Performance testing with JMeter

Ilídio Oliveira | ico@ua.pt

TQS | 2020/04/28

universidade de aveiro
departamento de eletrónica, telecomunicações e informática
deti

Why performance testing?

Non function requirements

- Performance, latency
- How far can you load the system ensuring error-free behavior?

How?

- Synthetic load generation
    - simulate user actions (functional)
- Measurement & reporting instrumentations

Apache 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.

What can I do with it?

Apache JMeter may be used to test performance both on static and dynamic resources (Webservices (SOAP/REST), Web dynamic languages - PHP, Java, ASP.NET, Files, etc. -, Java Objects, Data Bases and Queries, FTP Servers and more). It can be used to simulate a heavy load on a server, group of servers, network or object to test its strength or to analyze overall performance under different load types. You can use it to make a graphical analysis of performance or to test your server/script/object behavior under heavy concurrent load.

What does it do?

Apache JMeter features include:

- Ability to load and performance test many different server/protocol types:
  - Web - HTTP, HTTPS
  - SOAP / REST
  - FTP
  - Database via JDBC
  - LDAP
  - Message-oriented middleware (MOM) via JMS
  - Mail - SMTP(S), POP3(S) and IMAP(S)
  - Native commands or shell scripts
  - TCP
- Complete portability and **100% Java purity**.
- Full **multithreading** framework allows concurrent sampling by many threads and simultaneous sampling

