

Software Testing Technique

Chapter 9

Performance Testing and Jmeter

Zan Wang (王赞)

Office: 55-A413

Email: wangzan@tju.edu.cn

Outline

- **Introduction to Performance Testing/ Load Testing**
- **Linux Performance Analysis**
- **Jmeter**

INTRODUCTION TO PERFORMANCE TESTING

Performance is a huge topic

- **How long will you wait for loading a web page ?**
 - 2s?
 - 5s?
 - 8s?
- **Response time:**
 - 2s : good
 - 5s : not too bad
 - 8s : limitation
- **Stability**
 - 2s in morning but 8s at night.

System Administrator

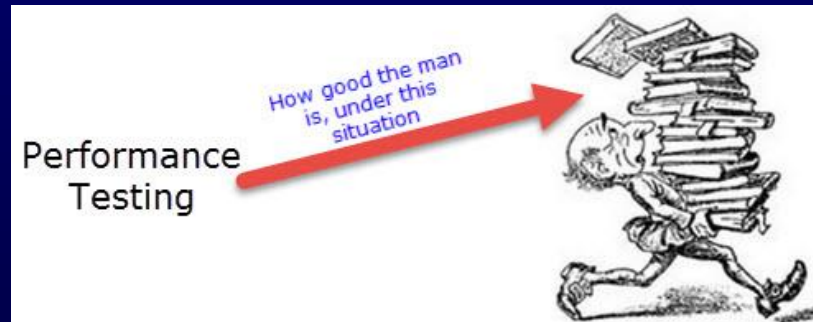
- Synchronous issues
- Important issues:
 - CPU
 - IO
 - RAM
 - Network

From Developer

- **Design of Software Architecture**
- **Design of Database**
- **Optimization of programs, such as SQL and Database tuning**

Performance Testing

- Performance testing is in general, a testing practice performed to determine how a system performs in terms of responsiveness and stability under a particular workload.
- It can also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.
- Performance test indicates response time for the entire application from the user's perspective.
- The goal of performance testing is not only find the bugs in the system but also eliminate the performance bottlenecks from the system.



Why do performance testing?

- If system goes live without doing performance testing may cause the issues like:
 - running system slow while simultaneously accessing system by several users
 - poor usability which likely to gain the bad reputation and it affects the expected sales goal directly.

Load Testing

- **Load Testing is a type of performance testing to check system with constantly increasing the load on the system until the time load reaches to its threshold value.**
- **It is also called as “Endurance testing” and “Volume testing”. The main purpose of load testing is to monitor the response time and staying power of application when system is performing well under heavy load.**
- **Load testing is performed to make sure that what amount of load can withstand the application under test. The successfully executed load testing is only if the specified test cases are executed without any error in the allocated time.**

Stress Testing

- **Stress Testing is a kind of performance testing to check the stability of software when hardware resources are not sufficient like CPU, memory, disk space etc.**
- **To determine or validate an application's behavior when it is pushed beyond normal or peak load conditions.**
- **Stress testing is Negative testing where we load the software with a large number of concurrent users/processes which cannot be handled by the system's hardware resources.**

