Performance testing and reporting with JMeter

Introduced by: Nghi Nguyen Van
~ March, 1st 2012 ~

Agenda

1. *Introduction*
2. *What is performance testing, why is it important*
3. *Performance testing and monitoring tools*
4. *Performance testing with JMeter*
5. *What to focus in a performance test report*
6. *Tips and tricks*
7. *Q & A*

Introduction

Purpose of this presentation:

Introduce about performance testing activity in eXo SEA

Share experience in doing performance testing

Share experience in working with JMeter

This presentation serves to:

- ***Leaders to understand about performance testing activity, reports***
- ***Who is new in using JMeter to gain knowledge about using JMeter***
- ***Who experienced in JMeter want to gain, share & discuss about using JMeter***
- ***Testers who want to understand about building, using performance testing system***

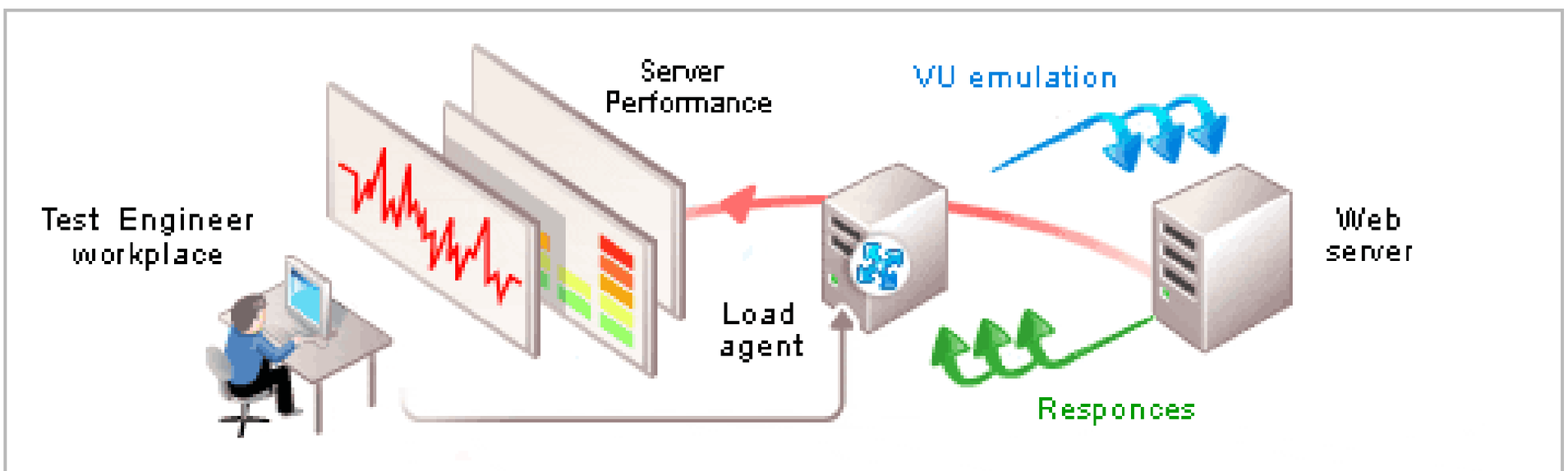
What is performance testing, why is it important?

What is performance testing

Definition:

Performance testing is a type of testing intended to determine

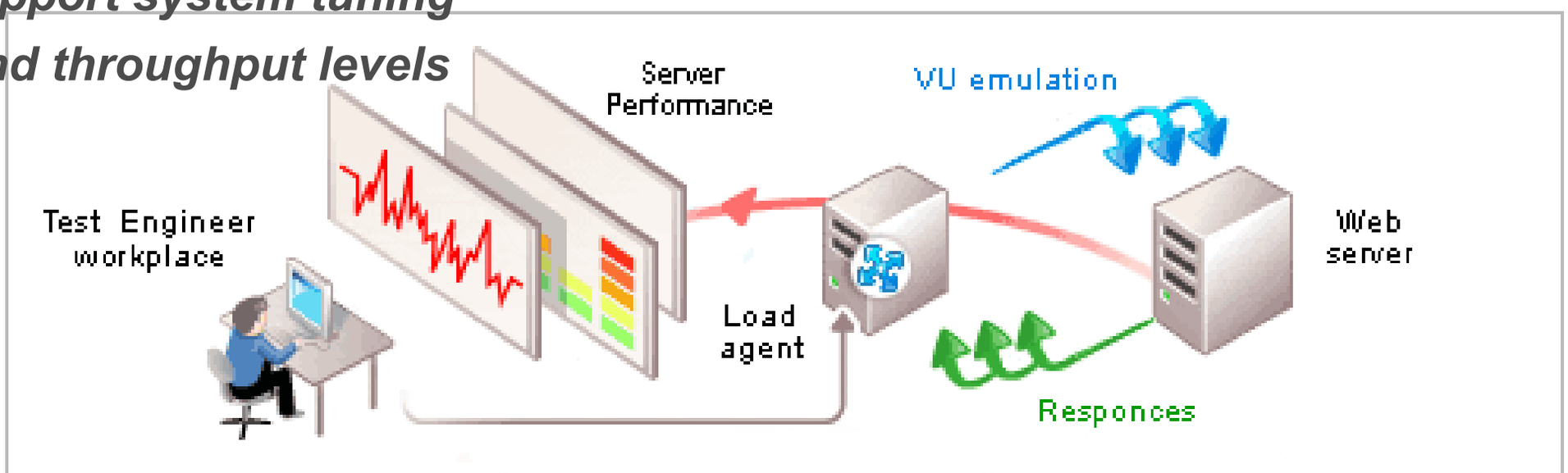
- ***the responsiveness***
- ***throughput***
- ***reliability***
- ***and/or scalability of a system under a given workload***



What is performance testing...

Performance testing is commonly conducted to accomplish the following:

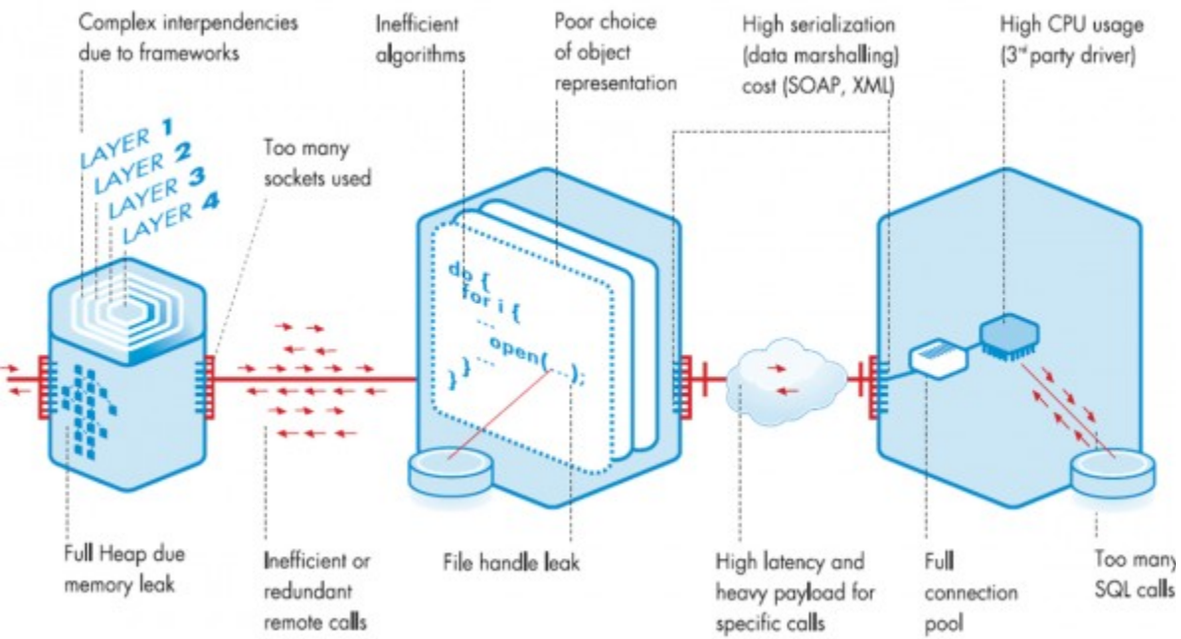
1) *Assess production readiness*
2) *Evaluate against performance criteria*
3) *Compare performance characteristics of multiple systems or system configurations*
4) *Find the source of performance problems*
5) *Support system tuning*
6) *Find throughput levels*

