Performance Testing

MIEIC - TVVS
2019/2020

Nuno Oliveira
Rui Araújo

What is Performance Testing?

- Non-Functional Testing
- It’s a type of software testing to ensure software applications will perform well under their expected workload.

Goals:

- See if the software meets the performance requirements
- See whether there are any hardware or software factors that impact on the system's performance (bottlenecks)
- Provide valuable information to tune the system
- Predict the system's future performance levels

Performance Testing

The focus of Performance Testing is checking a software program's

- Speed - Determines whether the application responds quickly
- Scalability - Determines maximum user load the software application can handle.
- Stability - Determines if the application is stable under varying loads

Types of Performance Testing

- **Load testing -** Load testing focuses on the ability of a system to handle increasing levels of anticipated realistic loads resulting from transaction requests generated by controlled numbers of concurrent users or processes.
- **Stress testing -** involves testing an application under extreme workloads to see how it handles high traffic or data processing. The objective is to identify the breaking point of an application.
- **Endurance testing -** is done to make sure the software can handle the expected load over a long period of time.
- **Spike testing -** tests the software's reaction to sudden large spikes in the load generated by users.
- **Volume testing** - large number of Data is populated in a database and the overall software system's behavior is monitored. The objective is to check software application's performance under varying database volumes.
- **Scalability testing** - The objective of scalability testing is to determine the software application's effectiveness in "scaling up" to support an increase in user load. It helps plan capacity addition to your software system.

Common Performance Problems

- **Long Load time -** Load time is normally the initial time it takes an application to start. This should generally be kept to a minimum.
- **Poor response time -** Response time is the time it takes from when a user inputs data into the application until the application outputs a response to that input. Generally, this should be very quick. if a user has to wait too long, they lose interest.
- **Poor scalability -** A software product suffers from poor scalability when it cannot handle the expected number of users or when it does not accommodate a wide enough range of users.
- **Bottlenecking -** Bottlenecks are obstructions in a system which degrade overall system performance. Bottlenecking is when either coding errors or hardware issues cause a decrease of throughput under certain loads.

Testing metrics

- **Throughput**: how many units of information a system processes over a specified time;
- **Memory**: the working storage space available to a processor or workload;
- **Response time**: the amount of time that elapses between a user-entered request and the start of a system's response to that request;
- **Bandwidth**: the volume of data per second that can move between workloads, usually across a network;
- **CPU interrupts per second**: the number of hardware interrupts a process receives per second.
- **Concurrent users:** This the most common measure of load — how many active users at any point.
- **Peak response time:** This is the measurement of the longest amount of time it takes to fulfill a request. A peak response time that is significantly longer than average may indicate an anomaly that will create problems.

Difficulties in performance testing

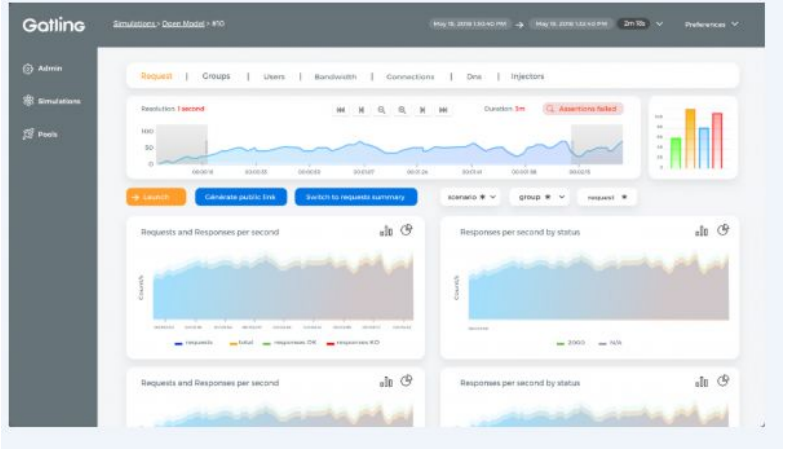
- The tool itself wastes resources
- Dependent on the network