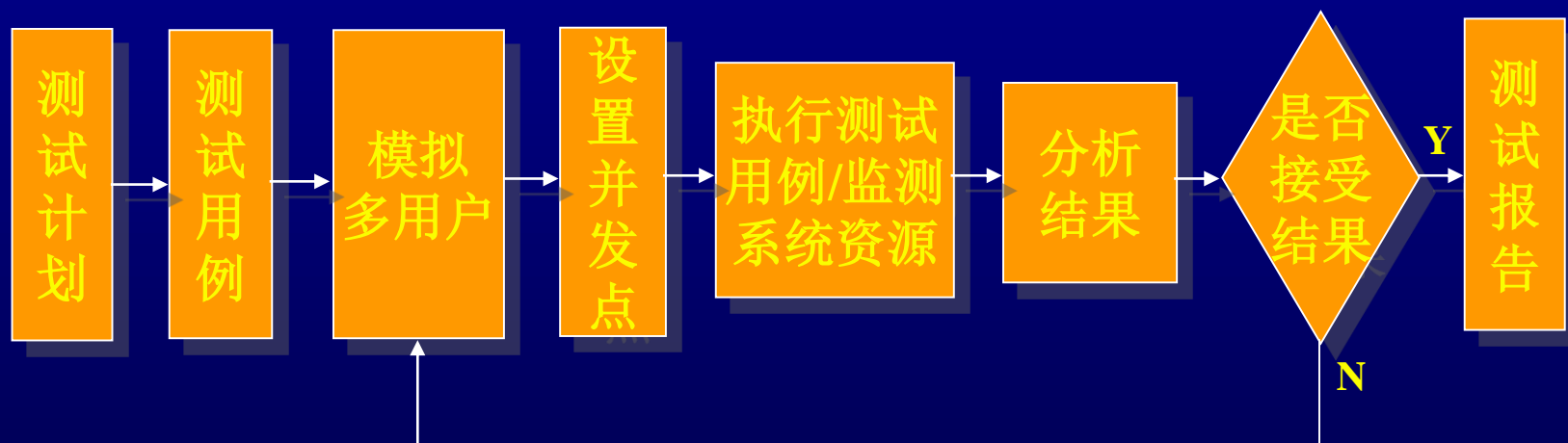
Some Performance Testing Tools

- **LoadRunner**
- **Apache Jmeter**
- **WebLOAD**
- **NeoLoad**
- **LoadUI**
- **OpenSTA**
- **WAPT**
- **LoadImpact**
- **Loadster**
- **Httpperf**
- **Rational Performance Tester**
- **QEngine (ManageEngine)**
- **Testing Anywhere**
- **CloudTest**
- **Loadstorm**
- ...

Load/stress testing

- **Load test** determines how is the performance of the application under the concurrent user sessions for a typical user scenario.
- **Stress test** examines how an application behaves under maximum load. In simple terms find the upper threshold for the application below which it can work normally.

压力测试流程图



压力测试计划

分析应用系统



定义压力测试对象与目标



选择压力测试工具



评审修改压力测试计划

Three step cyclic process of managing performance

- **Measure** - determine current performance levels
- **Estimate** - required/best case performance levels
- **Tune system** - to meet requirements/best case levels

User Load Estimation

- A detailed feed back from the marketing/business development will give an idea of the user load or the number of users using the product. This will determine the load to be used against the product in testing.
- Inexperienced staff may configure a load test to simulate 1000 v-user but user base for the application maybe not more than 400. This may result in licenses being lost for the v-user and time and effort as well.

User Load Distribution

- A detailed feed back from the marketing/business development will give an idea of the user load distribution.
 - The user load may be peak in the morning or afternoon and very less in the evening.
 - This may also help determine the concurrent users and simultaneous users.

User Load Distribution

- This factor will be input to the scenarios to be used and configured in the load test.
- BD may come up with a user load of 1000 users per day and each group to perform certain activities as say 30% in section A of the application rest in section B of the application etc.

User Activity Analysis

- A detailed discussion with marketing/BD may reveal the user activity details on which detailed scripts can be written.
- In the morning there may be 200 forms submitted or in the evening most of the users may login and perform results generation activity so in the morning scenario out of 1000 concurrent users, 70% may be performing this activity.