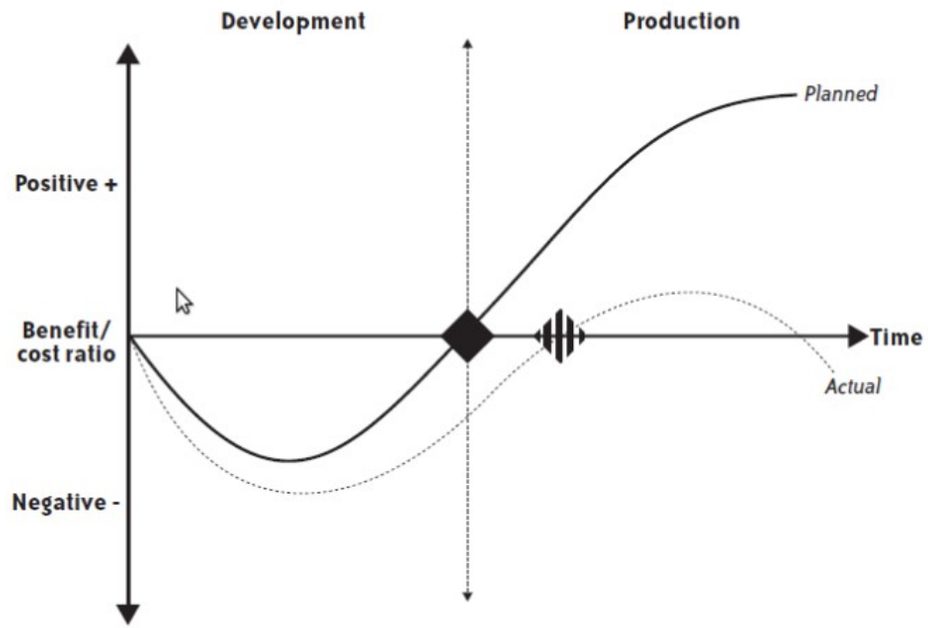


Types of Performance Testing

1. ***Performance test***
2. ***Load test***
3. ***Stress test***
4. ***Capacity test***

Why is it so important?

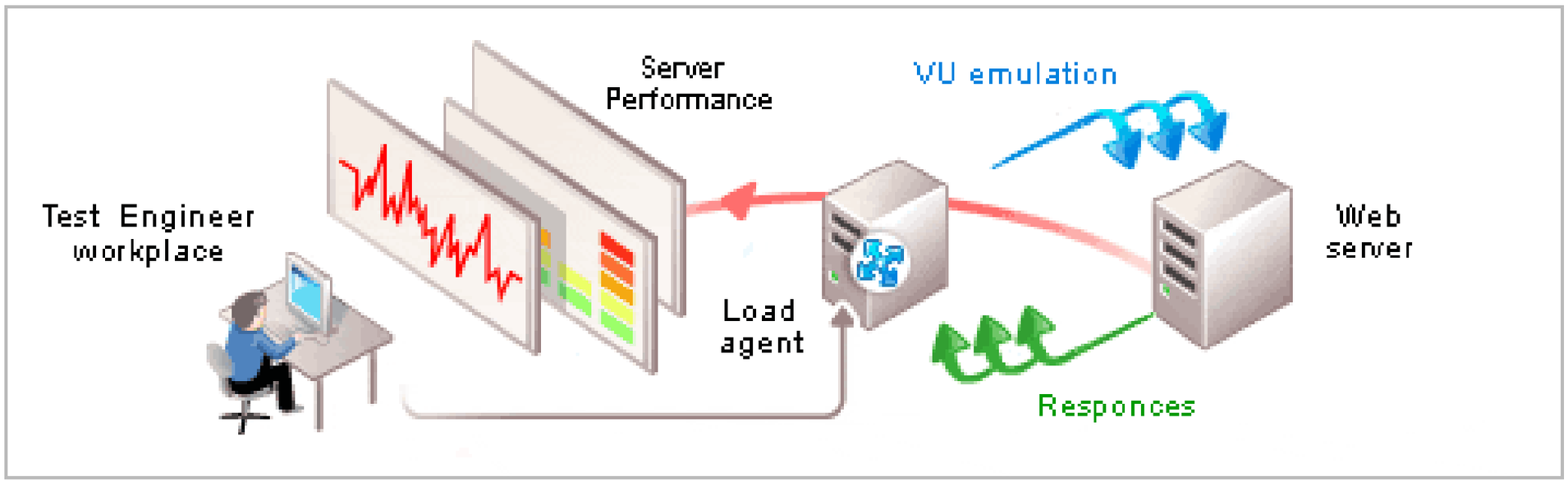




1. ***Application performance impacted by various of factors***
2. ***Last minute surprises are always bad***
3. ***Applications made up from hundred/thousand of components***
4. ***Components change day by day***
5. ***Performance issues are hard to fix***

Performance testing and monitoring tools

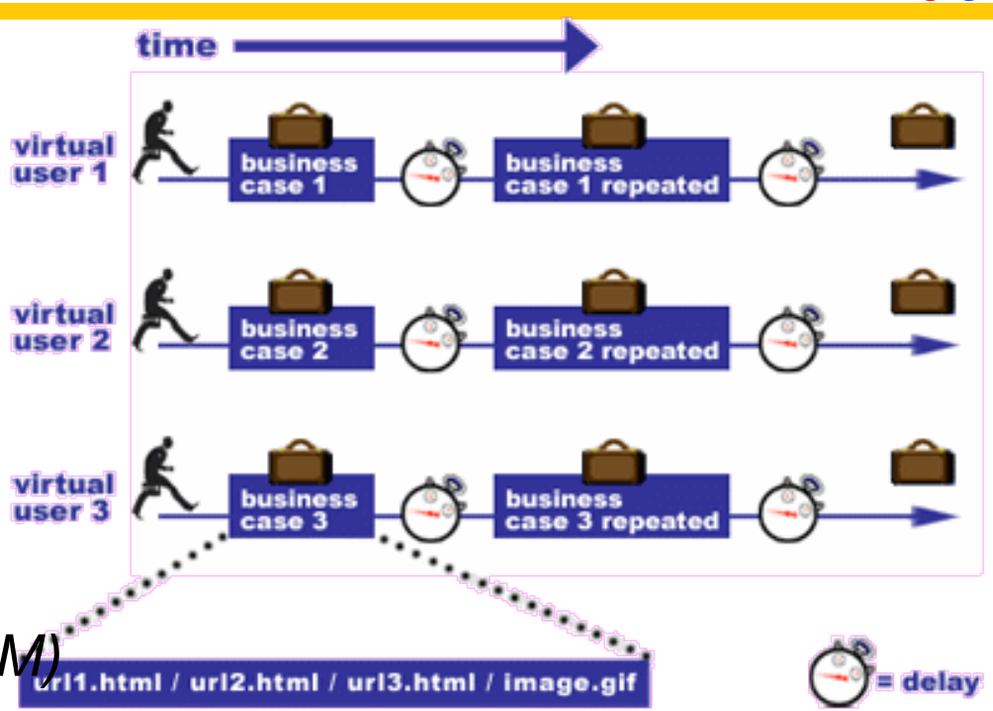
Imagine about a performance testing system



This system must provide information about:

1) ***Availability/Stability***
2) ***Response time***
3) ***Throughput***
4) ***Utilization***

Needed tools - Virtual users simulators



Can provide data:

1. Availability/Stability
2. Response time
3. Throughput

Commercial tools:

- *LoadRunner (HP)*
- *IBM Rational Performance Tester (IBM)*
- *Visual Studio Load Test (Microsoft)*

Opensource tools:

- *Apache JMeter (Apache Jakarta)*
- *The grinder*
- *OpenSTA (Open System Testing Architecture)*

Needed tools – System monitoring

Data can provide:

1. Percent of CPU utilization
2. Amount of free memory
3. Page file utilization
4. Disk time
5. Amount of free disk space

Tools:

1. *OpenNMS*
2. *Hyperic HQ*
3. *Zabbix*
4. *Perfmon (Windows)*

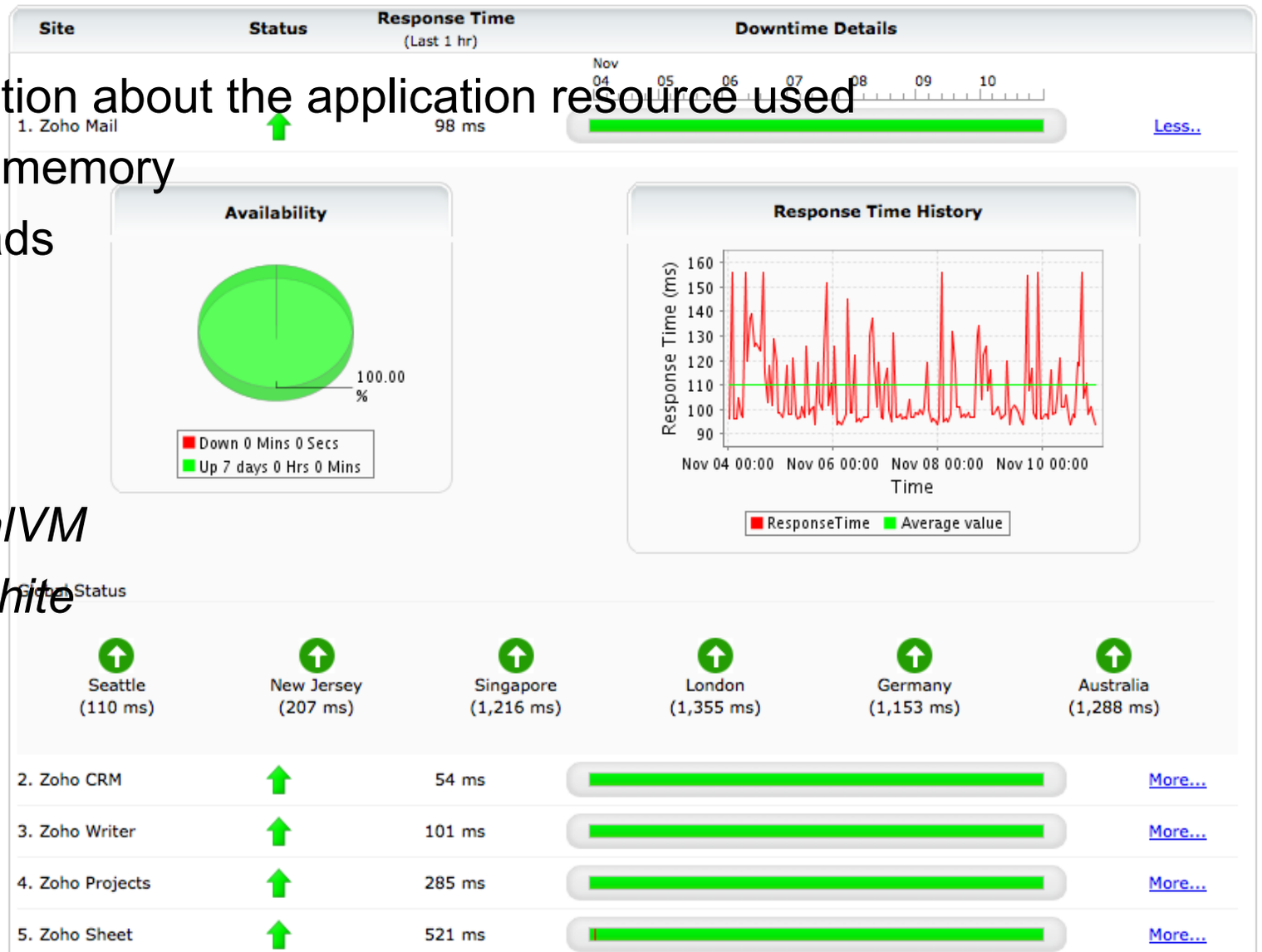
Needed tools – application monitoring

Can provide data:

1. Detailed information about the application resource used
2. Amount of used memory
3. A mount of threads
4. Openning files

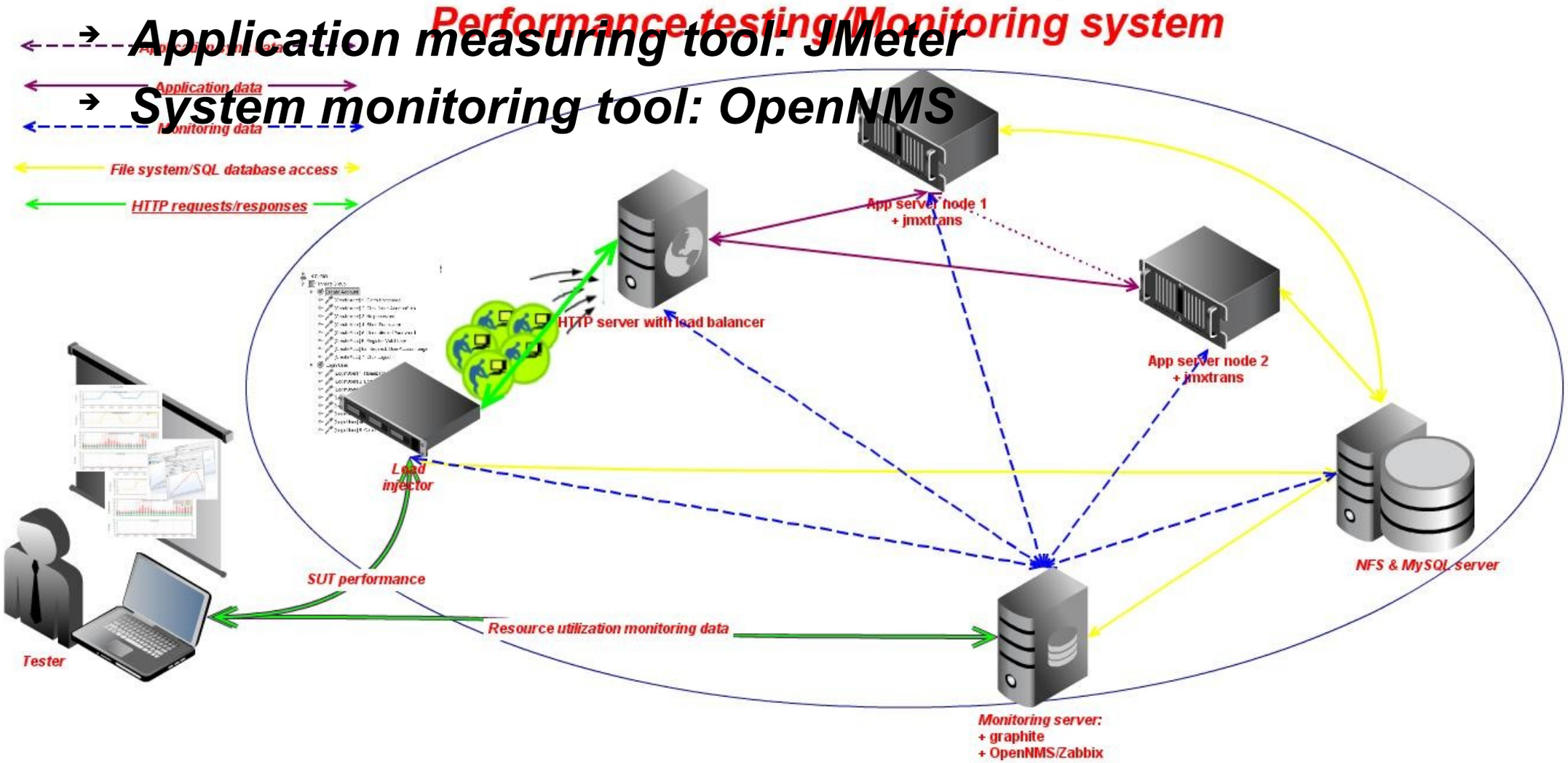
Monitoring tools:

- *JvisualVM/VisualVM*
- *Jmxtrans + graphite*
- *Jprofiler*
- *HypericHQ*
- *Jmanage*
- *Javamelody*



A real performance testing system

- ***Application monitoring tool: jmxtrans, graphite***
- ***Application measuring tool: JMeter***
- ***System monitoring tool: OpenNMS***