Browser vs JMeter as HTTP clients

Browser lifecycle

User performs an action

Browser sends an HTTP request

Server processes the request and responses

Browser parses the response and executes scripts

JMeter behaviour

~~User performs an action~~

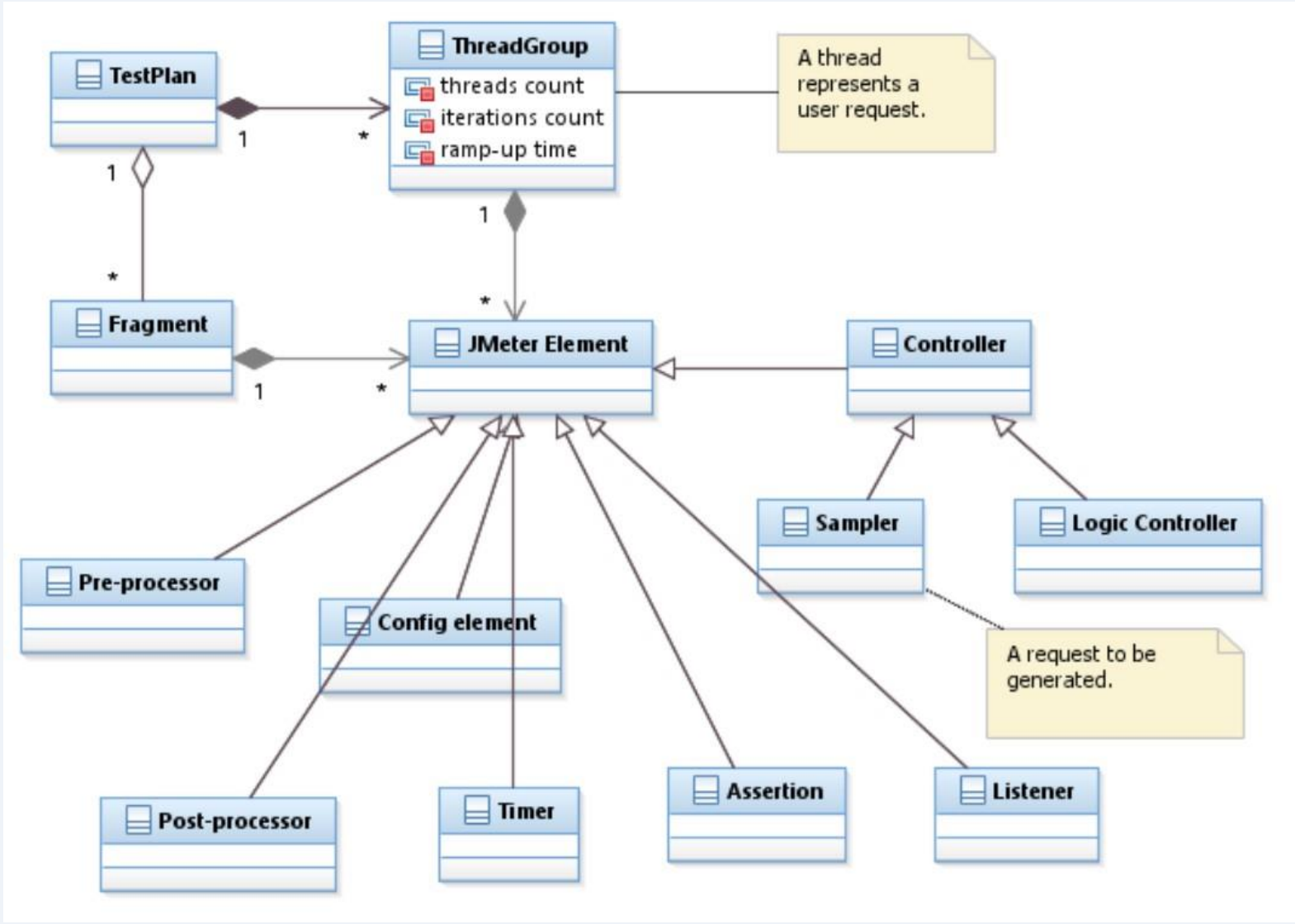
JMeter sends an HTTP request

Server processes the request and responses

JMeter parses the response ~~and executes scripts~~

Repeat

JMeter elements



