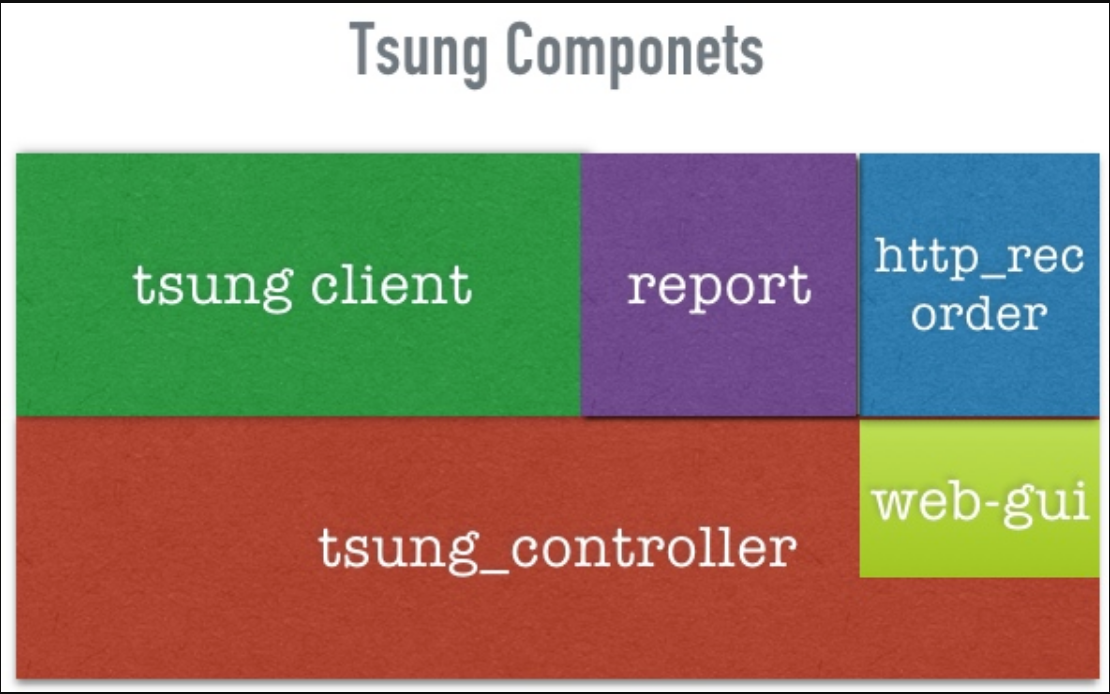
# Tsung 压力测试

## Tsung介绍

Tsung是一个**开源、多协议、分布式**的压力测试工具。它可以用来测试HTTP, WebDAV, SOAP, PostgreSQL, MySQL, LDAP 和Jabber/XMPP的服务器，并且**支持扩展协议。**

Tsung组成



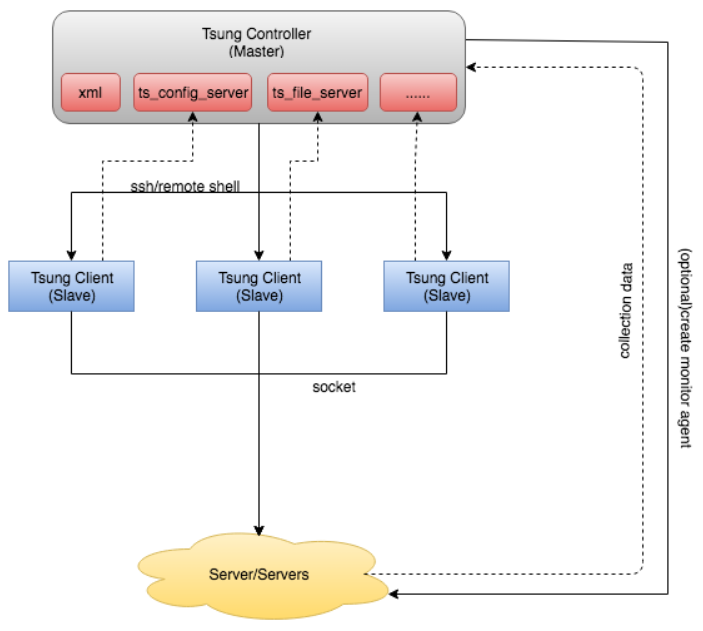
Tusng工作原理

1. Tsung的每一个虚拟用户就是一个erlang的轻量进程。
2. 虚拟用户完成session后就消失。
3. 大量的虚拟用户（erlang轻量进程）建立在erlangVM上。
4. 一台测试机可以启多个erlangVM，目前按照1个cpu启动1个erlangVM。