Performance testing with JMeter

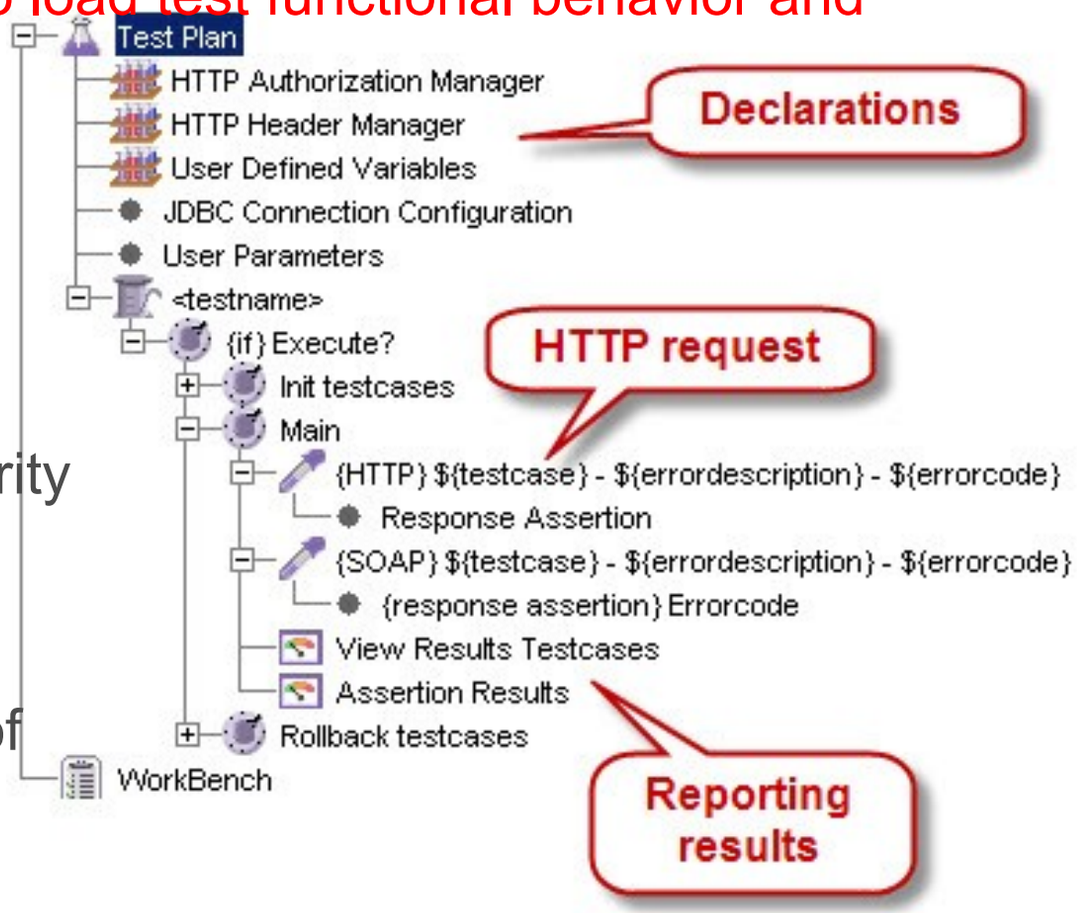
Apache JMeter introduction

The Apache JMeter™ desktop application is:

- open source software
- 100% pure Java application designed to load test functional behavior and measure performance
- designed for testing Web Application

JMeter features:

- Can load and performance test many different server types
- Complete portability and 100% Java purity
- Full multithreading framework
- Careful GUI design
- Caching and offline analysis/replaying of test results.
- Highly Extensible



Why JMeter is chosen/still be in used

+) Easy to use for both developer and tester

- ***record/modify with GUI***
- ***understand & modify under text mode***
- ***install the tool and distribute test scripts***

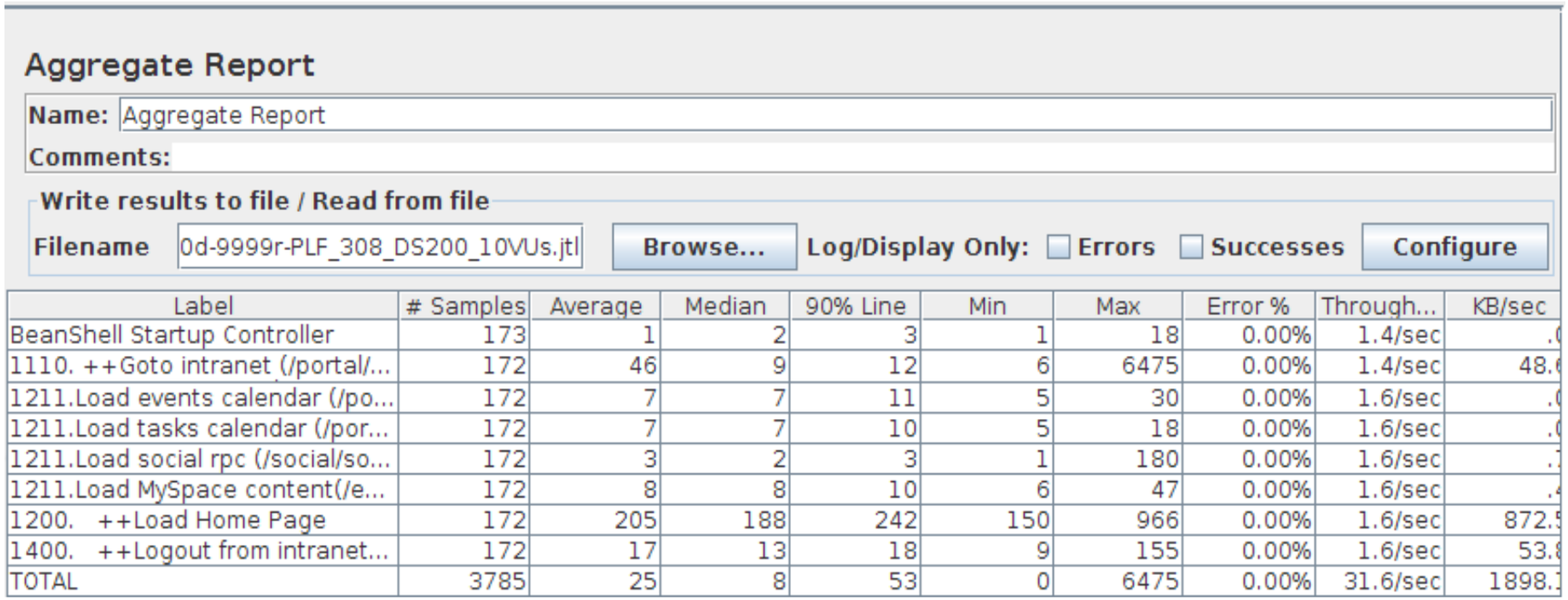
+) Good enough

- ***Rich elements that help doing performance test easily (especially web apps)***
- ***Stable***
- ***Good for both simple and advance test configuration***
- ***Good with CI/automation test model***
- ***Extendable via plugins***

+) Opensource

+-) JMeter is a great tool but poor document

Performance testing notices – Basic parameters



Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename 0d-9999r-PLF_308_DS200_10VUs.jtl Browse... Log/Display Only: Errors Successes Configure

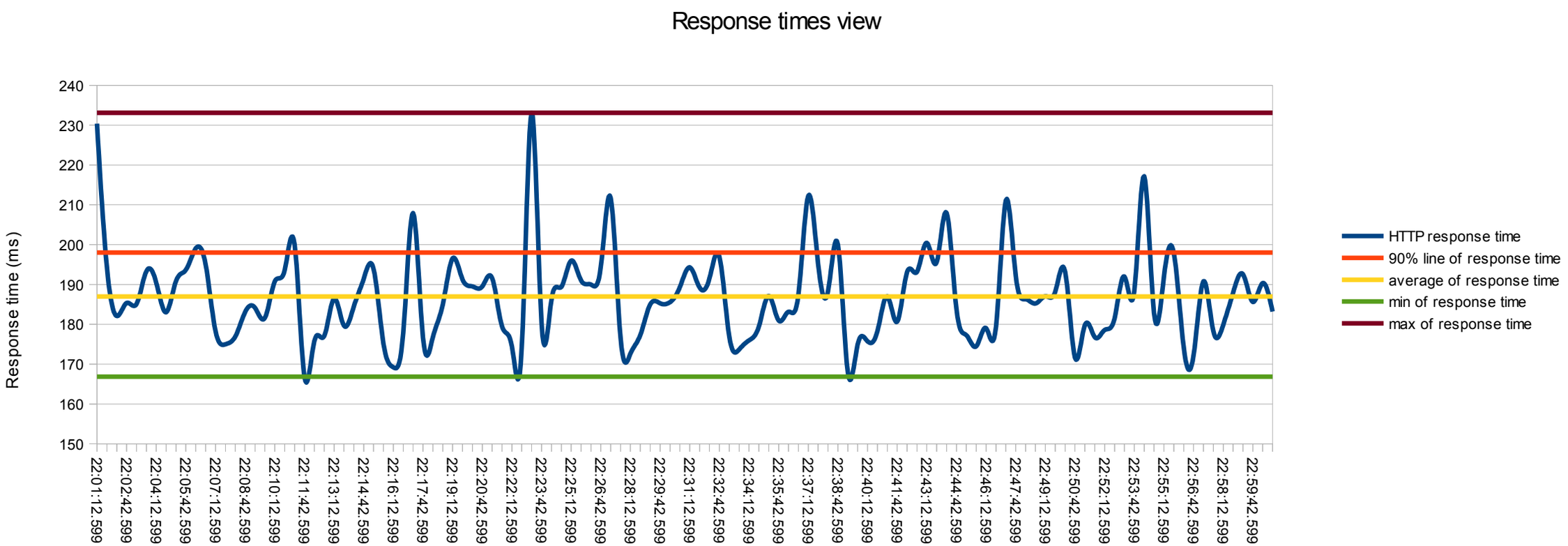
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Through...	KB/sec
BeanShell Startup Controller	173	1	2	3	1	18	0.00%	1.4/sec	.(
1110. ++Goto intranet (/portal/...	172	46	9	12	6	6475	0.00%	1.4/sec	48.(
1211.Load events calendar (/po...	172	7	7	11	5	30	0.00%	1.6/sec	.(
1211.Load tasks calendar (/por...	172	7	7	10	5	18	0.00%	1.6/sec	.(
1211.Load social rpc (/social/so...	172	3	2	3	1	180	0.00%	1.6/sec	.]
1211.Load MySpace content(/e...	172	8	8	10	6	47	0.00%	1.6/sec	.4
1200. ++Load Home Page	172	205	188	242	150	966	0.00%	1.6/sec	872.5
1400. ++Logout from intranet...	172	17	13	18	9	155	0.00%	1.6/sec	53.8
TOTAL	3785	25	8	53	0	6475	0.00%	31.6/sec	1898.1