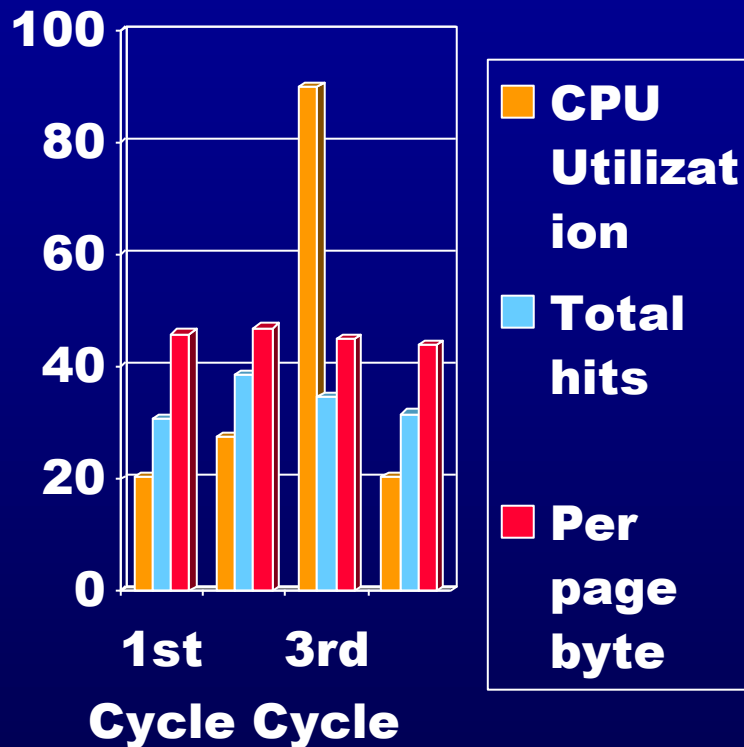
Report Generation

- **Generate two types of reports**
 - Detailed reports for engineering department
 - Generic reports for the activities performed in the scenario and response time details with other observations and conclusions for the management
- **Generating graphs also helps**
 - Helps in tracking per load cycle results
 - Easy interpretation for people

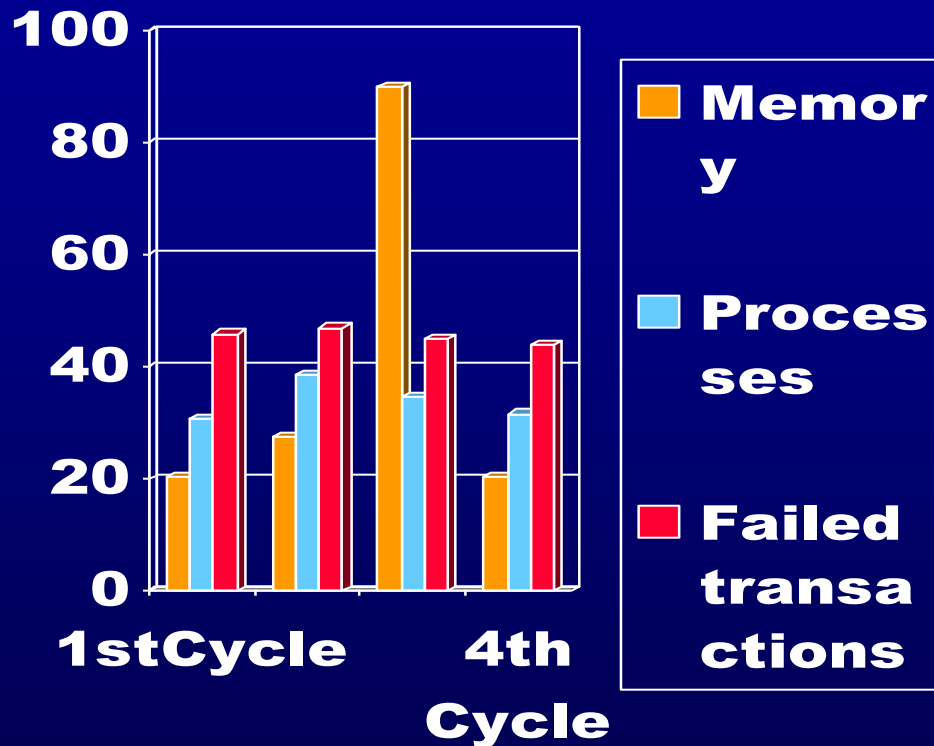
Simple Metric Collection

- Basic metrics can be collected as follows:

- CPU utilization : should not exceed 60%.
- Per page size. This is a indicator of the bulky-ness of the page. Response time for each transaction.
- Configuration details.
- Test bed details.