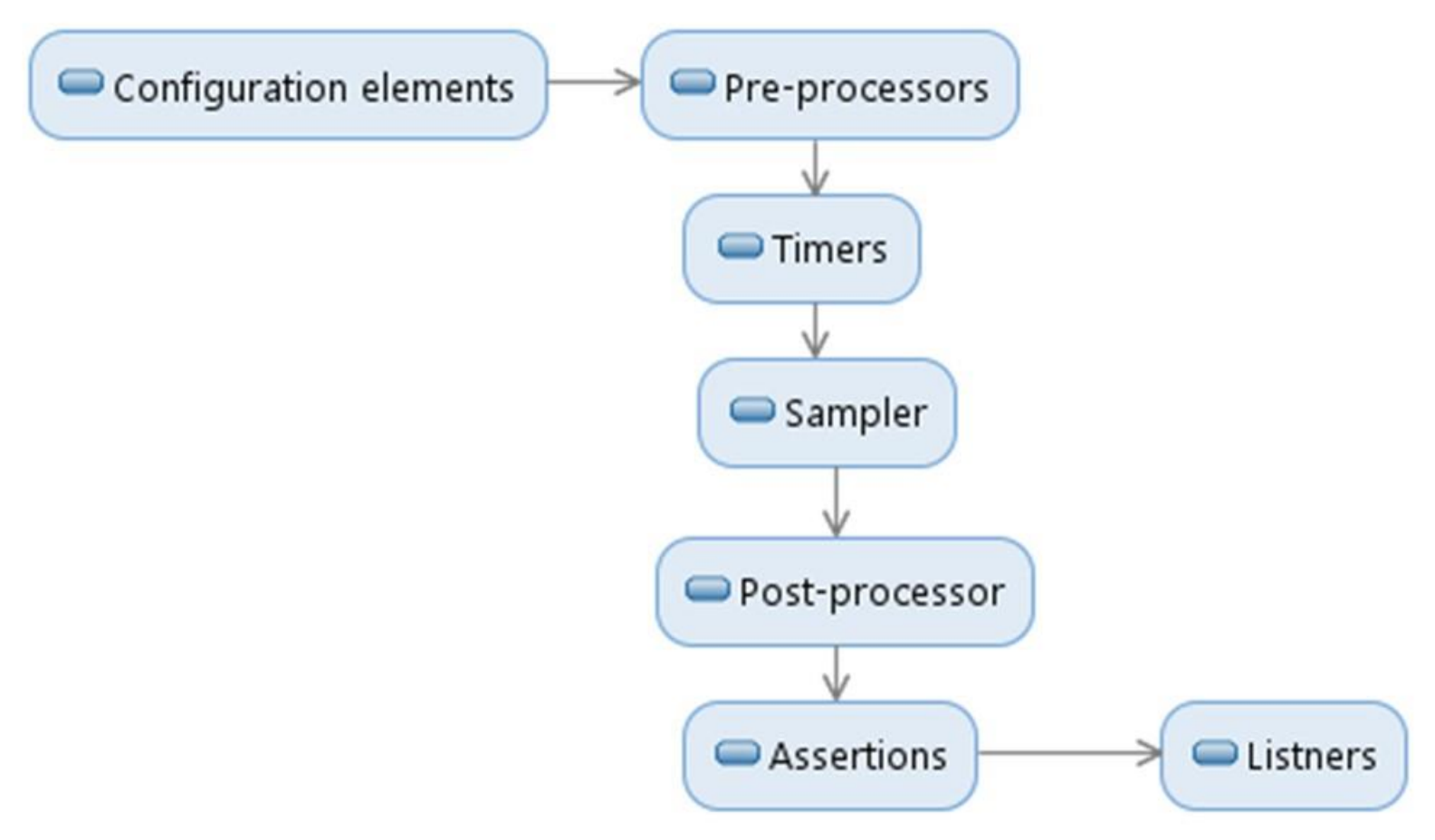
JMeter elements

Element	Semantics
Test Plan	A JMeter script.
Thread Group	Simulates a group of users (request ~ user)
Sampler	An action that causes a request.
Config	Additional configuration.
Timer	Add a predefined delay.
Assertions	Error checking to evaluate responses
Pre-processor	Modify the request before it is issued
Post-processor	Modify the response
Logic controller	Control node (alternatives, looping,…)