1) *Response time*

- ***90% line of response time*** *(90th percentile of response time)*
  - *90% of all the response time values less or equal than this value)*
- *Average response time*
  - *Simple average of all values=(sum of all values from 1st to Nth)/N*
- *Min and Max response time*

Performance testing notices – Most reliable response time

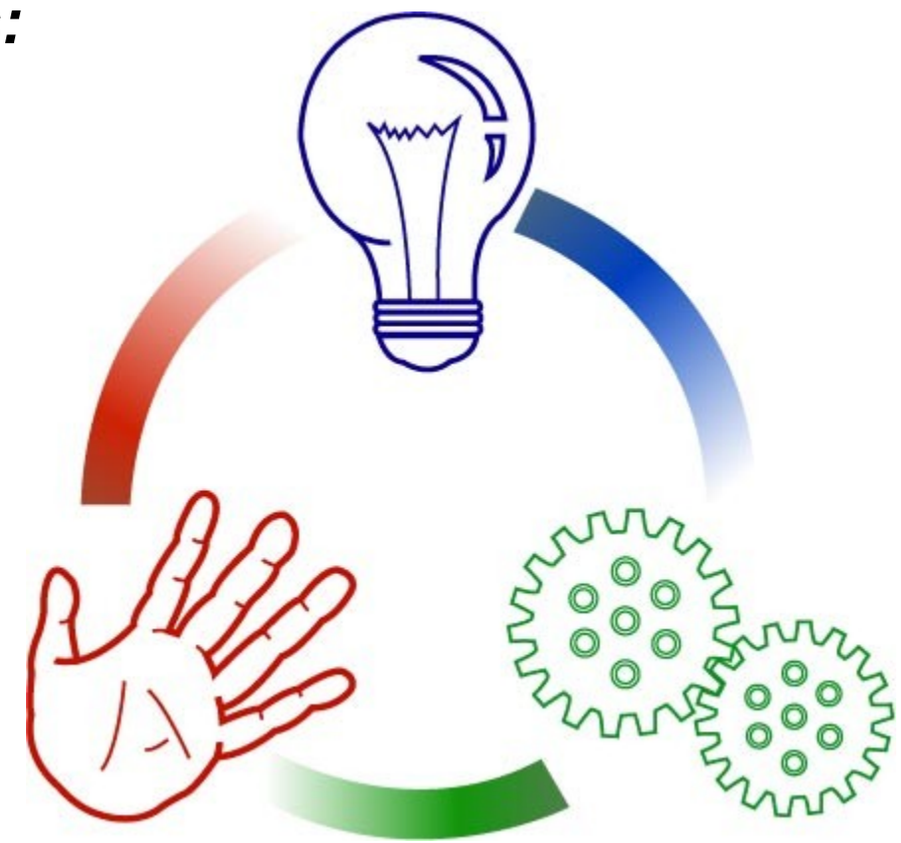
Most reliable response time = 90% line of response time (90th percentile of response time)!



Performance testing tips - Do stability test

Purpose

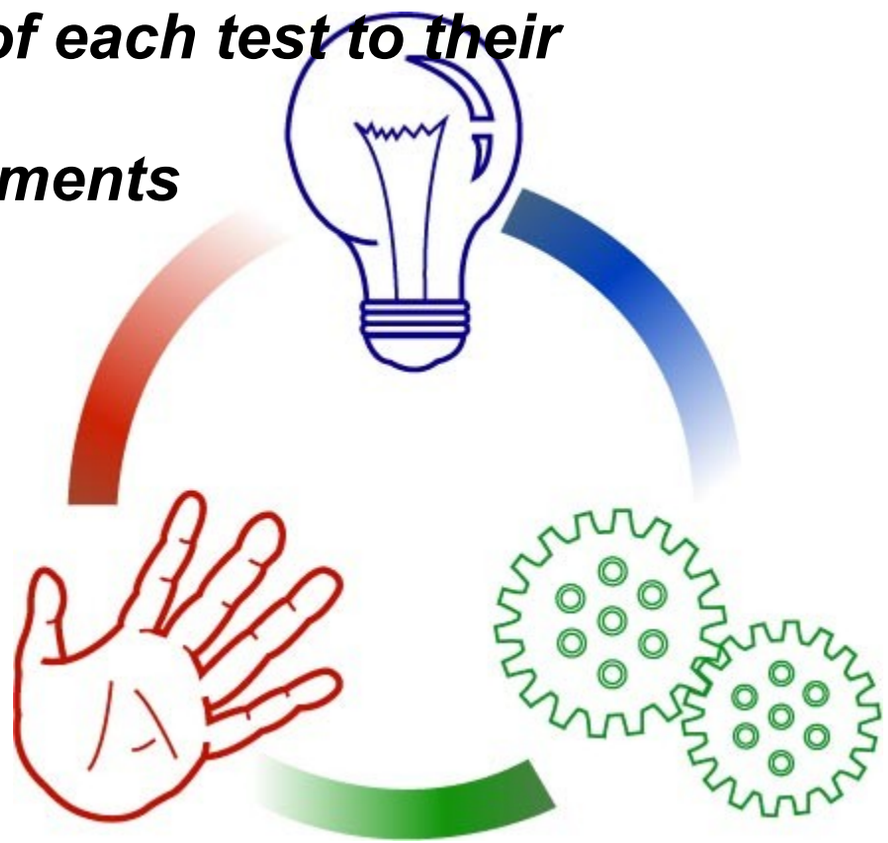
- ***Make sure that the test is reliable***
- ***Gain better knowledge about application/test***
- ***Give/confirm later tests' settings:***
  - ***Test duration***
  - ***Test cycles***
  - ***…***



Performance testing tips - Do stability test

How

- ***Choose a simple test***
- ***Repeat the test for many times (e.g: >=6 times)***
- ***Calculate difference from result of each test to their average***
- ***Apply the performance test statements***