Simple Metric Collection



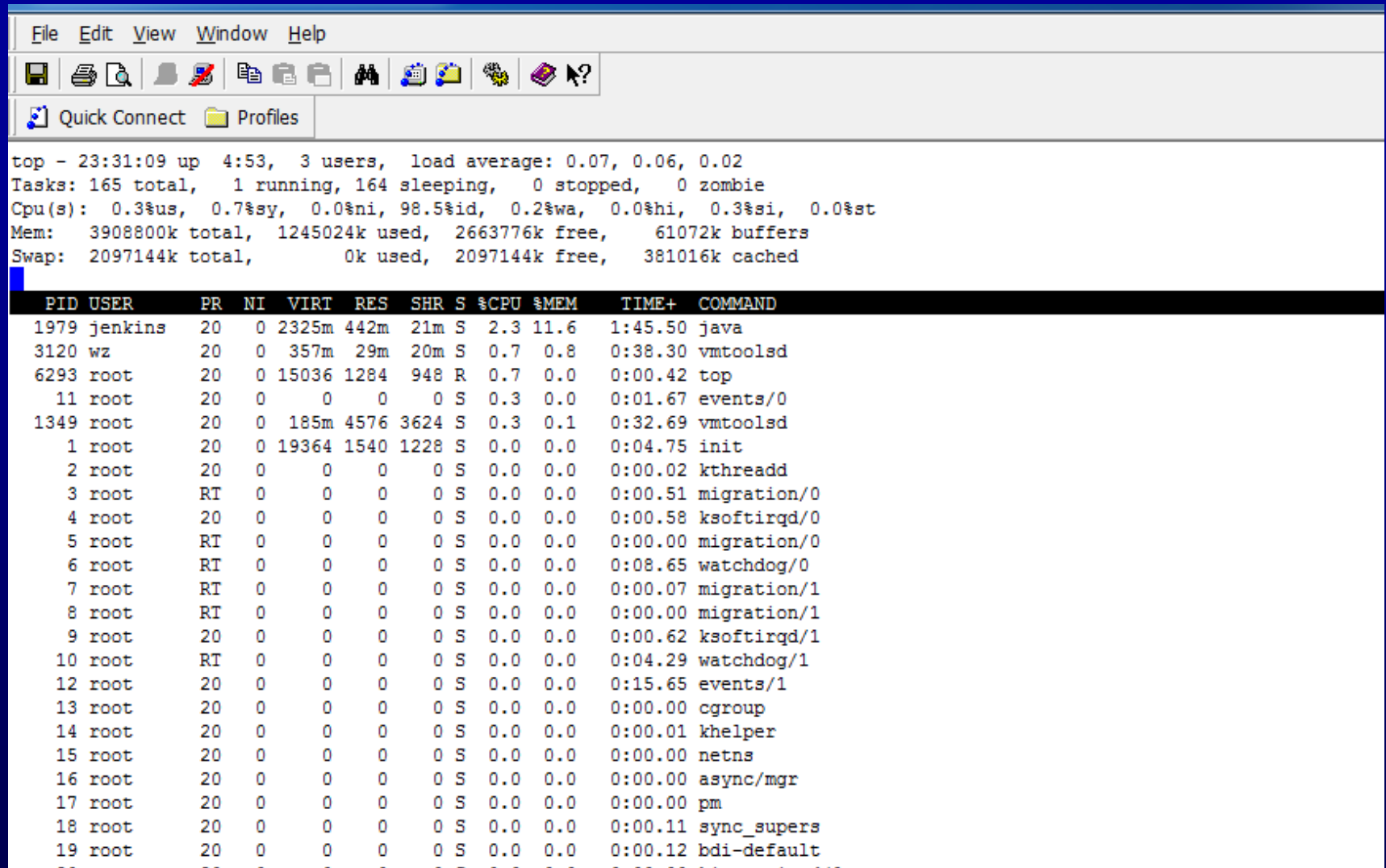
- Total hits and hits/second, should not be greater than 20 or request queue details need to be collected.
- % Failed transactions, should not be greater than 5%.
- Number of processes running on server.
- Memory details.