Performance Testing Process

TOOLS

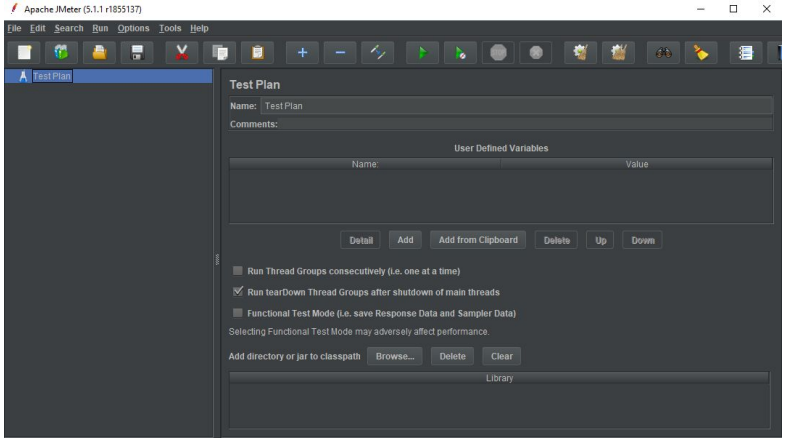
Gatling

- Open Source
- Any OS
- Recorder only GUI
- Language: scala
- Load Reports: html
- protocols:HTTP, JDBC,JMS
- No host monitoring



JMeter

- Open source
- Any OS
- Multiple load injectors
- Language: XML
- Test recorder: HTTP
- Language: Java, javascript, beanshell, jexl
- Load reports: XML, csv, tables, charts
- User-friendly GUI
- Simple charts for analyzing load statistics
- Protocols: HTTP, HTTPS, XML, SOAP, TCP (...)



LOCUST

- open source
- Any OS
- No test recorder
- Language: python
- Load reports in HTML
- Protocols: HTML
- no host monitoring

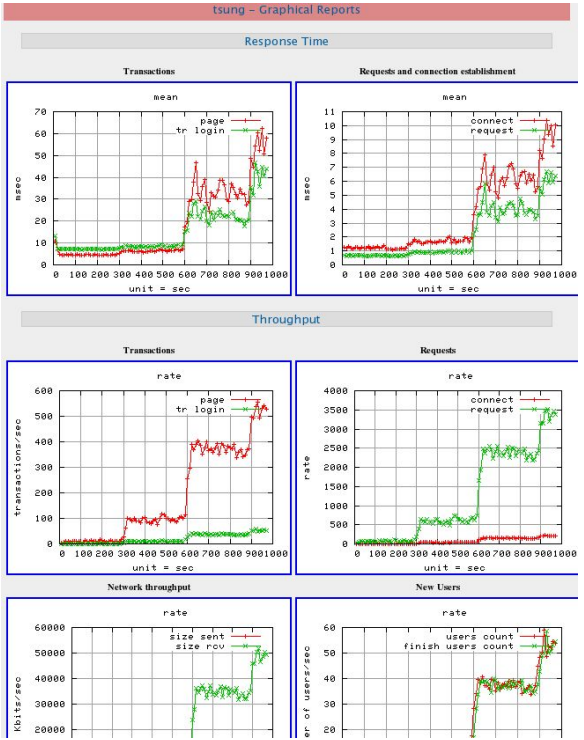
Type	Name	# Requests	# Fails	Median (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS
GET	/	5416	0	21	21	4	38	20336	44.1
GET	/blog	1745	0	27	26	3	49	20370	13.7
GET	/blog/[post-slug]	1824	0	15	15	2	27	19943	15.9
POST	/groups/create	185	0	57	55	5	108	3273	1.9
GET	/signin	10266	0	26	26	3	49	19949	66.6
POST	/signin	10266	0	82	82	45	120	20030	66.6
GET	/users/[username]	1802	0	31	31	6	55	20194	15

TSUNG

A distributed Load Testing Tool

TSUNG

- open source
- OS: linux
- Test recorder: HTTP, Postgres
- Language: XML
- Load reports: HTML
- Protocols: HTTP,Postgres,MySQL, WebDAV, XMPP…
- Host monitoring



WEBLOAD

- Native JavaScript and GUI
- Load tests on cloud and on-premise software
- Transactions recording
- Automatic bottleneck detection
- HTTP/HTTPS, WebSocket, PUSH, AJAX, SOAP, HTML5, WebDAV, etc

Load UI Pro

- Templates for stress and spike testing
- Load test creation without scripts
- HTTP, REST, SOAP, JSON, MQTT, etc
- Host monitoring
- GUI
- Language: XML
- Load reports: tables, charts
- Limitation: API only

Tool Comparison

	JMeter	Gatling	Tsung	LoadUI Pro
Open Source	yes	yes	yes	no
OS	All	All	Linux	All
HTTP protocol	yes	yes	yes	no
Websocket protocol	no	no	yes	no
Test Language	XML	Scala	XML	XML
Load reports	csv,xml,charts, tables	HTML	HTML	tables, charts
host monitoring	yes	no	yes	yes

QUESTIONS?