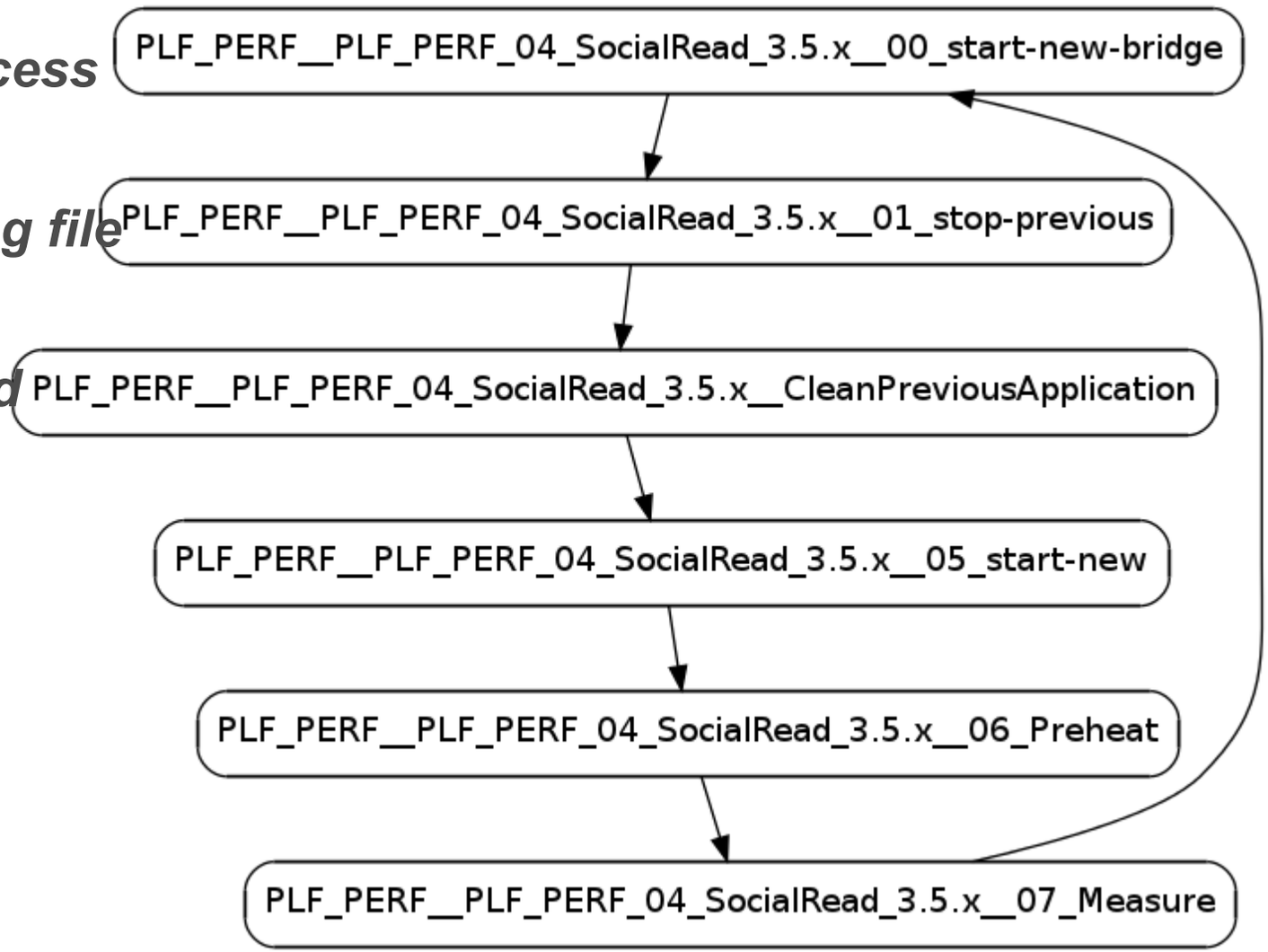


JMeter demo & notes

Demo notices – Jenkins flow

Jenkins flows:

- ***JMeter benchmark process***
- ***Perf-suite***
- ***Loop until end of setting file***
- ***Various of test settings***
- ***Easy to maintain/extend***



Demo notices – JMeter script

Introduce JMeter script

How is a script used in a benchmark flow with jenkins

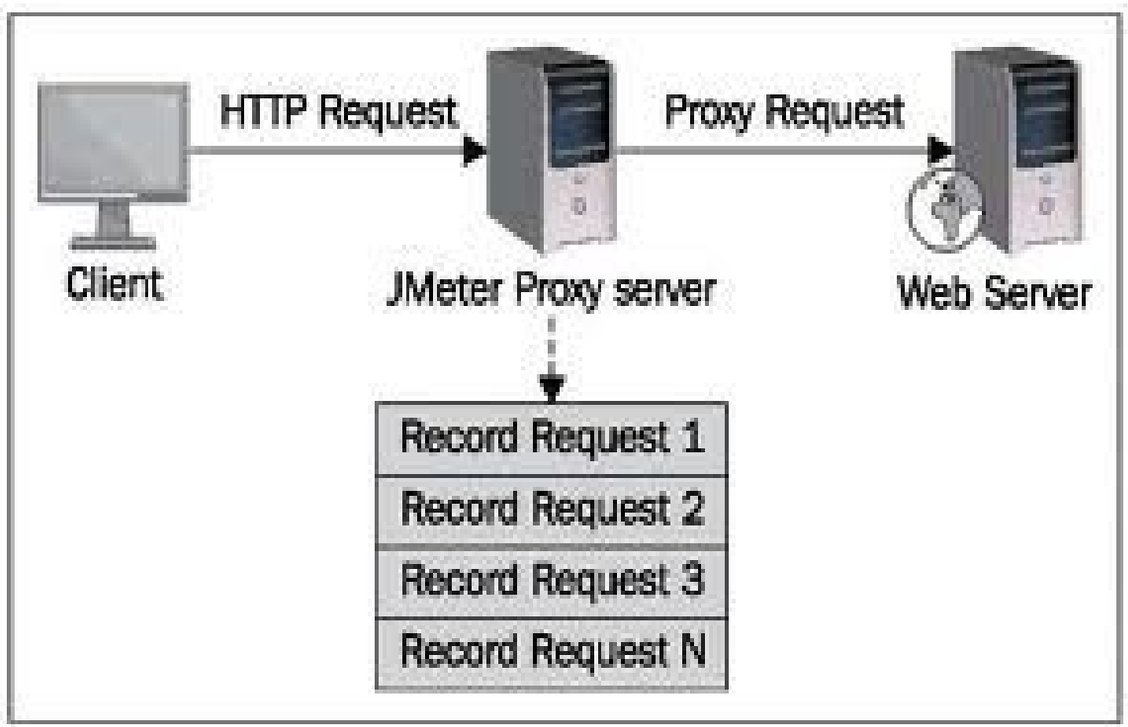
Scripts:

- *PLF_PERF_05_Login.jmx*
- *PLF_PERF_04_SocialRead-3.0.x.jmx*
- *PLF_PERF_03_ECMS-Public.jmx*

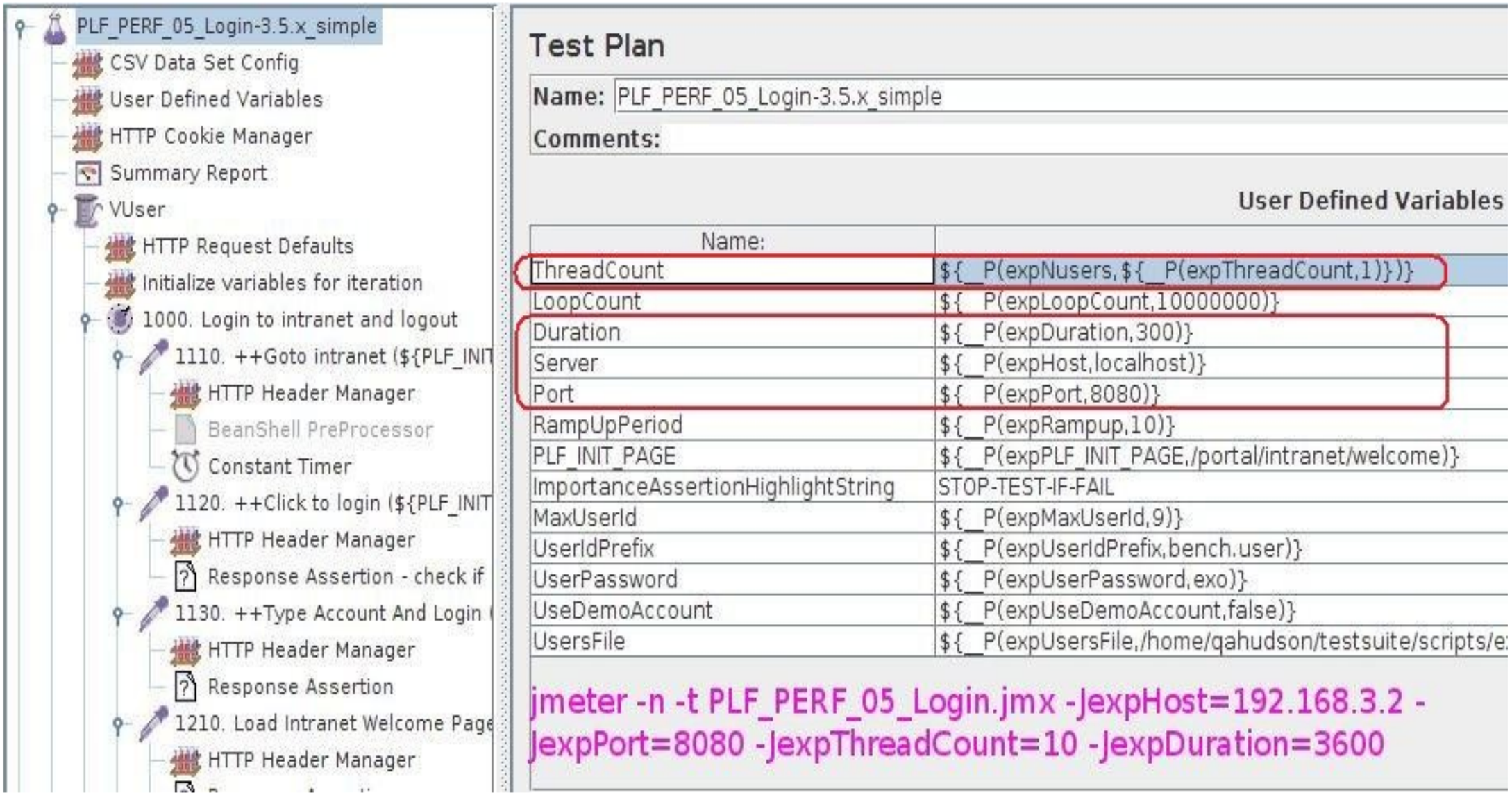
JMeter notices – recording/modifying script