LINUX PERFORMANCE ANALYSIS

Key measures of a server

- CPU
- Memory
- IO
- Network

Linux tool -Top



```
top - 23:31:09 up 4:53, 3 users, load average: 0.07, 0.06, 0.02
Tasks: 165 total, 1 running, 164 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.3%us, 0.7%sy, 0.0%ni, 98.5%id, 0.2%wa, 0.0%hi, 0.3%si, 0.0%st
Mem: 3908800k total, 1245024k used, 2663776k free, 61072k buffers
Swap: 2097144k total, 0k used, 2097144k free, 381016k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1979	jenkins	20	0	2325m	442m	21m	S	2.3	11.6	1:45.50	java
3120	wz	20	0	357m	29m	20m	S	0.7	0.8	0:38.30	vmtoolsd
6293	root	20	0	15036	1284	948	R	0.7	0.0	0:00.42	top
11	root	20	0	0	0	0	S	0.3	0.0	0:01.67	events/0
1349	root	20	0	185m	4576	3624	S	0.3	0.1	0:32.69	vmtoolsd
1	root	20	0	19364	1540	1228	S	0.0	0.0	0:04.75	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.02	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:00.51	migration/0
4	root	20	0	0	0	0	S	0.0	0.0	0:00.58	ksoftirqd/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
6	root	RT	0	0	0	0	S	0.0	0.0	0:08.65	watchdog/0
7	root	RT	0	0	0	0	S	0.0	0.0	0:00.07	migration/1
8	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/1
9	root	20	0	0	0	0	S	0.0	0.0	0:00.62	ksoftirqd/1
10	root	RT	0	0	0	0	S	0.0	0.0	0:04.29	watchdog/1
12	root	20	0	0	0	0	S	0.0	0.0	0:15.65	events/1
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cgroup
14	root	20	0	0	0	0	S	0.0	0.0	0:00.01	khelper
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	netns
16	root	20	0	0	0	0	S	0.0	0.0	0:00.00	async/mgr
17	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pm
18	root	20	0	0	0	0	S	0.0	0.0	0:00.11	sync_supers
19	root	20	0	0	0	0	S	0.0	0.0	0:00.12	bdi-default

<http://www.cnblogs.com/peida/archive/2012/12/24/2831353.html>

CPU

- Cpu Usage Rate。
- load average
- Sysstat:
 - sysstat
 - sar -u
 - uptime
 - Top
 - %us, %sy

Memory

- Idle Memory
- Useful tools:
 - **vmstat**
 - **free**
 - **cat /proc/meminfo**
 - **Sysstat: sar -r**

IO

- **IO rate**
 - **iowait % < 20%**
- **Useful tools:**
 - **sar -d**
 - **iostat**

Network

- 网络质量，丢包、重传等问题
- **UDP:**
 - 查看所有监听的UDP端口的网络情况
 - **netstat -lunp**
 - 对于UDP服务，查看丢包情况
 - **netstat -su**
 - 如果packet receive errors数值较大，则表明在丢包
- **TCP:**
 - 更专注重传率
 - **cat /proc/net/snmp |grep Tcp**
 - 重传率 = **RetransSegs/OutSegs**

sysstat

- **Some tools:**

- **Cpu usage:** `sar -u`
- **Average load:** `sar -q`
- **Memory usage:** `sar -r`
- **查看页面交换发生状况:** `sar -W`

- **History report:**

- **查看CPU使用记录:** `sar -f /var/log/sa/sa**`
- **查看1/5/15分钟平均负载记录** `sar -q -f /var/log/sa/sa**`
- **查看指定文件7点到9点CPU使用记录, 如要看负载加参数-q:**
 - `sar -s 07:00:00 -e 10:00:00 -f /var/log/sa/sa**`
- **查看历史内存、swap使用情况**
 - `sar -r -f /var/log/sa/sa**`

JMETER

Introduction to Jmeter

- The Apache JMeter™ application is open source software, a 100% pure Java application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions.
- <http://jmeter.apache.org>

Jmeter Performance Testing Contains

- **Load Testing:** Using JMeter, Load Testing models the estimated usage by simulating multiple user get the web services simultaneously.
- **Stress Testing:** Stress testing is important to recognize the upper limits of capacity within the system. Usually we go through this testing to recognize the system's robustness in terms of excess load and assist application administrators to judge if the system will perform satisfactorily if the current load goes well above the ordinary maximum.

Steps of Load Test

- **Recording Tests**
- **Setting parameters**
 - Variables
 - CVS data set
 - BeanShell script
 - Regular expression
- **Assertion**
- **Performance Analysis**

Test Script Recorder

- 详见jmeter_proxy_step_by_step.pdf

参数修改

- **User Defined Variables**
 - Catid, gid
 - CSV file : ../data/catgd.csv

Assertion

- **BeanShell Assertion**

- <http://www.beanshell.org/>

