



Securing Your Web World



Create for Taobao技术嘉年华 TCon 2011

懒人的幸福生活 — 无人值守的性能测试

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

懒人的哲学

- “世界如此的精彩都是拜懒人所赐。”
- “人类的很多发明都是因为想偷懒，比如…”

如何记录一个人的肖像？



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
 - ...



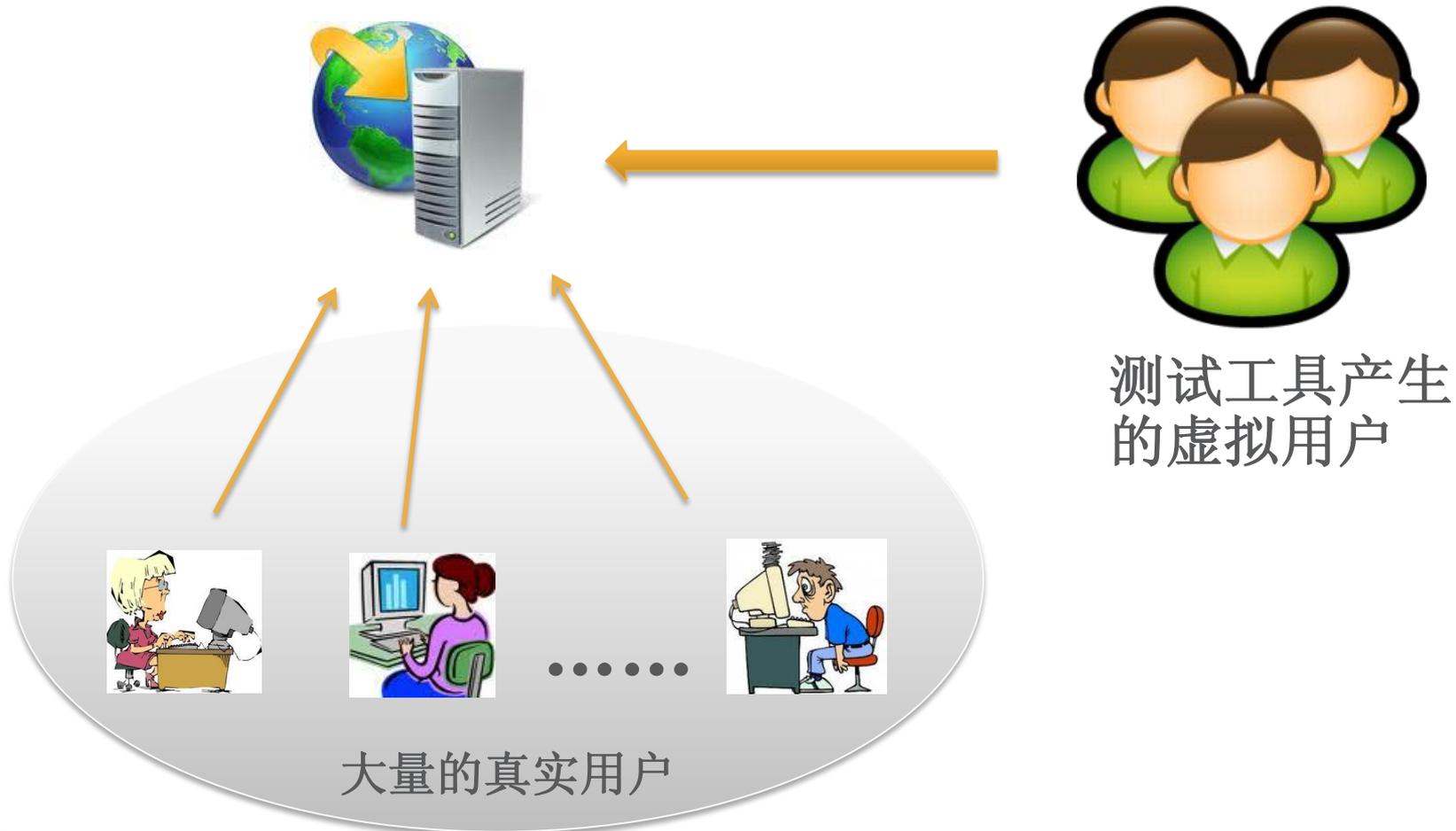
少数家庭用户
+ 旅游或者聚会

- 手动换片
- 手动测光
- 手动对焦
- 底片冲印



少数专业人士
正式场合

如何衡量一个系统的性能？



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
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
Anti-Spam Only <i>Only AS active</i>	Single CPU	Standalone <i>Scanner, Policy and Database on same server</i> 样本	1	400
			5	110
			15	934
			28	898
			28 (Mixed)	154
			53	169
			53 (Mixed)	533
			100	603
			500	365
Content Filter Only <i>Only CF active</i>	Dual CPU	Distribution <i>Scanner, Policy and Database on Different Server</i>	1	808
			5	969
			15	603
			28	579
			28 (Mixed)	659
			53	929
			53 (Mixed)	568
			100	287
			500	287
Default Configuration <i>AV, AS and CF active</i>	Dual CPU	Distribution <i>Scanner, Policy and Database on Different Server</i>	1	262
			5	942
			15	559
			28	937
			28 (Mixed)	525
			53	476
			53 (Mixed)	251
			100	888
			500	888