Use selenium script → this is for QA who have to repeat the scenario oftenly

- ***Save and backup the first recorded version, backup after every importance change***
- ***Find and replace all the changeable values by Jmeter variables such as***
  - ***host address***
  - ***port number***
  - ***date and time***
  - ***account used in the scenario***
  - ***Id of components***
- ***HTTP response assertions are needed to avoid something wrong***



JMeter notices – script organization.p1



JMeter notices – script organization.p2

Test plan interface

- *Jmeter properties usage can be used to make up an interface for a testscript*
- *Properties' value should be assigned to runtime variables at user defined variable table and use these variables instead*
- *This will help to make a script clear, easy to understand and maintain, especially with scripts need to be run with different configurations*

Ram-up, Loop duration, Think time, Thread count

- *Rampup value helps a heavy test can be loaded smoother*
- *Loop duration: a test need to be ended with a suitable duration/loops count*
- *Think time (timer) should be used to simulate a real user*
- *Threads count will impact to test results, it is number of virtual users that you need to simulate*

Cookies, Session, Iteration

- *Cookies manager is needed in most of the cases to use the same http session for each VU (iteration)*
- *Each round of a VU is an iteration*

JMeter notices – script usage

- ***Test process (startup, warm-up, measure)***
- ***Automate test rounds (jenkins & perf-suite)***
- ***Confirm the test script with single VU (as a result of preheat)***
- ***Perf test short duration & long duration***
- ***Jmeter GUI mode vs Non-gui mode: Non-gui mode is preferred***
- ***JTL file/Listeners: jtl file is needed/enough via the command*** `jmeter -n -l output.jtl -t …` ***listeners are not needed***

Data collecting methods

- JMeter plugins and command line mode with JTL file
- Use suitable listeners in Jmeter GUI mode
- Merge results from multiple jtl files
- Monitoring tools (jmxtrans graphite, opennms/zabbix): collect via browsers

Data collecting method – JMeter-plugins command line

- *JMeter plugins and command line mode with JTL file*
  - ***java -jar CMDRunner.jar --tool Reporter --input-jtl results.jtl --plugin-type <Plugin Type Classes> --generate-png/--generate-csv <file_name> [… other options...]***

Plugin Type Classes:

- ***AggregateReport = JMeter's native Aggregate Report, can be saved only as CSV***
- ***ThreadsStateOverTime = Active Threads Over Time***
- ***HitsPerSecond***
- ***ResponseTimesDistribution***
- ***ResponseTimesOverTime***
- ***ResponseTimesPercentiles***
- ***ThroughputOverTime***
- ***TransactionsPerSecond***

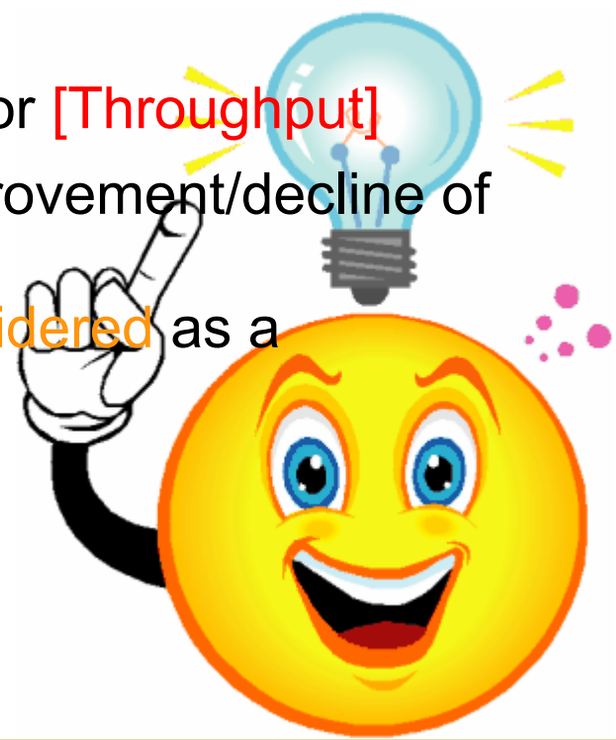
Data analysis and reporting

- **Performance test statements**
- Stability of cycles
- % of difference in response time
- Throughput distribution
- Application cycles
- Memory leak
- Session leak
- Resource utilization (see bottle necks)
- Error rate
- Throughput, 90% line of response time

Data analysis and reporting- Performance test statements

[90% line of response time] is more reliable than [average response time]

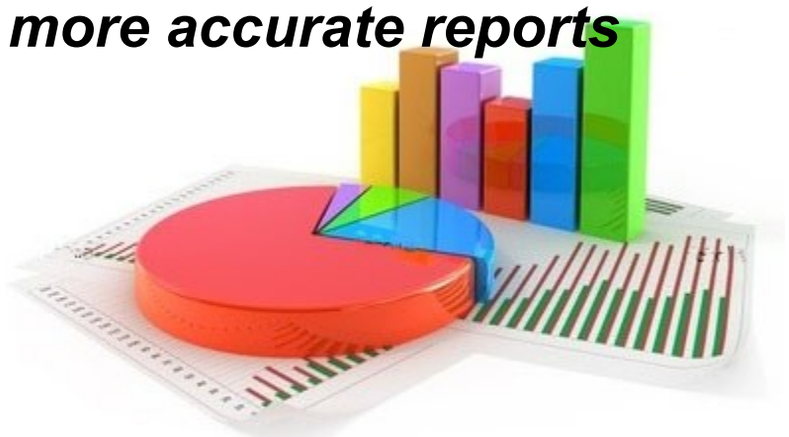
- Variation in range of [-5%,5%] should be treated as normal for [response time]
- Variation out of range [-10%,10%] should be treated as an improvement/decline of [response time]
- Variation in range of [-10%,-5%) and (5%,10%] should be considered as a decline/improvement of [response time]
- Variation in range of [-3%,3%] should be treated as normal for [Throughput]
- Variation out of range [-5%,5%] should be treated as an improvement/decline of [Throughput]
- Variation in range of [-5%,-3%) and (3%,5%] should be considered as a decline/improvement of [Throughput]