分布式部署Tsung在运行时，会产生ssl\_esock、beam、beam.smp3种进程，ssl-esock是erlangVM用的port程序， 协助完成ssh功能；beam是单处理器版本的erlangVM；beam.smp就是control程序，负责协调系统的运作。

Tsung主从模型

设定环境为分布式环境下Tsung集群，下面简单梳理一下主、从节点启动流程。



流程大致说明：

* 主节点（tsung\_controller）通过SSH或其它远程终端（后面会讲到操作更为轻量的完全替代SSH方式）连接到从服务器启动tsung从节点运行时环境
* 主节点RPC批量启动tsung client进程
* 主节点为每一个从节点启动会话监控，控制会话速度，开启ts\_client模拟终端
* 从节点请求主节点具体业务进程，获取会话指令以及会话具体内容
* 从节点建立到目标压测服务器的SOCKET网络连接，开始会话
* 主节点可以通过SSH/其它终端方式连接到目标压测服务器，启动从节点，然后收集数据（可选，具体细节，后续文字会讲到）

## 使用docker安装Tsung

为了方便Tsung部署，我制作了一个docker镜像，通过此docker镜像，能快速部署Tsung，进行压力测试的时候，只需要编写好配置文件。

相关文件说明如下：



上图中tsung目录下文件说明如下：



进行压力测试

将上图中的tsung文件夹和tsung-compse.yml 上传到服务器，根据需求编辑上图中的tsung.xml配置文件，在tsung-compose.yml所在的目录下，执行如下命令：docker-compose –f tsung-compose.yml build tsung\_prima && docker-compose –f tsung-compose.yml up –d ，将启动测试，然后打开浏览器，访问服务器的8091端口，可以看到测试报告，刚开始看到的不一定是最终的结果，需要过一段时间刷新，直到完成了压测目标。

Tsung 相关概念

Tsung 安装后，会生成两个命令文件

1. tsung
2. tsung-recorder

tsung命令用于压力测试，tsung-recorder命令用于录制压力测试脚本，我们主要使用的是tsung命令。

在启动 tsung 之前，我们需要一个 XML 格式的配置文件，tsung会根据该配置文件进行测试。/usr/share/doc/tsung/examples 目录下有一些 XML 配置文件范例。XML配置文件可以使用-f参数指定，如果不指定，则默认使用 ~/.tsung/tsung.xml

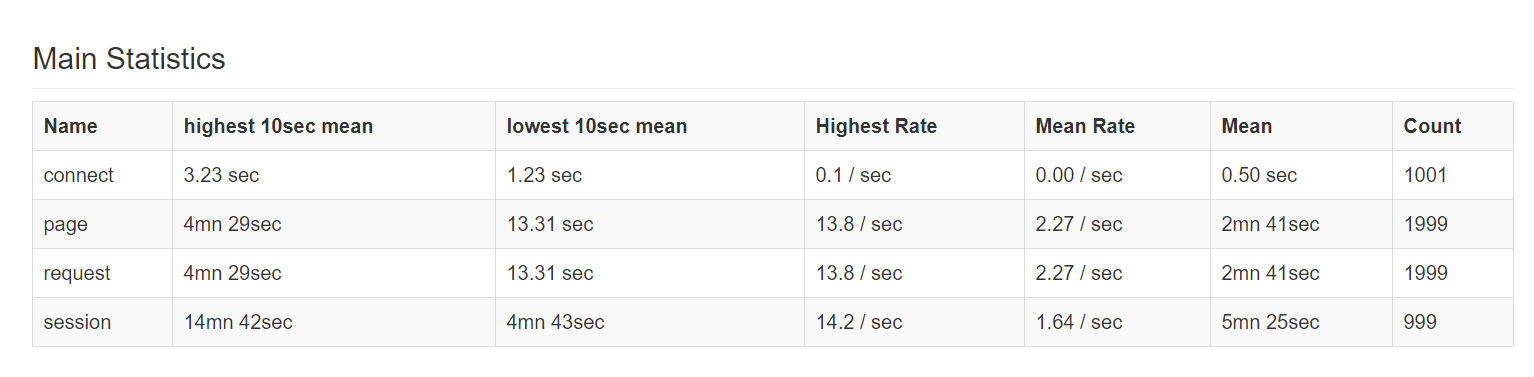
tsung 测试结束后，测试结果Log 默认保存在目录 ~/.tsung/log/ 下，也可以使用-l 参数指定文件夹，每启动一个新的测试时，会在这个目录下面创建一个新的子目录，用以保存测试的数据，Log 格式为当前日期和时间的组合，例如：~/.tsung/log/20160217-0940。默认情况下，控制节点会启动一个嵌入的网站服务器，侦听8091端口，测试结束该服务器会关闭，使用-k 参数可以保持不关闭，通过访问该服务的8091端口可以看到测试结果。