配置 x 5

硬件 x 3

部署 x 2

样本 x 9

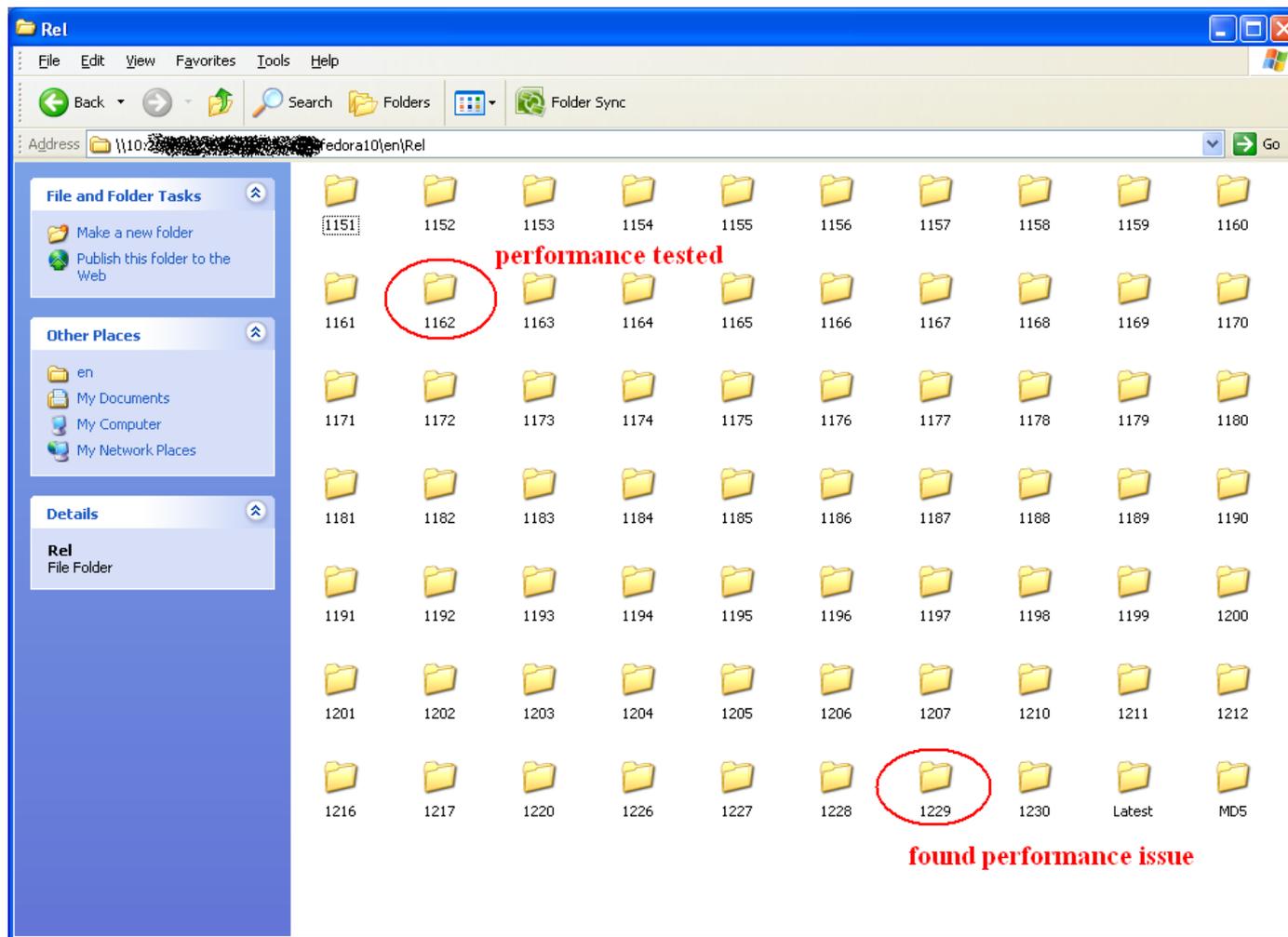
270 次测试!

