

压测工具Tsung介绍

褚霸 chuba@taobao.com
2010/07/15

Tsung是什么？



开源的, 多协议的, 分布的压力测试工具.

设计Tsung的目的是模拟真实用户测试基于IP的服务器程序的伸缩性和性能

为什么要用Tsung

- 传统压力测试工具要收费, 而且很贵, 难以扩展, 无法很好的集群形成大的压力.
- Tsung内置支持 HTTP, WebDAV, SOAP, PostgreSQL, MySQL, LDAP and Jabber/XMPP协议.
- Tsung可以同时模拟上百万用户, 如果你有够多的机器的话
- Tsung稳定可靠.

谁在用Tsung

国外:

- *DGI* (Direction Générale des impôts): French finance ministry
- *Cap Gemini Ernst & Young*
- *IFP* (Institut Français du Pétrole): French Research Organization for Petroleum
- *LibertySurf*
- Sun™ for their Mooddlerooms platform on Niagara processors

国内:

- 华为
- SNS网站

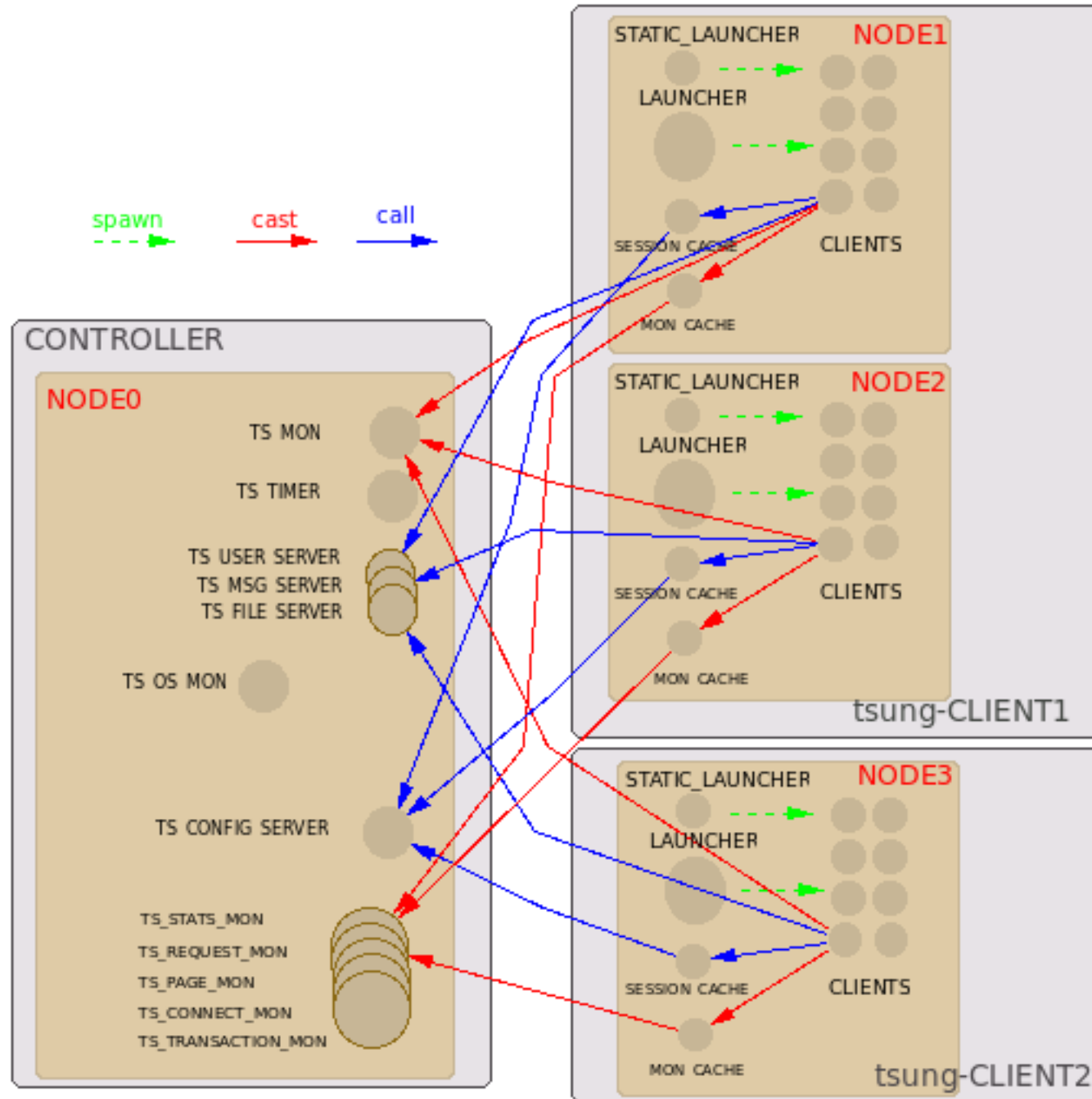
Tsung的特性

- 高性能
- 插件机制支持多协议
- 利用OS IP别名技术在单个机器上使用多IP
- 服务器监控(CPU, 内存, 网络流量), 支持SNMP, Erlang, MUnin方式
- XML方式的配置文件
- 模拟真实的流量, 虚拟用户的发呆时间和抵达律使用随机的概率分布
- HTML或者图表的方式报告测试结果

Tsung的亮点

- 高性能和分布Benchmark
- 易于使用
- 多协议支持
- 监控目标服务器, 方便找出瓶颈

Tsung内部结构图



Tsung测试流程 三步走

配置录制或者编写

- tsung-recorder

执行压测

- tsung <options> start|stop|debug|status

统计报告

- tsung_stats.pl

Tsung压测对象

网路服务器 支持TCP, UDP, SSL

其他的对不起目前还不支持.

理解配置文件

XML格式的

场景描述都是通过这个配置文件

整个集群只用一份

Tsung支持的协议

- Raw
- HTTP 支持的很好
- WebDAV
- SOAP
- PostgreSQL
- MySQL 比较粗
- LDAP
- Jabber/XMPP

Client配置和选择

最简单的情况

- `use_controller_vm="true"`

`maxusers`参数: 决定了一个节点能够支持的最多虚拟用户数, 超过了这个数目, 自动开启更多的节点.

IP别名技术, 模拟多个机器, 目的有二个:

- 方便服务器作Loadbalance
- 解决单个IP的端口65535的问题

支持从Job调度器获取Client地址

`Cpu`参数: 如果设置的话, Tsung将按照设置的数目启动`beam.plain`. 否则将启动`beam.smp`.

`Weight`参数: 决定了每个节点上面用户的比例

配置选项

