GFT

FOUNDATIONS OF SOFTWARE TESTING

Test automation with Jmeter

Piotr Sobieraj
02/09.12.2016

GFT

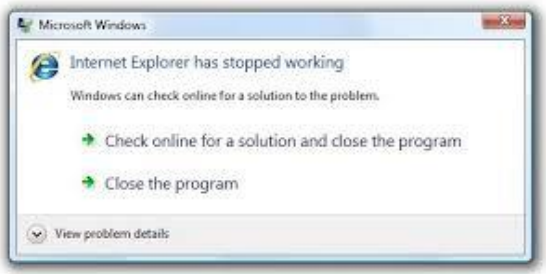
AGENDA

1. Introduction to performance tests
2. What is Jmeter ?
3. What we can do with Jmeter?
4. Excercises
5. Summary

Non-functional testing

- The testing of software attributes which are not related to any specific function or user action like performance, scalability, security or behavior of application under certain constraints.
- Non functional testing has a great influence on customer and user satisfaction with the product.
  - Why? Any examples ?

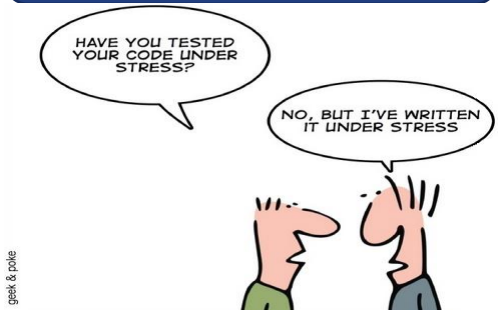
Nobody's madder than me about the website not working as well as it should, which means it's going to get fixed.



Types of performance testing

Validate that the system performs as expected when concurrent users access the application and gets the expected response time.

Stress testing is the test to find the ways to break the system. The test also gives the idea for the maximum load the system can hold.



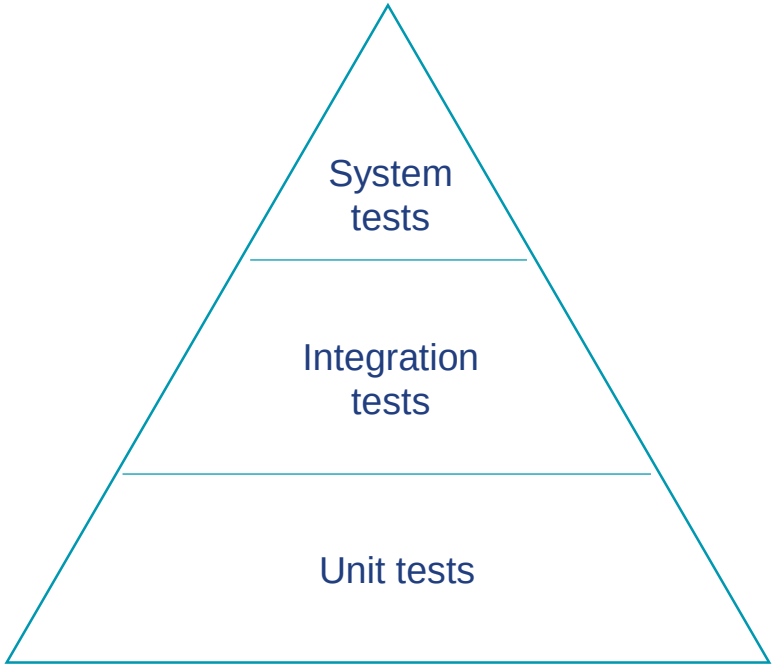
Dramatic increase of the load and then decrease. Should verify that an application recovers between periods of spike activity.