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

不符合懒人哲学！

• 懒人的性能测试之5大境界

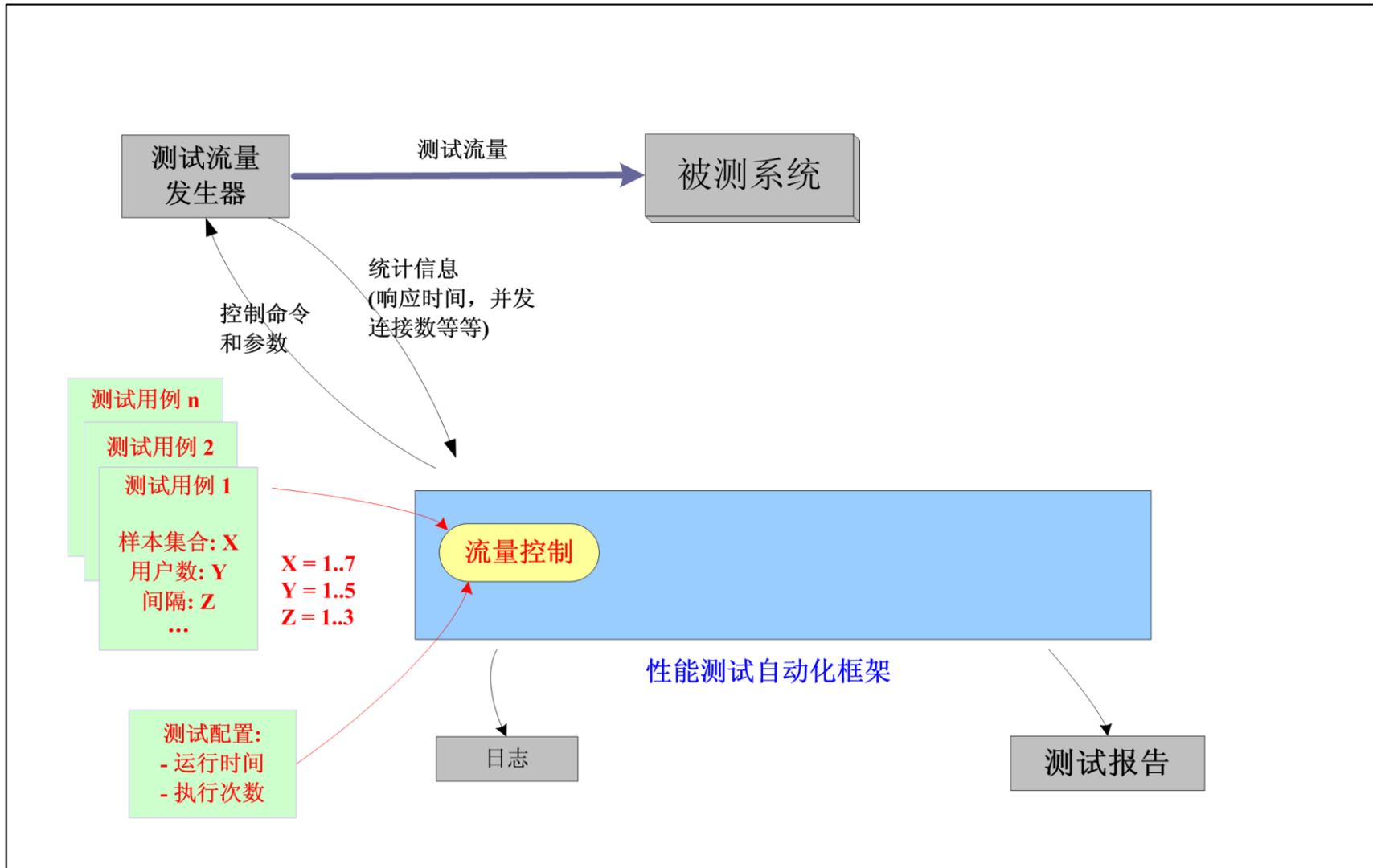
境界一： 让不同用例的执行自动化

➤ 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



This report was generated by PerfTest V3.24 at 2010-11-08 13:07:30.

Tester: Ricky Qiu
 Test Start at: 2010-11-07 17:22:26
 Test End at: 2010-11-08 13:07:29

~20 hours

Test Config
 Command Executor: Before_Test = Y, After_Test = Y, Before_Every_Case = Y, After_Every_Case = Y
 Remote Monitor: CPU/Mem = Y, CPU Load = Y, Queue = Y, Work Process = Y
 DUT: 192.168.21.163

~40 min/run

Test Result

Case	Thread	No	Upstream Result					Downstream Result				
			total	fail	size	KBps	Rsp(ms)	msg/s	received	size	KBps	msg/s
PerfTest_ISS_53KB_10W_T50	50	1	101031	20	7.9	295.86	318.0	37.45	60132	100.84	1529.91	15.17
		2	101026	20	7.9	289.26	348.0	36.61	60124	100.93	1496.96	14.83
		3	101028	19	7.9	283.78	374.0	35.91	60129	100.94	1469.25	14.55
		4	101031	20	7.9	284.53	372.0	36.01	60130	100.85	1471.62	14.59
		5	101037	19	7.9	284.03	375.0	35.95	60129	100.89	1469.28	14.56
PerfTest_ISS_53KB_10W_I70	70	1	101029	20	7.9	347.47	564.0	43.98	60161	100.7	1793.38	17.8
		2	101035	18	7.9	345.09	574.0	43.67	60187	100.84	1781.74	17.66
		3	101035	852	7.41	320.44	581.0	43.22	59728	100.99	1764.4	17.47
		4	101035	20	7.9	345.32	574.0	43.71	60153	100.87	1785.01	17.69
		5	101035	20	7.9	336.17	615.0	42.55	60138	100.9	1739.32	17.23
PerfTest_ISS_53KB_10W_I90	90	1	101033	17	7.9	375.55	849.0	47.53	60235	100.8	1936.24	19.2
		2	101035	18	7.9	378.7	832.0	47.93	60288	100.82	1949.85	19.33
		3	101031	19	7.9	374.08	853.0	47.34	60244	100.77	1927.07	19.12
		4	101031	20	7.9	376.5	842.0	47.65	60232	100.52	1935.53	19.25
		5	101030	20	7.9	370.46	872.0	46.89	60224	100.83	1911.16	18.95
PerfTest_ISS_53KB_10W_T100	100	1	101031	19	7.9	388.64	979.0	49.19	60271	100.56	1994.57	19.83
		2	101032	20	7.9	386.91	990.0	48.97	60382	100.27	1972.96	19.67
		3	101031	20	7.9	391.59	965.0	49.56	60253	100.94	2007.95	19.89
		4	101026	18	7.9	389.79	975.0	49.33	60264	100.59	1998.24	19.86
		5	101033	19	7.9	388.51	981.0	49.17	60287	100.63	1996.26	19.83
PerfTest_ISS_53KB_10W_T120	120	1	101028	20	7.9	400.92	1299.0	50.74	60488	100.63	2055.13	20.34
		2	101036	18	7.9	385.13	1395.0	48.74	60488	100.63	2055.13	20.34
		3	101030	19	7.9	404.41	1282.0	51.18	60488	100.63	2055.13	20.34
		4	101024	20	7.9	397.94	1316.0	50.36	60398	100.81	2039.72	20.23
		5	101032	20	7.9	395.56	1329.0	50.07	60275	100.74	2026.6	20.11
PerfTest_ISS_53KB_10W_T150	150	1	101035	19	7.9	347.23	2327.0	43.95	60310	100.71	1485.04	14.74
		2	101026	20	7.9	323.97	2572.0	41.0	60342	100.99	1387.18	13.73
		3	101023	20	7.9	323.12	2582.0	40.89	66010	101.56	1327.93	13.07
		4	101025	20	7.9	320.98	2603.0	40.62	67736	101.03	1327.28	13.13
		5	101023	18	7.9	317.71	2643.0	40.2	68433	100.83	1304.09	12.93

