模块

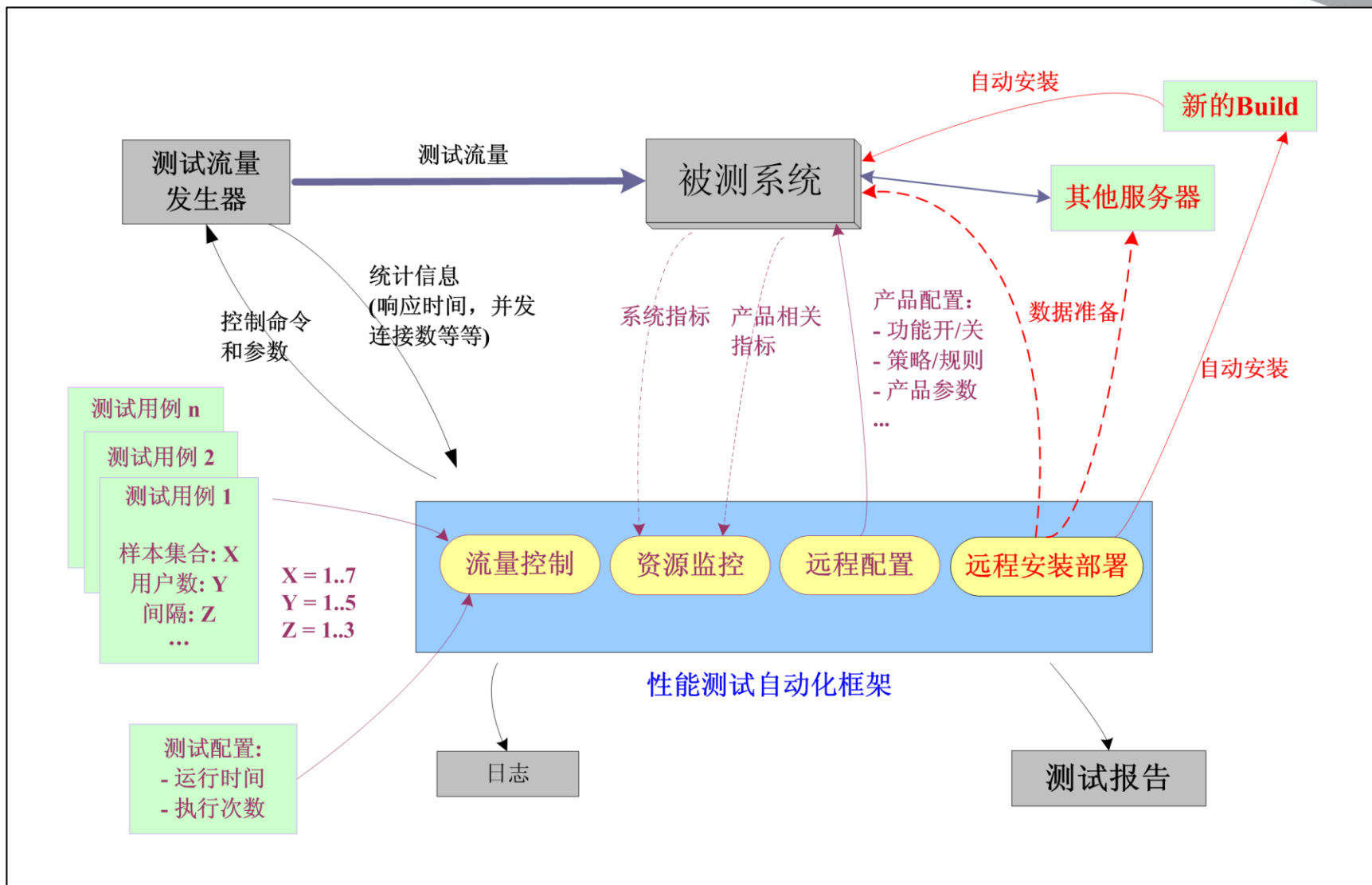
• 数据准备的自动化

- DB
- DNS
- LDAP
- ...



Note:

- 数据的真实性
- 高效
- 清除后无残留，对后续测试没有影响。



境界五：无人值守（只需按下快门）



性能测试自动化的好处

效果

- ✓ 提高性能测试的覆盖率
- ✓ 方便地应用于可扩展性测试
- ✓ 快速定位性能缺陷的引入

效率

- ✓ 提高性能测试的效率
- ✓ 有可重复性

人员和组织

- ✓ 对使用者的要求降低了，分为测试的设计者和执行者
- ✓ 性能测试的内部标准化

一些心得

需求 vs 技术



自动化不只是功能测试



不断检视自动化的测试系统



- 需求在变
- 产品在变
- 数据在变
- 硬件和软件都在升级

***Scott Barber: 10 Tips for Performance Test Automation,**
for Test Automation Day 2011, Zeist, NE

<http://www.perftestplus.com/presentations.htm>

Q&A