测试结束之后，可以使用tsung\_stats.pl生成报表，例如：cd ~/.tsung/log/20160305-0933/ && /usr/lib/tsung/bin/tsung\_stats.pl ，但是在生成前需要确保安装了如下软件：yum install gnuplot && yum install perl-Template-Toolkit 。之后将整个文件夹打包发给他人，他人点击report.html 查看报表。

## Tsung配置文件

查看tsung.xml文件

## Tsung报表说明



Main Statistics: 主要统计数据

connect: 建立连接过程的时间

highest 10sec mean: 连接最长持续时间

lowest 10sec mean: 连接最短持续时间

Highest Rate: 最高连接速率

Mean Rate: 平均连接速率

Mean: 平均每个连接持续时间

Count: 总连接数

page: 表示一组请求集合的响应时间,（一个页面是不包含“思考时间”的一组请求）

request: 每个请求的响应时间

highest 10sec mean: 请求最长响应时间

lowest 10sec mean: 请求最短响应时间

Highest Rate: 响应请求最高速率

Mean Rate: 平均响应请求速率

Mean: 平均响应请求时间

Count: 总响应次数

session: 表示每个用户会话持续时间

highest 10sec mean: 会话最长持续时间

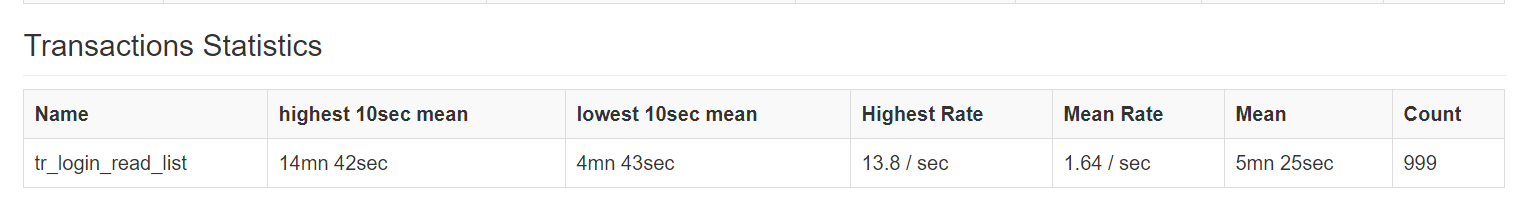
lowest 10sec mean: 会话最短持续时间

Highest Rate: 会话最高速率

Mean Rate: 平均会话速率

Mean: 平均会话持续时间

Count: 总会话次数



Transactions Statistics: 事物统计数据

highest 10sec mean: 事务最长花费时间

lowest 10sec mean: 事务最短花费时间

Highest Rate: 事务最高完成速率

Mean Rate: 事务最低完成速率

Mean: 事务平均花费时间

Count: 总事务数



Network Throughput: 网络流量