thinktime

tcp_snd_buffer 设定TCP发送缓冲区 tcp_rcv_buffer

hibernate 开启大大节约内存

定义Load

虚拟用户的形式：动态和静态

不同的用户，不同的进入频度

指定Load执行总的时间

用户模拟

一个用户一个Erlang进程, 可以模拟成千上万的用户.

行为模拟, 通过随机发呆时间来模拟真实流量

静态用户可以控制进入点

动态用户由系统定期产生

定义Session

Sessions define the content of the scenario itself. They describe the requests to execute.

Each session has a given probability. This is used to decide which session a new user will execute. The sum of all session's probabilities must be 100.

A transaction is just a way to have customized statistics.

支持请求循环

定义Request

典型的HTTP请求

```
request<http url="/" method="GET" version="1.1"></http>  
</request>
```

典型的Mysql请求

```
<request><mysql type="sql">SHOW  
TABLES</mysql></request>
```

产生差异性

- 动态替换 `%%Module:Function%%`,前提是设置了 `subst="true"` 参数
- 从外部csv格式文件获取
- 动态变量从httpHeader里面获取
- 从请求的响应中获取

压力产生

根据用户进入的频度, 在单位时间内产生一定数量的用户.

根据Client的配置情况,生成不同的节点, 然后把用户按照节点的能力和配置, 分配出去.

每个用户运行顺序执行特定的session.

每个session包括请求和Thinking, 也就是说用户不是在请求就是在thinking.

thinking的时候是不占用CPU的, 所以我们可以支持很大数量的用户.

统计报告

- `tsung_stats.pl`
- `tsplot.pl`
- `tsung-rrd.pl`

统计图表自定义

tsung.log是csv格式文本文件,可以很方便自己来解释

对服务器的监控

支持Erlang, snmp, Munin方式
获取CPU, 内存, 网络流量的使用情况
方便找出瓶颈

插件机制

不支持我家的协议, 怎么办, 怎么办?

如何写插件, 三步走, 无需关心网络和分布:

- getmessage
- parse
- 处理动态替换

故障排除

- ssh信认配置
- 文件句柄用尽了
- 执行结果不对?`dumptraffic="true"`
- Erlang版本以及安装路径一致

硬件和操作系统的选择

- 内存要大,越大越好
- CPU核心要多,越多越好
- 操作系统, RHEL 5挺好的
- 协议栈, OS vm微调
- 32位和64位的选择
- Erlang 版本的选择, Halfword?

谢谢大家

Any question?