sample/load

多次执行

性能的拐点

境界二：让资源监控自动化

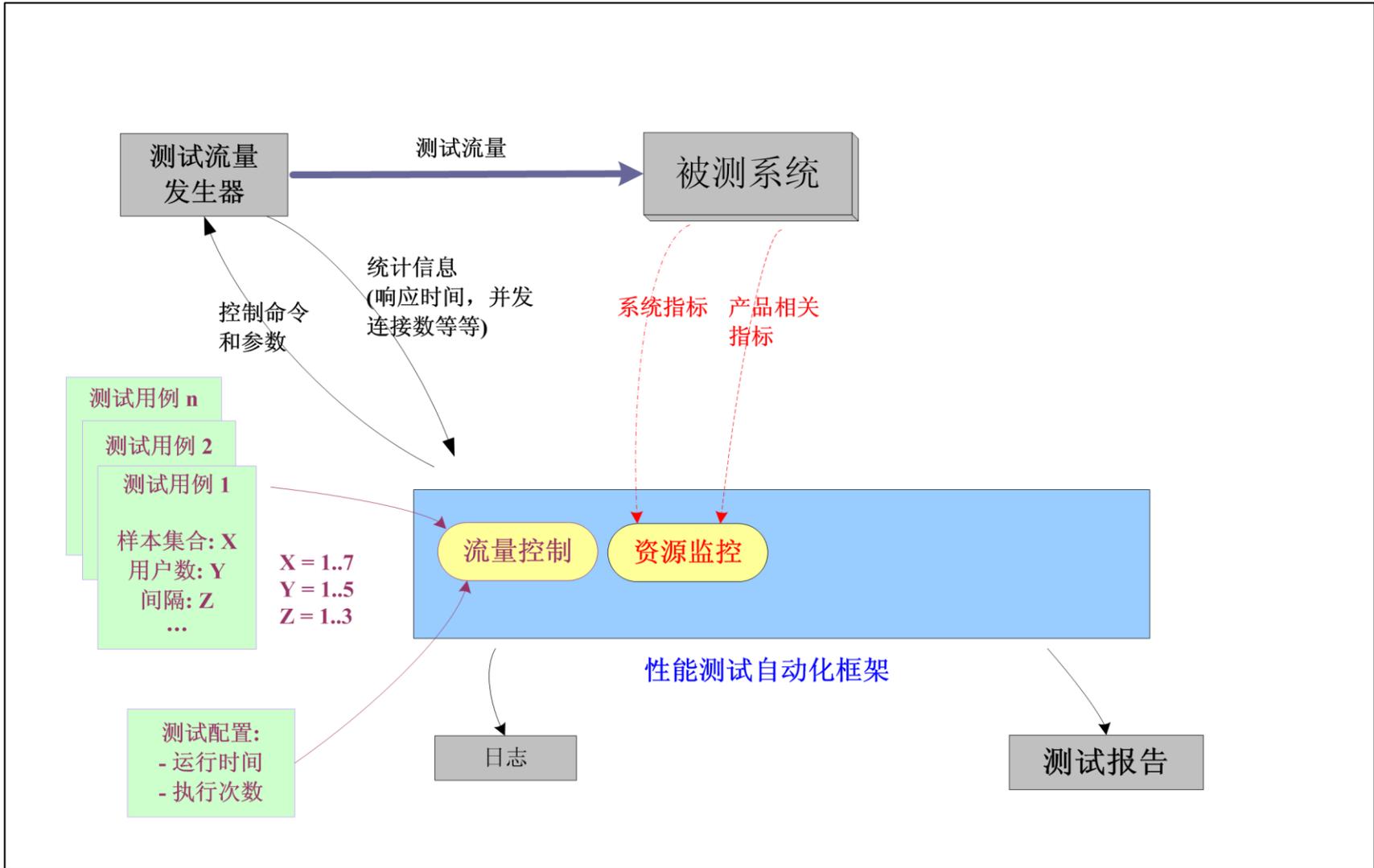
我们有办法采集到数据：

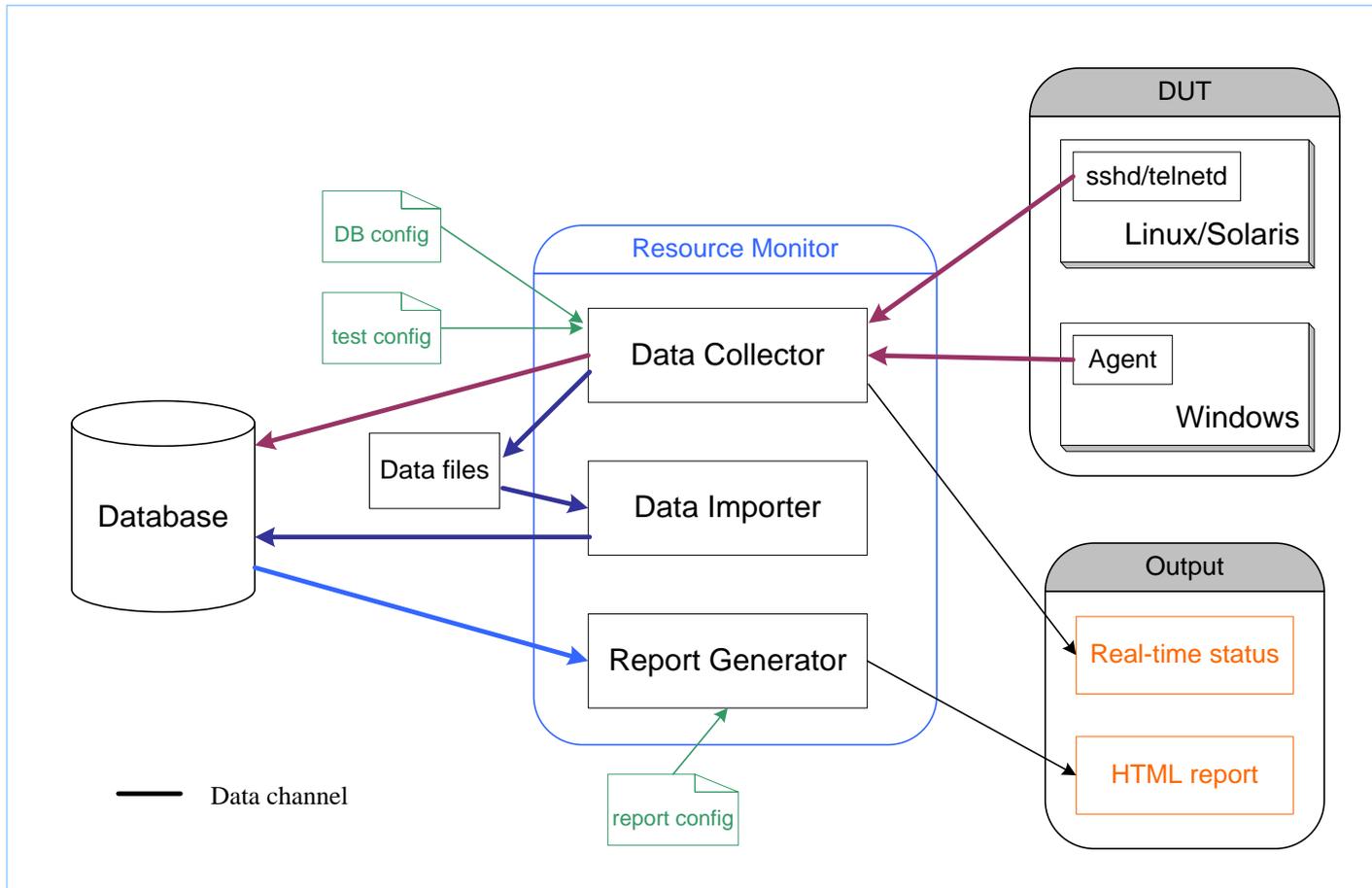
- Windows: Performance Monitor
- Linux: vmstat, iostat, sar, top, /proc,

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

多组测试

performance_alpha

多台机器

10.204.168.207 10.204.168.208 10.204.168.209

DUT Information

Hostname of DUT : localhost.localdomain
 Running time of DUT : 4 days, 1:55
 Current date of DUT : Mon Apr 27 17:20:52 CST 2009