size\_rcv: 表示响应请求数据量

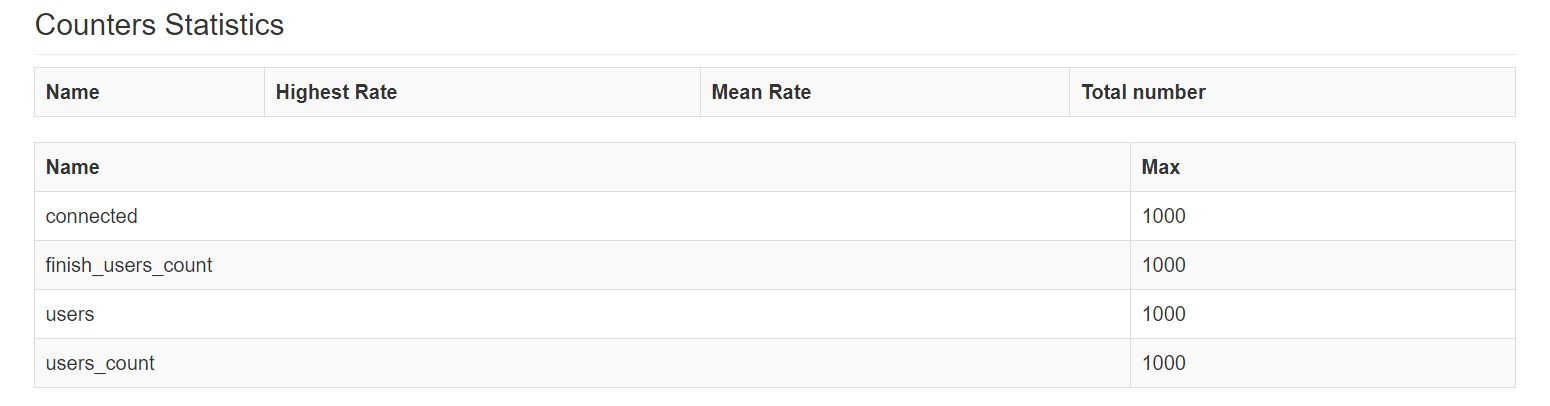
Highest Rate: 每秒最高响应请求数据量

Total: 响应请求总数据量

size\_sent: 表示发送请求数据量

Highest Rate: 每秒最高发送请求数据量

Total: 发送请求总数据量



Counters Statistics: 检查点统计数据

match: 匹配到的数据

Highest Rate: 最高的匹配数速率

Mean Rate: 平均匹配数速率

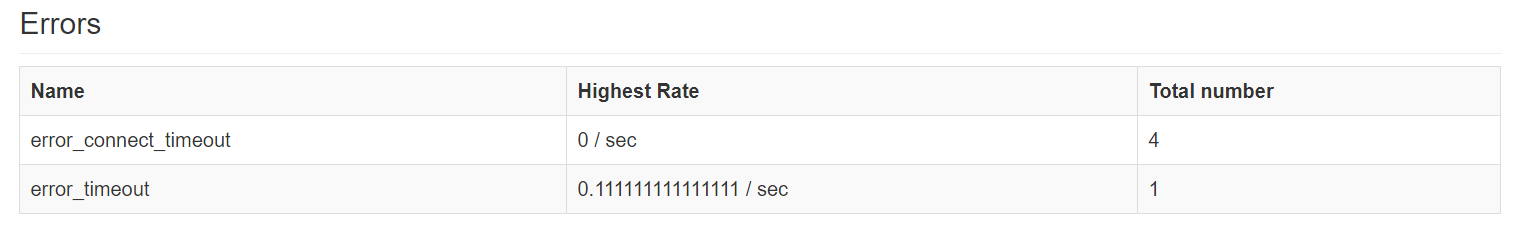
Total number: 总匹配数

connected: 表示会话开始且尚未结束，并且已建立连接的最大用户数（对于http，在思考时间内，不占用tcp连接，思考时间超时后会重新建立tcp连接）

finish\_users\_count: 已经完成了会话的最大用户数

users: 测试过程中，tsung建立起的最大并发用户数,包括还没有建立连接的用户数（会话已经开始，但是还没有结束）

users\_count: 表示Tsung总共生成的用户总数



Errors: 错误统计

Error\_abort\_max\_conn\_retries: 重新尝试连接错误

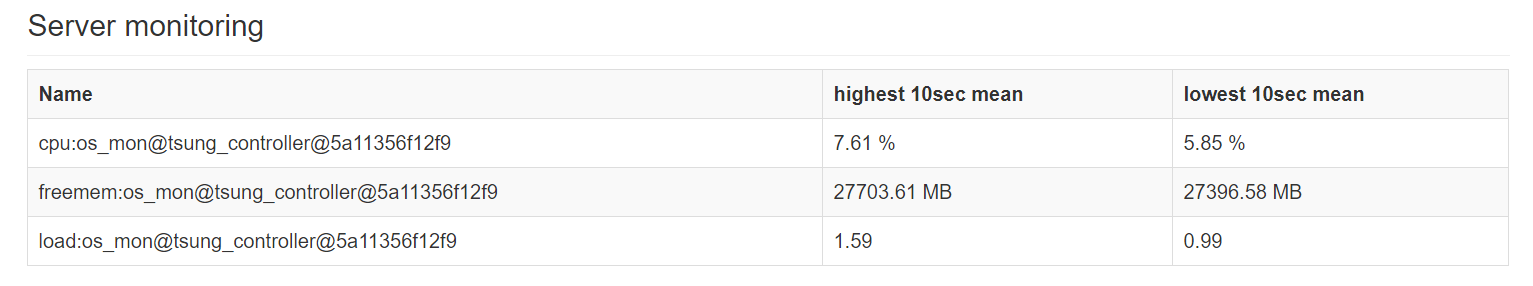
Error\_connect\_timeout: 连接超时错误

Error\_connect\_nxdomain: 不存在的域错误

Error\_unknown: 位置错误

Highest rate : 发生错误最高速率

Total number : 发生该错误总个数

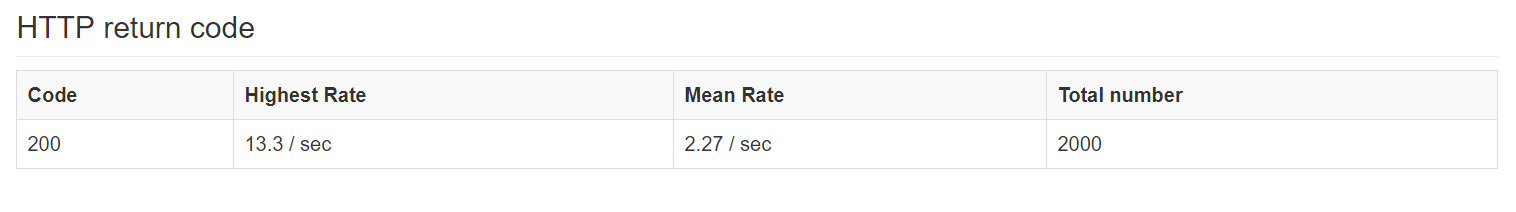


Server monitoring: 资源监控:使用的是erlang监控

cpu:os\_mon@localhost: 节点机器cpu使用率

freemem:os\_mon@localhost: 内存空闲资源

load:os\_mon@localhost: 负载。多核CPU的话，满负荷状态的数字为 "1.00 \* CPU核数"，即双核CPU为2.00，四核CPU为4.00。