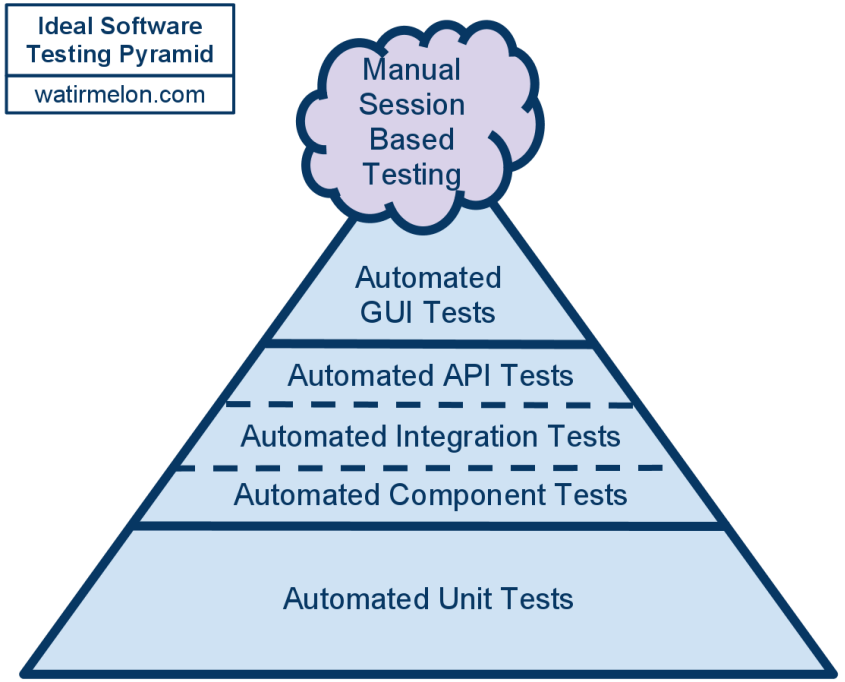
A long-running test that is used to determine application performance and/or stability over time. Often used to find memory leaks.

Volume test is to verify if the performance of the application is not affected by volume of data that is being handled by the application

Quite similar to load testing but focused more on checking what can be improved on hardware site to keep the good performance.

Testing pyramid - automation





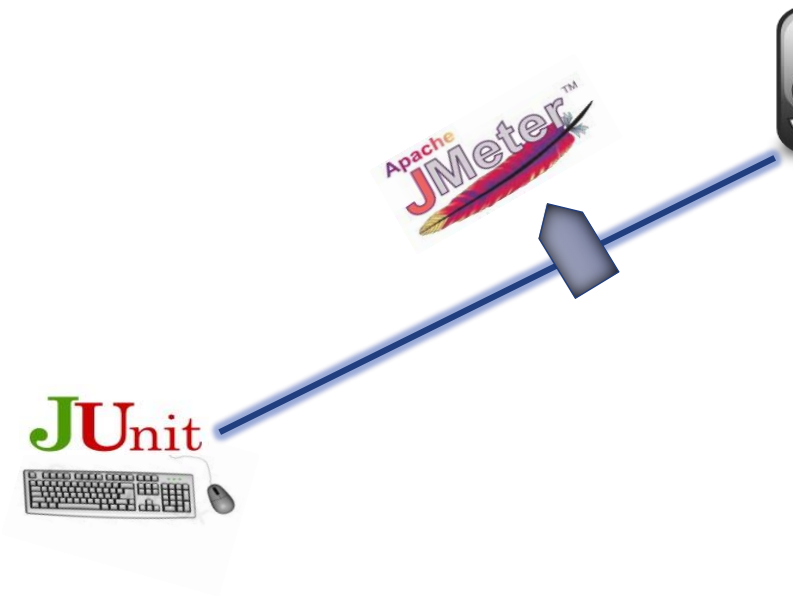
Automating performance tests

- Measure what is measurable and what is not measurable – make measurable - Galileo Galilei
- Why automation is the best choise for that types of tests:
  - Hard to perform manually
  - Hard to measure manually
  - Hard to repeat manually
- Performance testing tools can:
  - Simulate real or stress performance scenarios
  - Monitor and report the metrics and fails

GFT

Why JMeter ?

GFT



Apache JMeter

Open source license	Friendly GUI	Platform independent
Full multi-threading framework	Visualize Test Result	Easy installation
Highly extensible	Unlimited testing capabilities	Support multi protocol

Task 1 – getting familiar with the tool

GFT

1. Ensure that you have JDK installed
2. Ensure that JAVA_HOME is properly set
3. Download Jmeter 3.1
4. Unzip and run the executable ApacheJMeter.jar file
5. Add Thread Group to the Test Plan
6. Get familiar with the structure and elements which can be added to the Thread Group

- http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- set JAVA_HOME = C:\Program Files\Java\jdk1.8.0_xx
- echo %JAVA_HOME%
- http://jmeter.apache.org/download_jmeter.cgi