系统标准数据

CPU Usage:IO Wait

Interval : 10000 ms Status : Running Show Chart

Data : 40.4 19.56

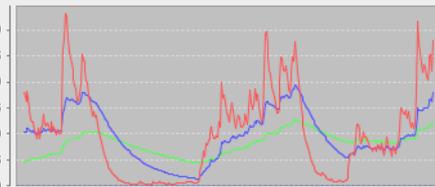


cpu_us

CPU load 1 minute:5 minutes:15 minutes

Interval : 10000 ms Status : Running Show Chart

Data : 28.0 17.88 12.42

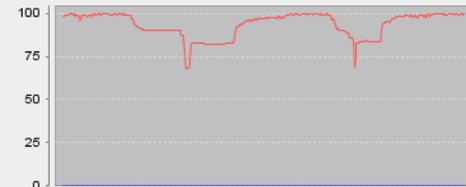


cpu_load1 cpu_load5 cpu_load15

Memory Usage:Swap Usage

Interval : 10000 ms Status : Running Show Chart

Data : 99.39 0.0



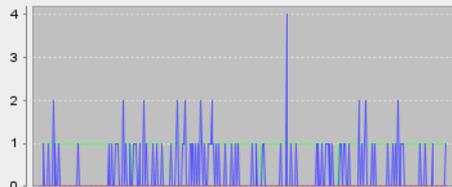
mem_usage swap_usage

自定义数据

Queue Status

Interval : 10000 ms Status : Running Show Chart

Data : 0.0 1.0 1.0



defer active incoming

private mem of [redacted] in KB

Interval : 30000 ms Status : Running Show Chart

Data : 344868.0

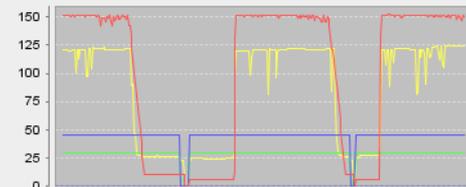


mem_private