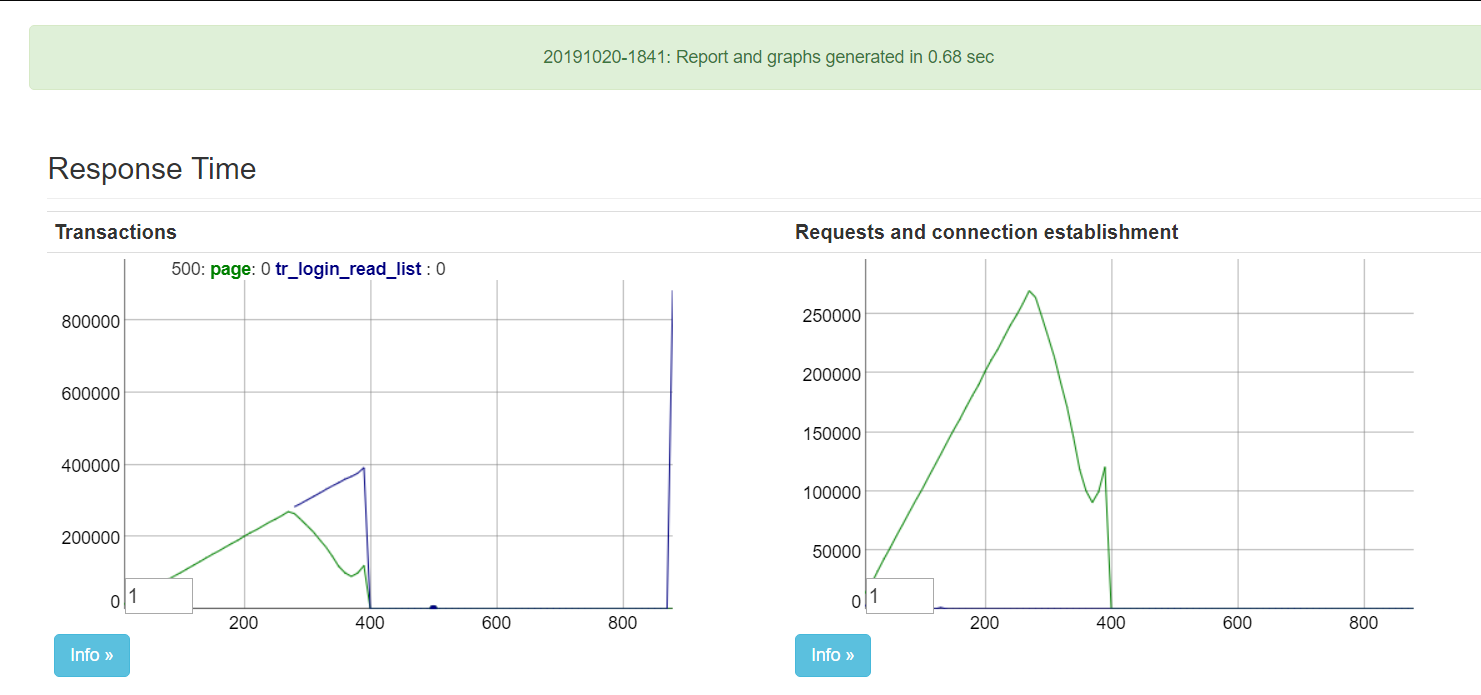
HTTP return code: http响应状态码统计

Code: 状态码

Highest Rate: 状态码返回最高速率

Mean Rate: 状态码返回平均速率

Total number: 返回状态码的总个数

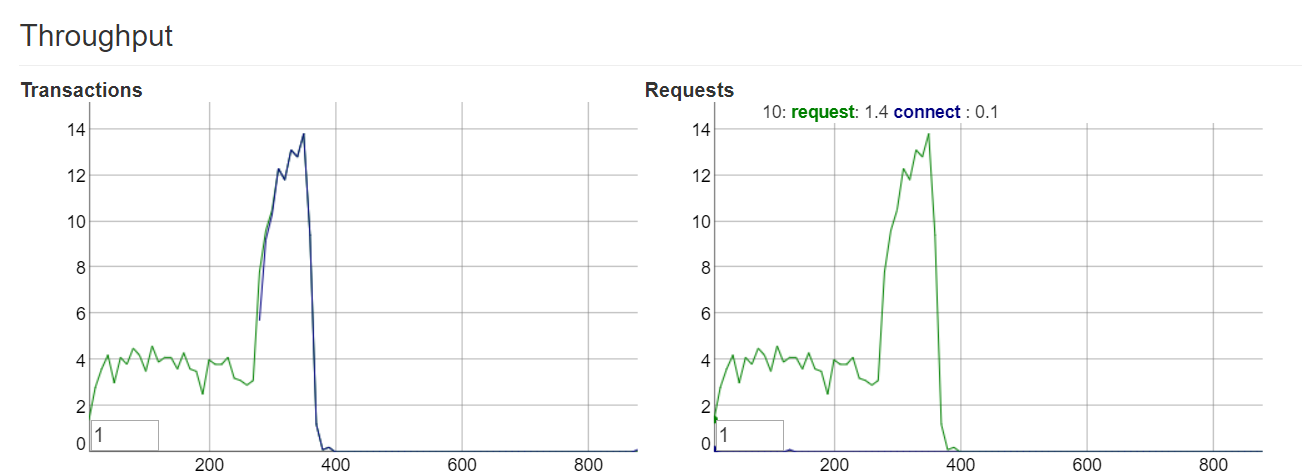


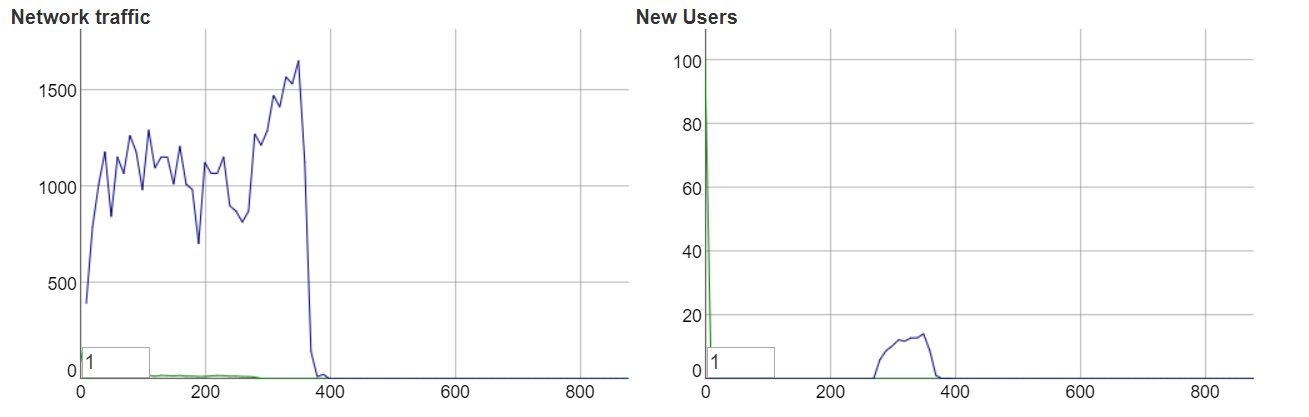
Response time: 响应时间

两图横坐标是时间，单位是秒，纵坐标也是时间，单位是毫秒

左图是页面和事务的响应时间随x轴时间的变化而变化