Task 1 – getting familiar with the tool

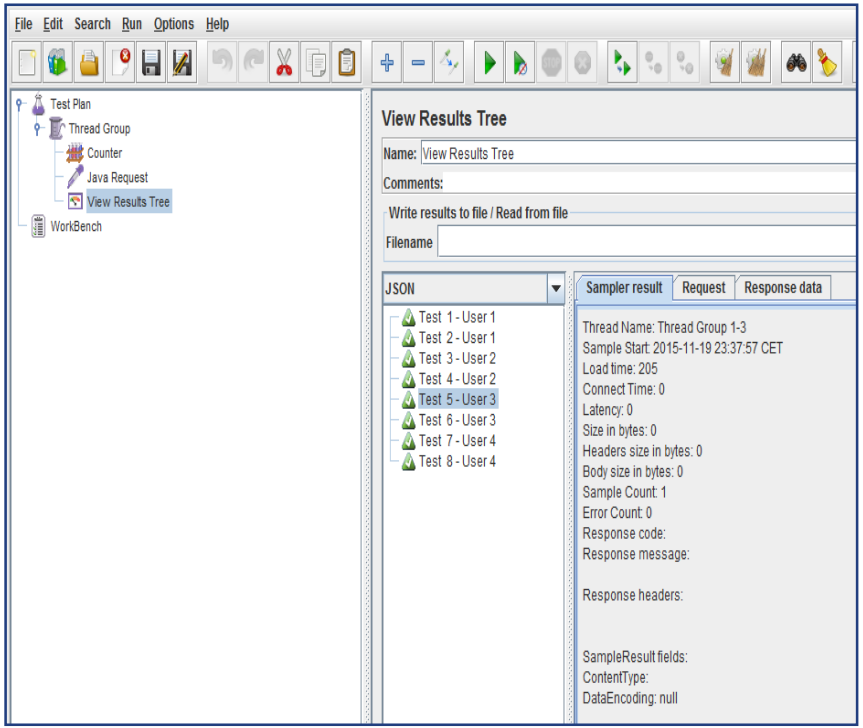
GFT

Element	Functionality
Thread groups	All controllers and samplers must be under a thread group. Other elements, may be placed directly under the test plan, in which case they will apply to all the thread groups. This element controls the number of threads JMeter will use to execute your test.
Logical Controllers	Let you customize the logic that JMeter uses to decide when to send requests.
Samplers	Tell JMeter to send requests to a server.
Listeners	Are ways to access or show the information about run results. For example there are Result Graphs or View Results Trees.
Timers	Allows to configure a delay between requests, in order not to overwhelm the server.
Assertions	Statements of a fact that represents the rule of what must be returned by the application by a certain point.
Configuration	They work with samples and are able to add or modify requests.

Task 2 – Create simple JavaRequest

GFT

1. Add JavaRequest from Samplers and ResultsTree from Listeners (right click on Thread Group)
2. Change Label parameter in JavaRequest and run test plan (Ctrl+R).
3. Check results in Results Tree
4. Increase Threads Parameters and run test plan again (useful shortcuts ->Ctrl+S, Ctrl+E, Ctrl+R)
5. Change label to „Test User”
6. Generate function string for __threadNum from FunctionHelperDialog and add this string at the end of Label
7. Run Test Plan and check results
8. Add Counter element and use it also in JavaRequest Label (use following method ${xx} where xx is Counter Reference name)
9. Run Test Plan and check results



Task 3a – Prepare application

1. Download RCApp zip from GITRepository
2. Unzip the archive and go to tomcat bin directory (.. rcapp_external_db\rcapp\apache-tomcat-6.0.44\bin)
3. Run command line from that path and start tomcat by executing „**startup**” command
4. Once Tomcat server will be started go following url: http://localhost:8080/registrationform/?contestId=1
5. Check if you see the the screen similar to presented on the left
6. Check basic functionality of the contest (register your answer for whichever question)