[redacted] related process number

Interval : 10000 ms Status : Running Show Chart

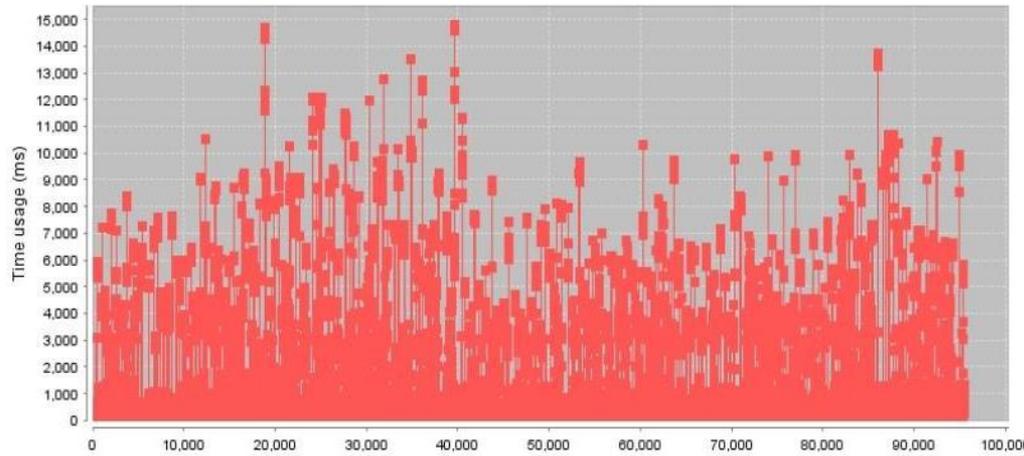
Data : 150.0 46.0 30.0 124.0



httpd [redacted] postgre

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

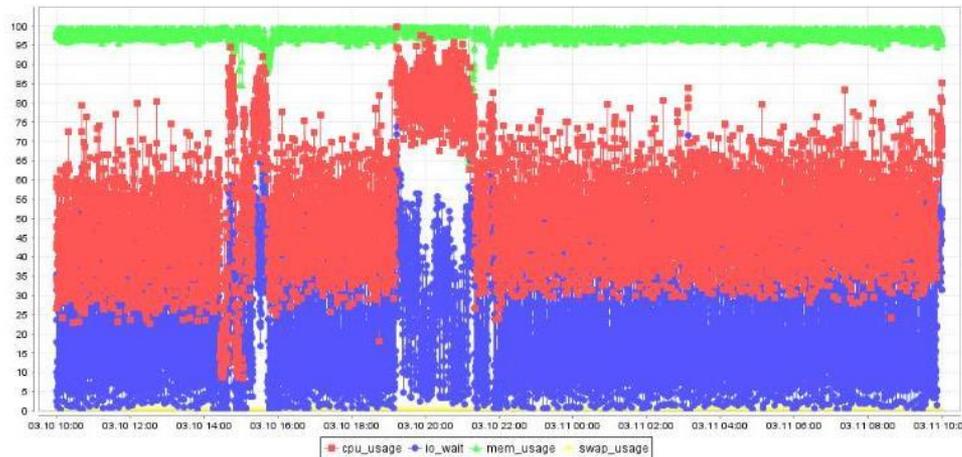
Response Time



- 自动生成图表
- 同一个时间轴

```
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
```

resource - 10.204.148.172

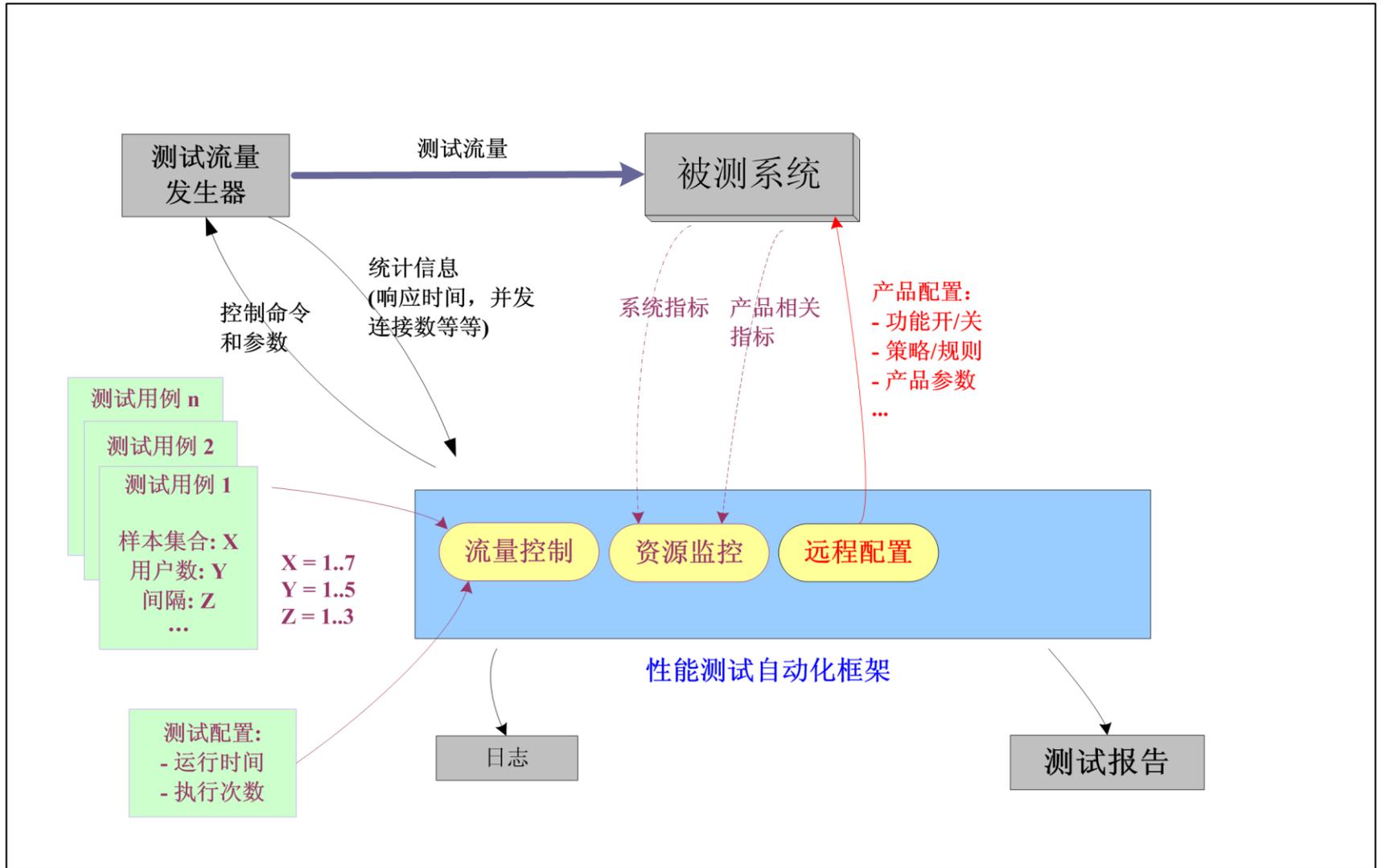


境界三： 让产品配置自动化

- 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

Restful API

测试流量

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

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

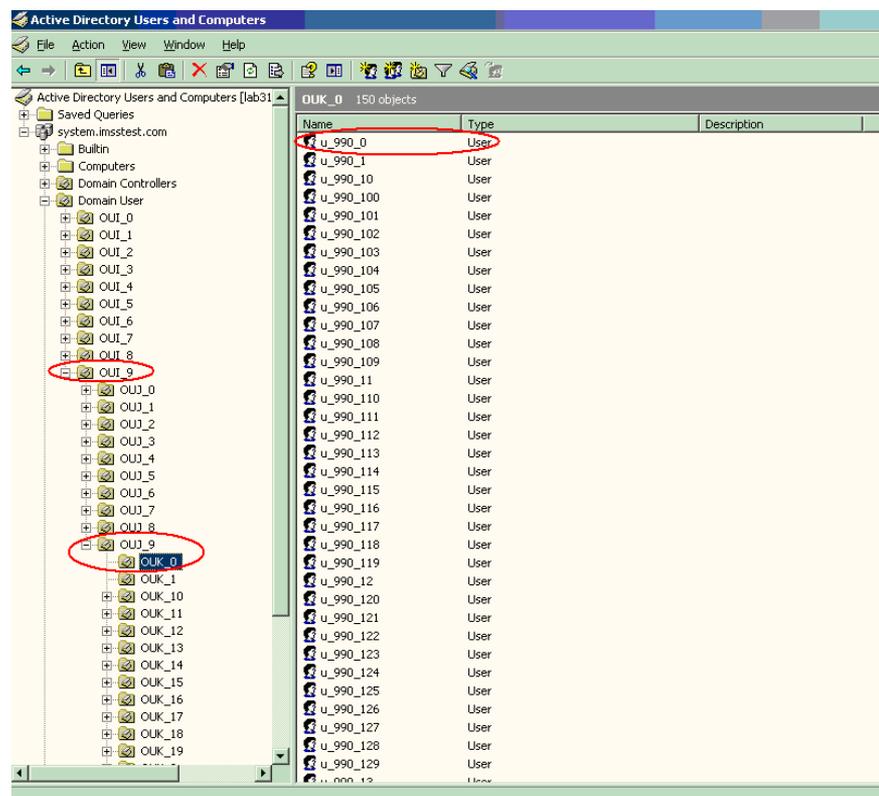
- daily build集成
- CI (静态检查 + 功能自动化 + 性能自动化)
- 自动的notification

Note:

- 在特定的阶段
- 针对特定的核心模块

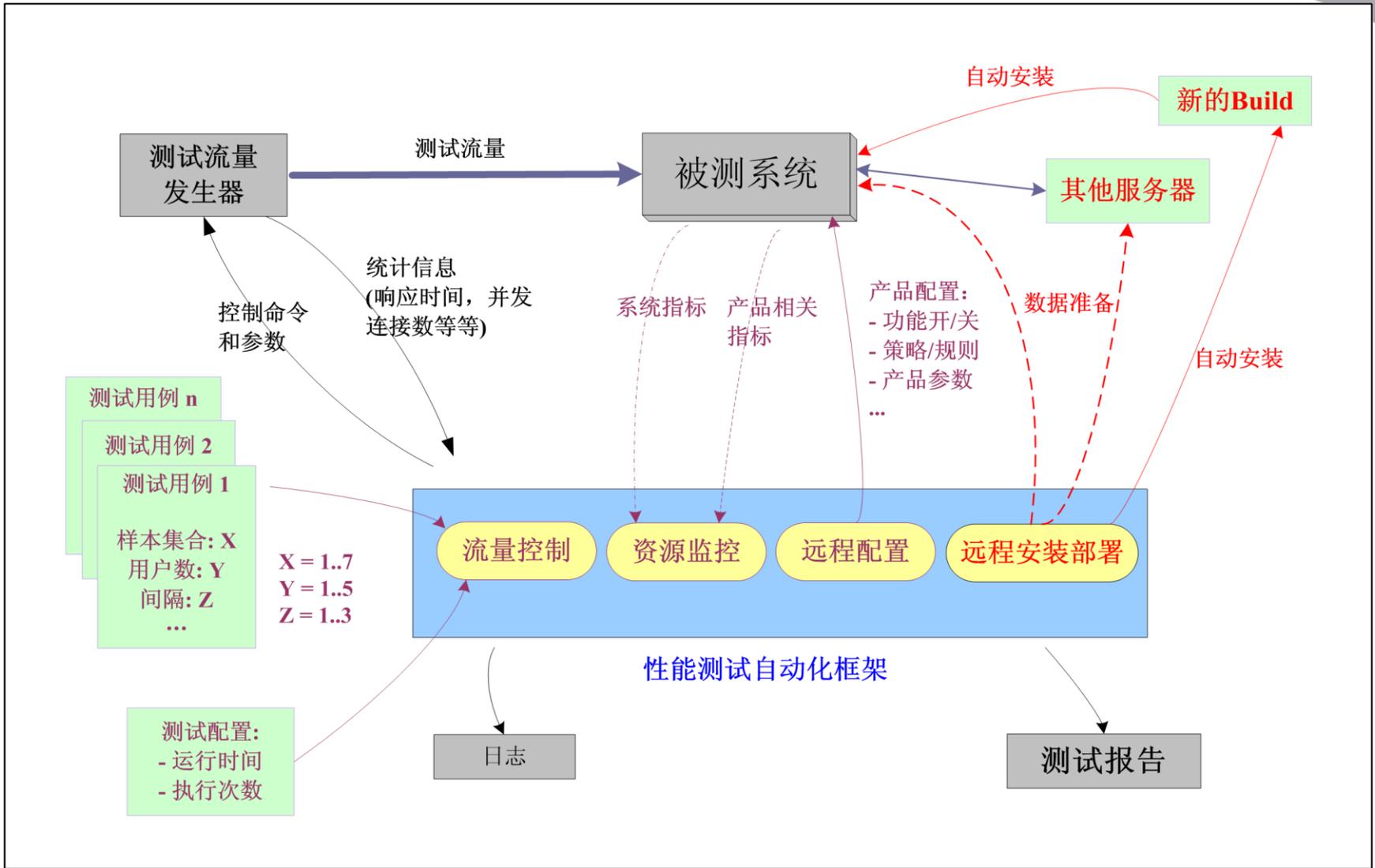
• 数据准备的自动化

- 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）

安
装
部
署

配
置

执
行

监
控

分
析
报
告

发起请求

查看测试结果

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

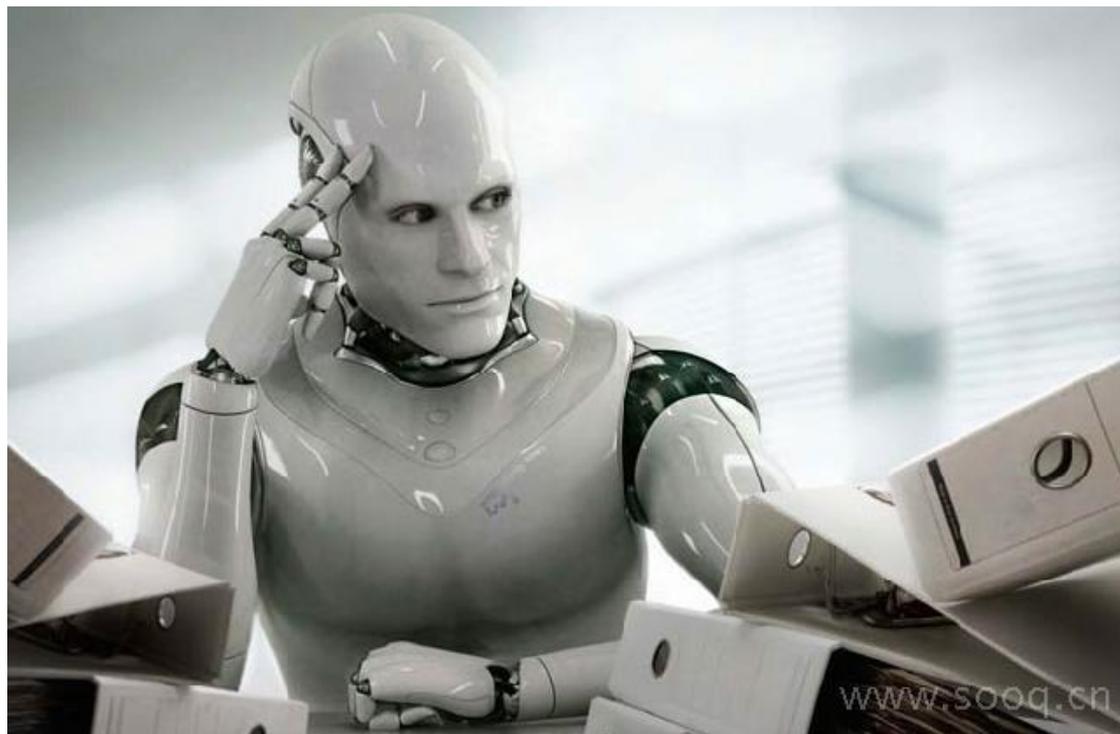
- ✓ 对使用者的要求降低了
- ✓ 性能测试的内部标准化

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