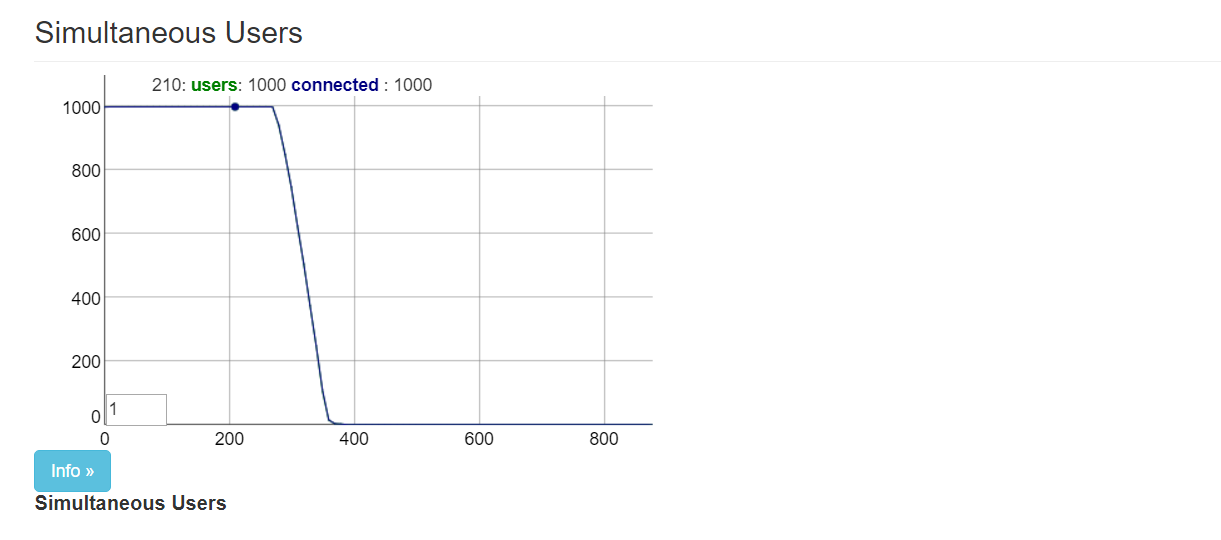
右图是连接和请求花费的时间随x轴时间的变化而变化





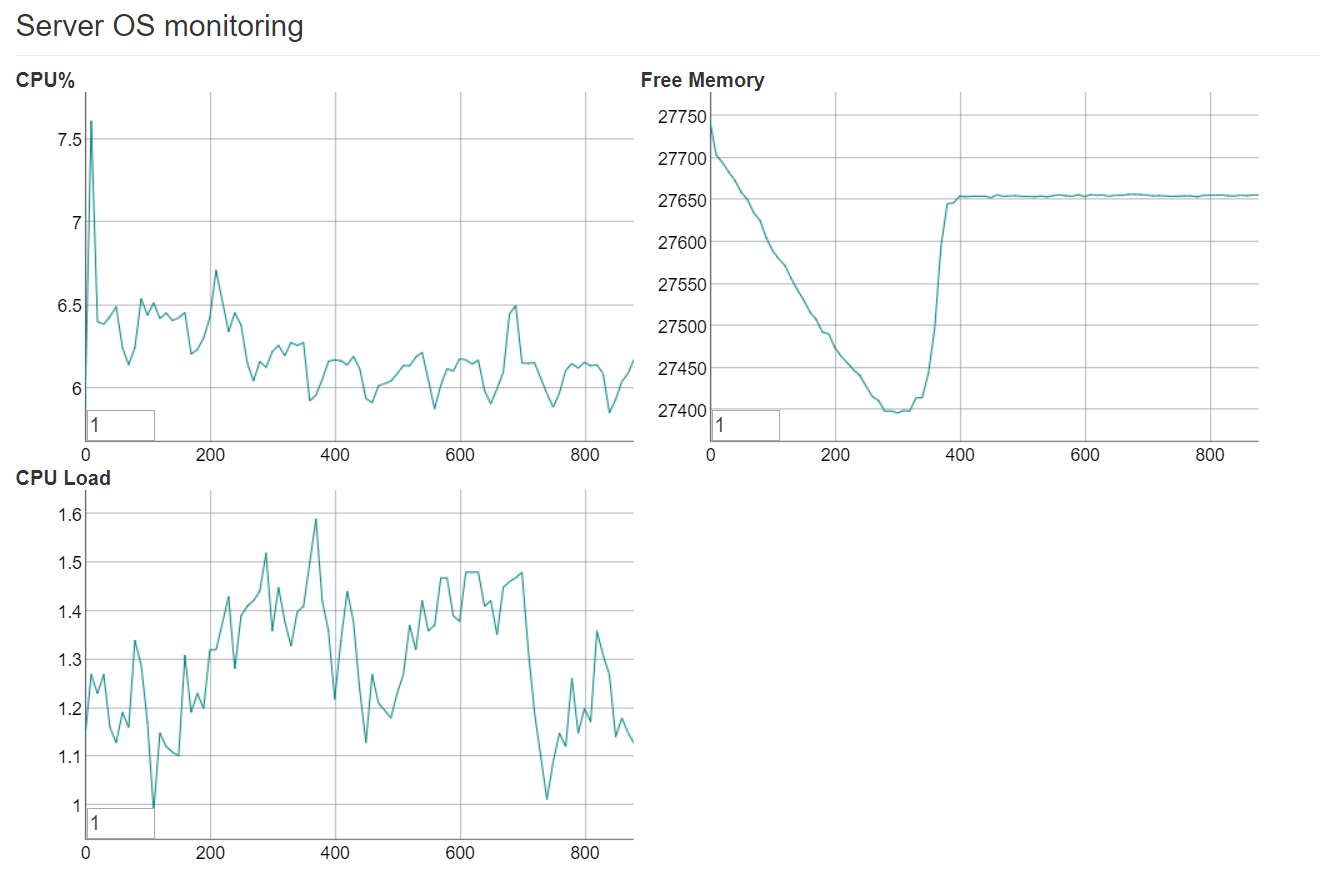
Throughput: 吞吐量

两图 横坐标是时间，单位是秒，纵坐标代表的都是速率



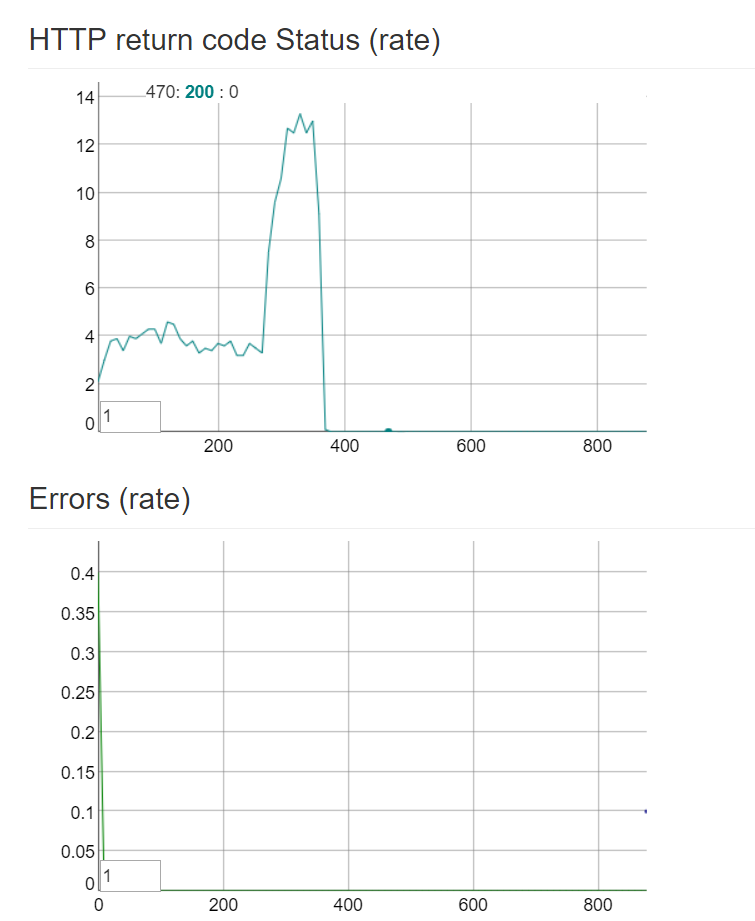
Simultaneous Users: 并发用户

横坐标是时间，单位是秒，纵坐标是数量，单位是个



Server OS monitoring: 系统监控

横坐标是时间，单位是秒，纵坐标是百分数



横坐标是时间，单位是秒，纵坐标是数量，单位是个/每秒