```
java.util.regex.Pattern p = java.util.regex.Pattern.compile("id=(\\d+)");
java.util.regex.Matcher m = p.matcher(bsh.args[0]);
boolean found = m.find();
if (found) {
    if(m.group(1).equals(bsh.args[1])) {
        Failure = false;
        FailureMessage = m.group(1) + "<>" + bsh.args[1];
    }
    else Failure = true;
}
else failure=true;
```

结果分析

- Aggregate Report:

Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename

Browse...

Log/Display Only: ☐ Errors ☐ Successes

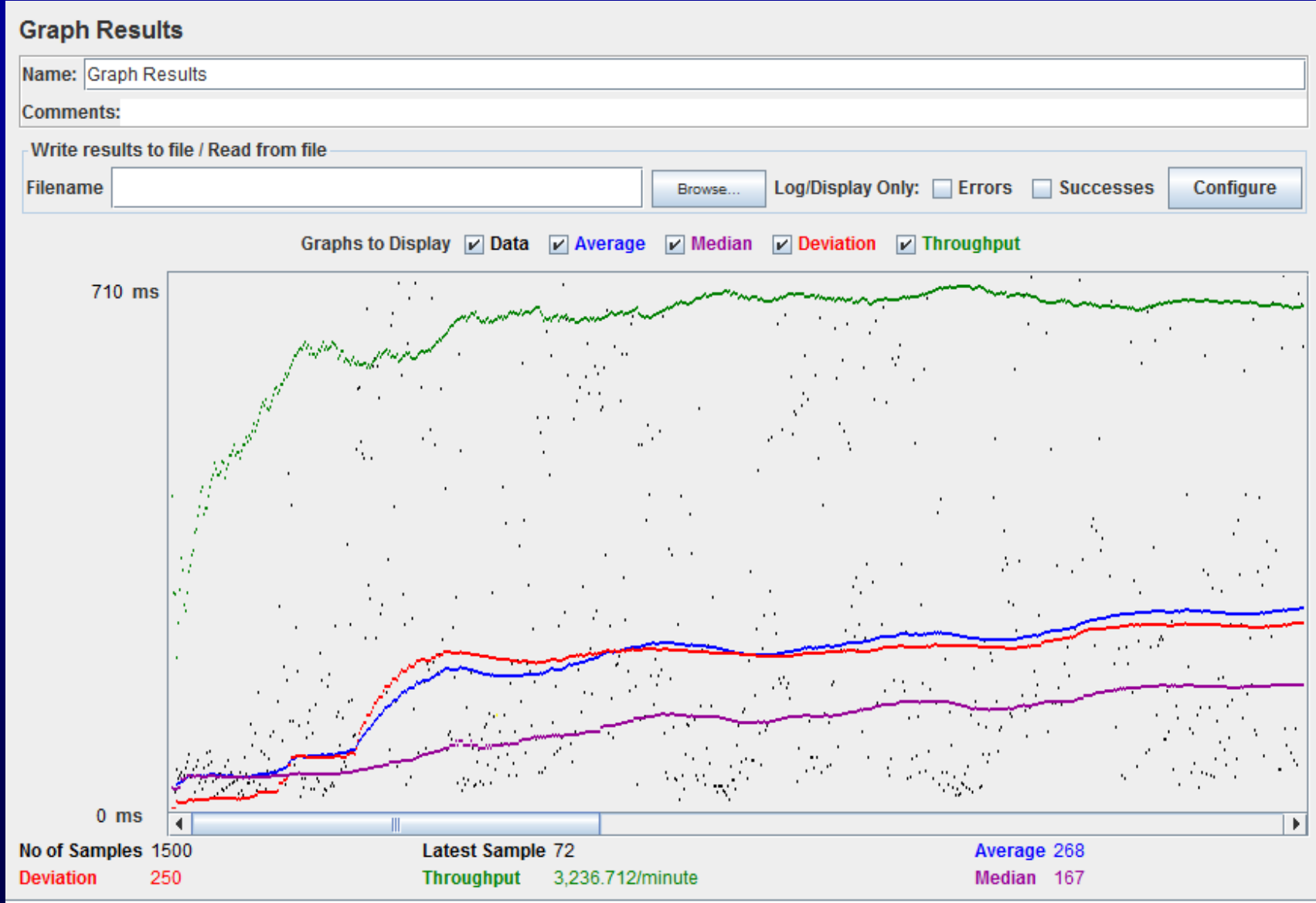
Configure

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Through...	KB/sec
114 /ecshop/	500	711	465	1448	2237	4338	22	4673	0.60%	11.8/sec	403.1
204 /ecshop/category.php	500	291	158	531	1328	1998	11	2862	0.00%	12.3/sec	268.9