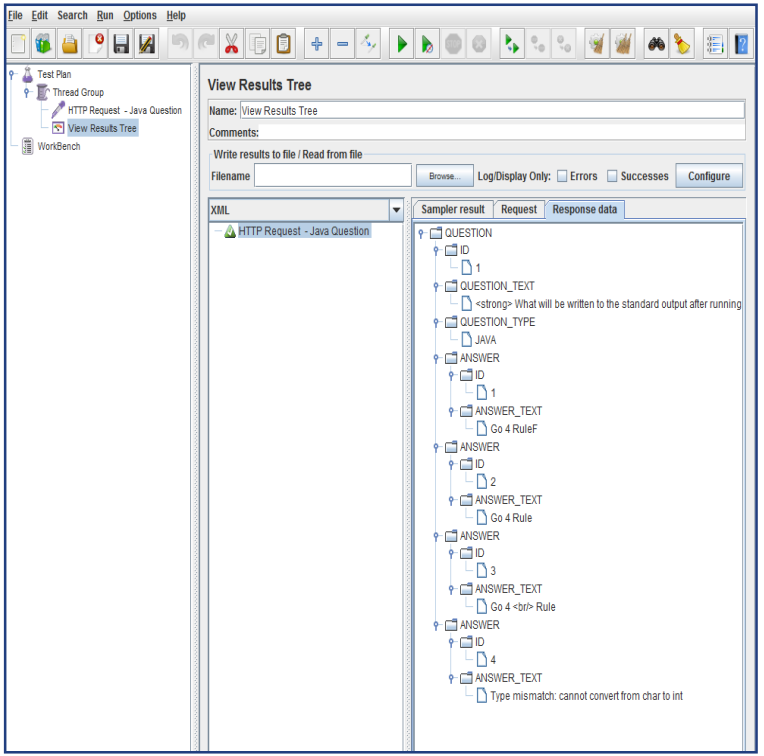
Task 3b – Using HTTP Request retrieve a JAVA question

GFT

1. Create an empty Test Plan and change its name to Test Plan – Task 3b.
2. Save the test plan by the name of Task_3b.jmx.
3. Add Thread Group.
4. Add HTTP Request (Add > Sampler > HTTP Request) to the Thread Group and change its name to HTTP Request – JAVA question. Change the relevant settings as below:

   a. Server Name or IP: 127.0.0.1
   b. Port Number: 8080
   c. Protocol [http]: HTTP
   d. Method: GET
   e. Content encoding: UTF-8
   f. Path: /registrationform/registrations/JAVA

5. Add View Result Tree to the Test Plan to observe flow of the test.
6. Run test plan (please note that API will repond with randomly chosen question)
7. Click on View Result Tree to see Response Data (change response format to XML)
8. Add two more similar HTTP Requests for SQL and DOTNET (only last parameter in PATH should be different)



Task 3c – Using HTTP Request [POST] register new user

GFT

1. Create an empty Test Plan and change its name to Test Plan – Task 3c
2. Save the test plan by the name of Task_3c.jmx.
3. Add Thread Group.
4. Add HTTP Request and change its name to HTTP Request – register user. Change the relevant settings as below:
5.

   a. Server Name or IP: 127.0.0.1
   b. Port Number: 8080
   c. Protocol [http]: HTTP
   d. Method: **POST**
   e. Content encoding: UTF-8
   f. Path: /registrationform/registrations/

6. Switch onto Body Data tab and put xml content there
7. Run test plan – response code 415
8. Add HTTP Header Manager and parameter Content-Type: text\xml
9. Run Test plan and check if user was registered
10. Run test plan again – what happened?
11. Add Assertion for the Response code

```
<?xml version="1.0" encoding="UTF-8"?>
<REGISTRATION>
<FIRST_NAME>name</FIRST_NAME>
<LAST_NAME>name2</LAST_NAME>
<EMAIL>mailj@gft.com</EMAIL>
<QUESTION>Which one of the following keywords
allows to sort the query
result?
GROUP BY
ORDER FROM
ASC BY
ORDER BY</QUESTION>
<CHALLENGE>null</CHALLENGE>
<ANSWER>null</ANSWER>
<QUESTION_ID>7</QUESTION_ID>
<ANSWER_INDEX>28</ANSWER_INDEX>
</REGISTRATION>
```

http://localhost:8080/registrationpanel/#RegistrationsPlace:Registrations
Login: admin
Passw:

Other JMeter posibilities

GFT

- DB connections and testing
- Using external files to upload the data
- Veryfing HTM and web services responses
- Parsing documents
- Distributed performance tests
- Recording tests
- And many many more...

Links and materials

GFT

- http://osworld.pl/testy-wydajnosciowe-za-pomoca-apache-jmeter/
- https://www.digitalocean.com/community/tutorials/how-to-use-jmeter-to-record-test-scenarios
- https://www.urbaninsight.com/2011/07/18/simple-load-test-with-jmeter
- https://lincolnloop.com/blog/load-testing-jmeter-part-1-getting-started/
- http://blog.sourcepole.com/2011/01/04/jmeter-series/

GFT

Thank you

Piotr Sobieraj
GFT
Piotr.Sobieraj@gft.com