## 测试

以电力系统为例

1个cpu，2G内存，每秒产生100个用户，总共产生500个用户，模拟 登录，思考2秒，查询读表列表 这个流程，测试的结果在如下文件夹：20191027-2247

1个cpu，2G内存，每秒产生100个用户，总共产生1000个用户，模拟 登录，思考2秒，查询读表列表 这个流程，测试的结果在如下文件夹：20191027-2300

1个cpu，2G内存，每秒产生100个用户，总共产生500个用户，模拟 登录，创建市场活动 这个流程，测试结果在如下文件夹：20191027-2318

1个cpu，2G内存，每秒产生100个用户，总共产生1000个用户，模拟 登录，创建市场活动 这个流程，测试结果在如下文件夹：20191027-2329

2个cpu，4G内存，每秒产生100个用户，总共产生1000个用户，模拟 登录，创建市场活动 这个流程，测试结果在如下文件夹：20191027-2342

2个cpu，4G内存，每秒产生100个用户，总共产生500个用户，模拟 登录，创建市场活动 这个流程，测试结果在如下文件夹：20191027-2349

2个cpu，4G内存，每秒产生100个用户，总共产生1000个用户，模拟 登录，思考2秒，查询读表列表 这个流程，测试的结果在如下文件夹：20191027-2355

2个cpu，4G内存，每秒产生100个用户，总共产生500个用户，模拟 登录，思考2秒，查询读表列表 这个流程，测试的结果在如下文件夹：20191028-0008