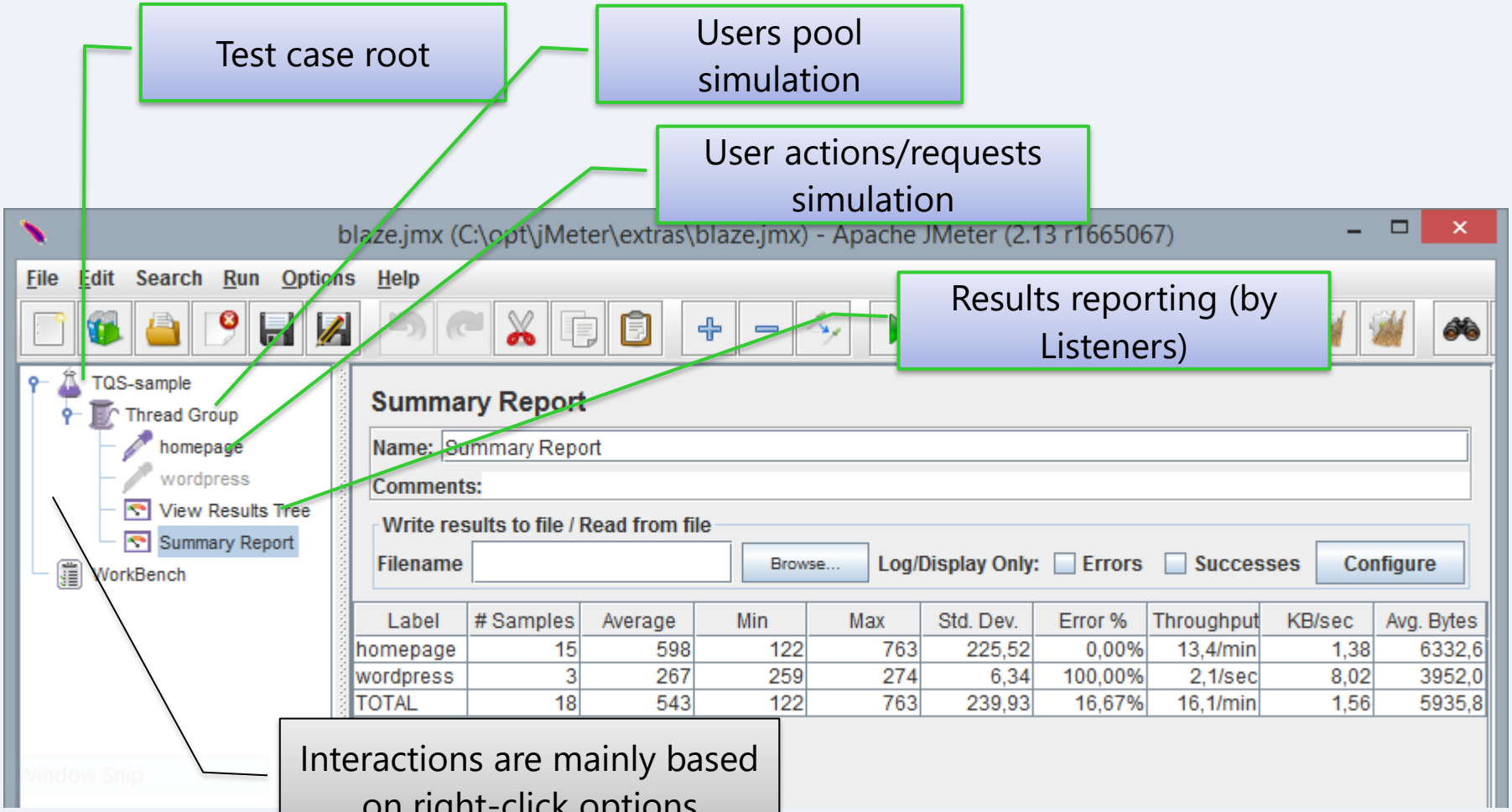
Some useful Samplers

- HTTP Request
- FTP Request
- JDBC Request
- Java Request
- SOAP/XML Request
- RPC Requests

Execution order of test elements



JMeter dashboard



Basic http request

New test plan, with descriptive name

Simulate

- 4 users
- 1 visit
- to UA's home page (http request)

Display the results

- in Results Tree view
- in Summary report

http://www.tutorialspoint.com/jmeter/jmeter_web_test_plan.htm

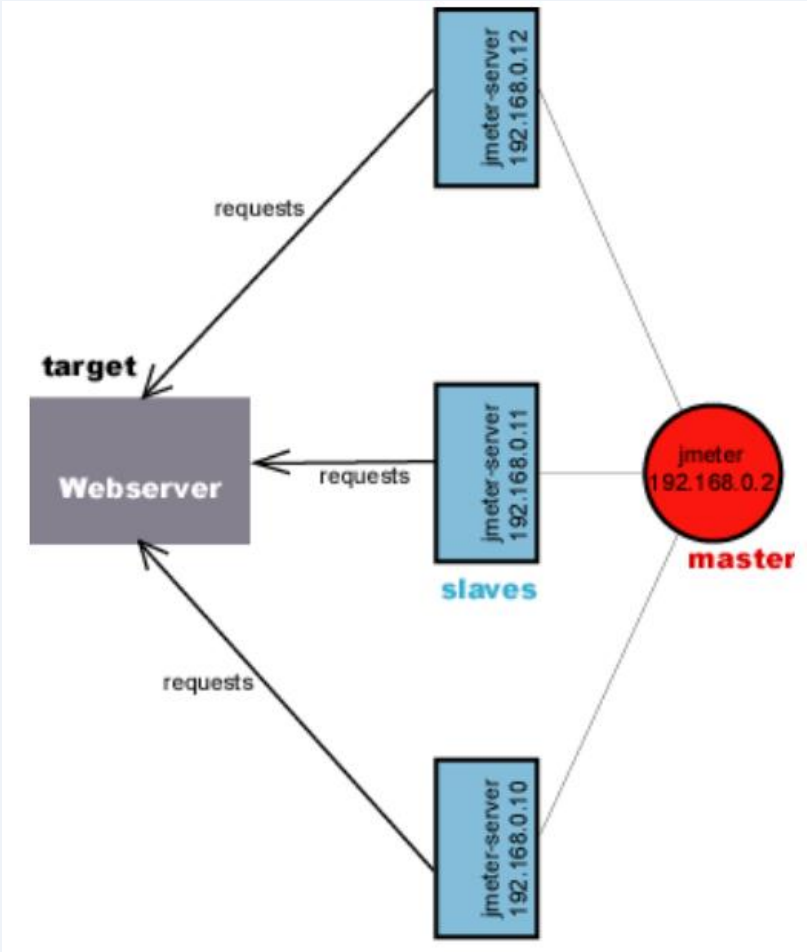
Test a REST endpoint

- Add http request
- Provide details for GET
- Change the http-header to send json

http://www.testautomationguru.com/how-to-test-rest-api-using-jmeter/

Distributed performance testing

Master/slave architecture



JMeter practices

Use multiple instances of JMeter if using many threads

JMeter can be run without GUI

- jmeter -n -t test.jmx -l test.jtl

Use as few Listeners as possible

- deactivate/activate

Prefer CSV over XML to save results

If the machine running the test case is resource-exhausted (e.g.: CPU 100%), the results will not be reliable.

Remember GUI mode is for Script creation and debugging, not for load testing

Resources and references

Apache JMeter site

- includes demo tutorials

The JMeter Manual

JMeter and WebDriver integration

Chrome plug-in to save and export as JMeter scripts