208 /ecshop/goods.php	500	1337	1299	2069	2459	3586	48	4037	0.20%	12.7/sec	407.3
TOTAL	1500	780	490	1776	2112	4112	11	4673	0.27%	34.3/sec	1005.7

指标详细说明

- **Label:** 每个Jmeter的element都有一个Name属性，其实就是访问元素的名称。
- **#Samples:** 表示你这次测试中一共发出了多少个请求，如果模拟10个用户，每个用户迭代了10次，那么这里就是100.
- **Average:** 平均响应时间，默认情况下是单个Request的平均响应时间，单位是ms（毫秒）
- **Median:** 中位数，也就是 50% 用户的响应时间
- **90% Line:** 90%用户的响应时间
- **95% Line:** 95%用户的响应时间
- **99% Line:** 99%用户的响应时间
- **Min、Max:** 最小、最大响应时间
- **Error%:** 本次测试中出现错误的请求的数量/请求的总数
- **Throughput:** 吞吐量，每秒完成的请求数
- **KB/Sec:** 每秒从服务器端接收到的数据量。

Graph Result



- **Throughout:** 吞吐量, 这里是3236/minute, 也就是50tps
- **Average:** 平均响应时间, 268毫秒
- **Median:** 中位数
- **Deviation:** 偏差

监控性能

- 使用sysstat监控CPU、IO、Memory及Network情况。
 - `sar -u -f`
 - `sar -q -f`
 - `sar ...`