Data analysis and reporting – stability of cycles

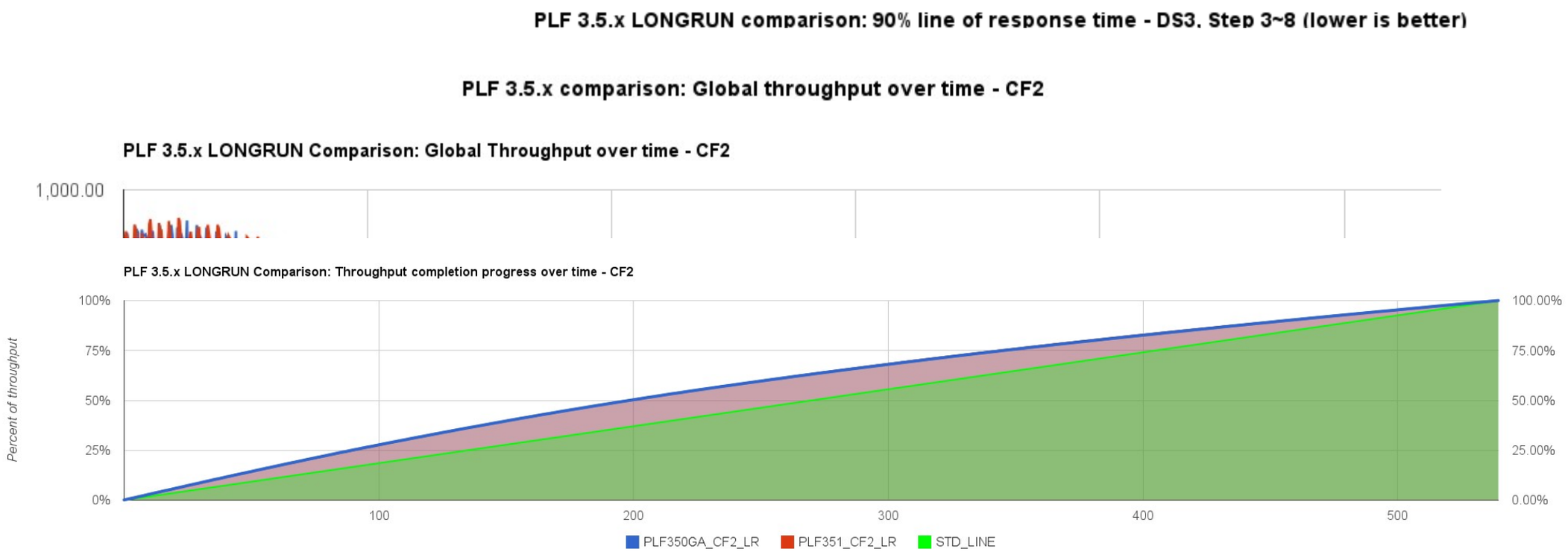
As a result of the statements:

- ***Tests cycles on the same package and same test configuration should not be vary out of range [-5%, 5%] in comparison to their aggregation results***
- ***The aggregation of cycles, itself, makes the test result more accurate, reduces error but unstable results still need to be checked.***
- ***The stability of test rounds help us to give more accurate reports***



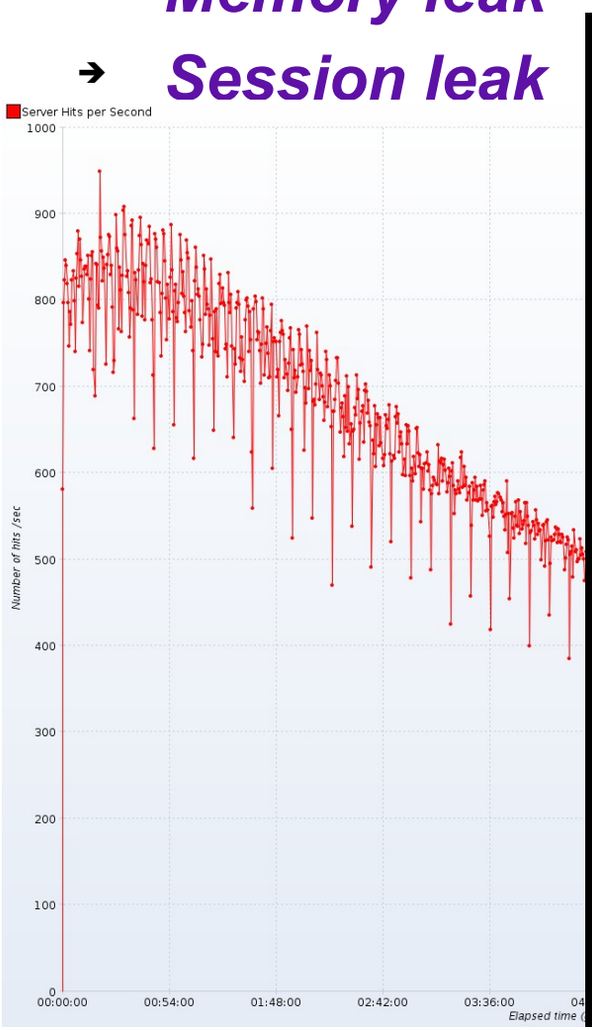
Data analysis and reporting – detail

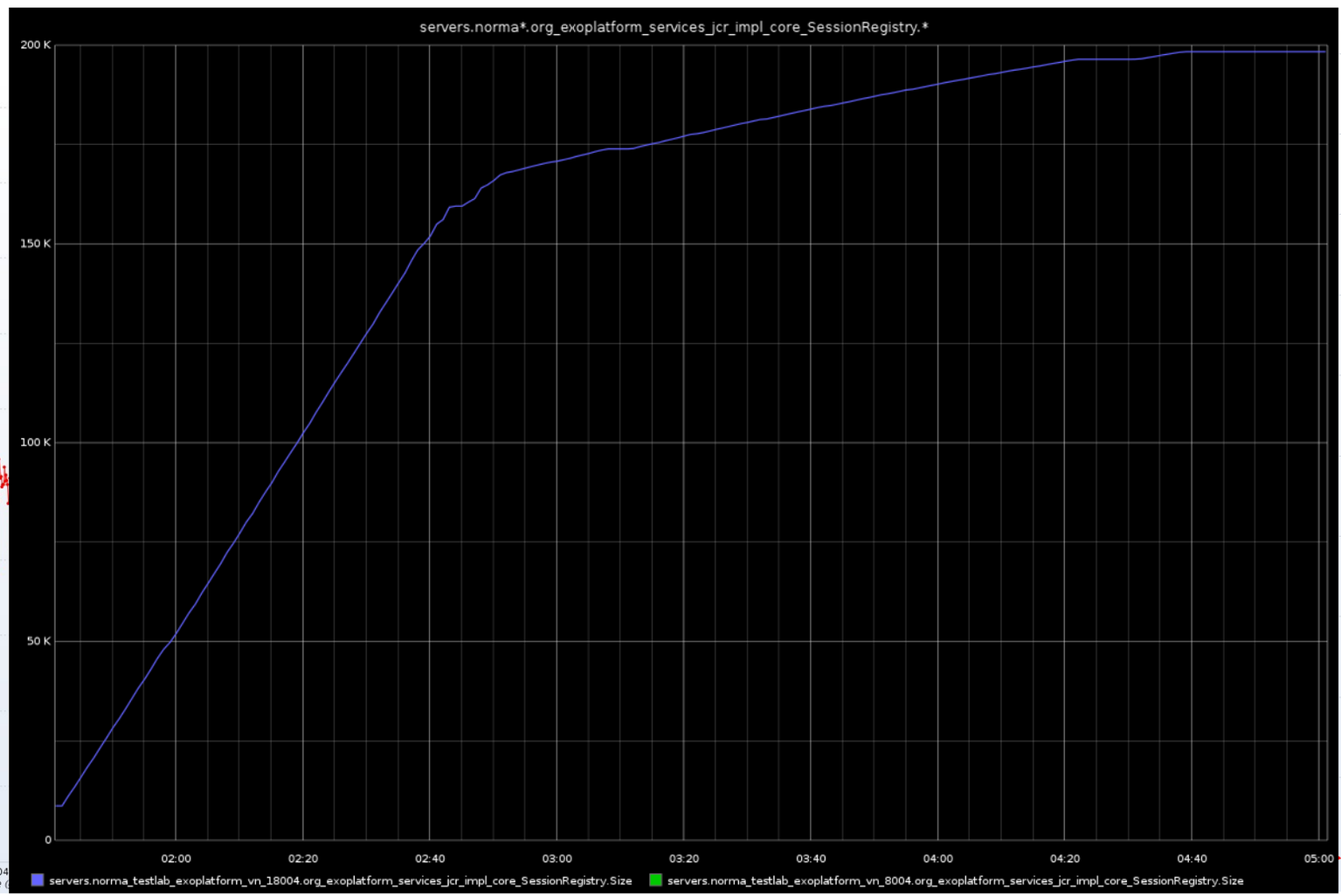
- *% of difference in response time*
- *Throughput distribution*
- *Application cycles*

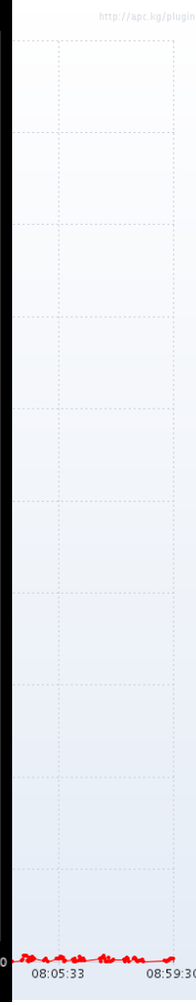


Data analysis and reporting – detail

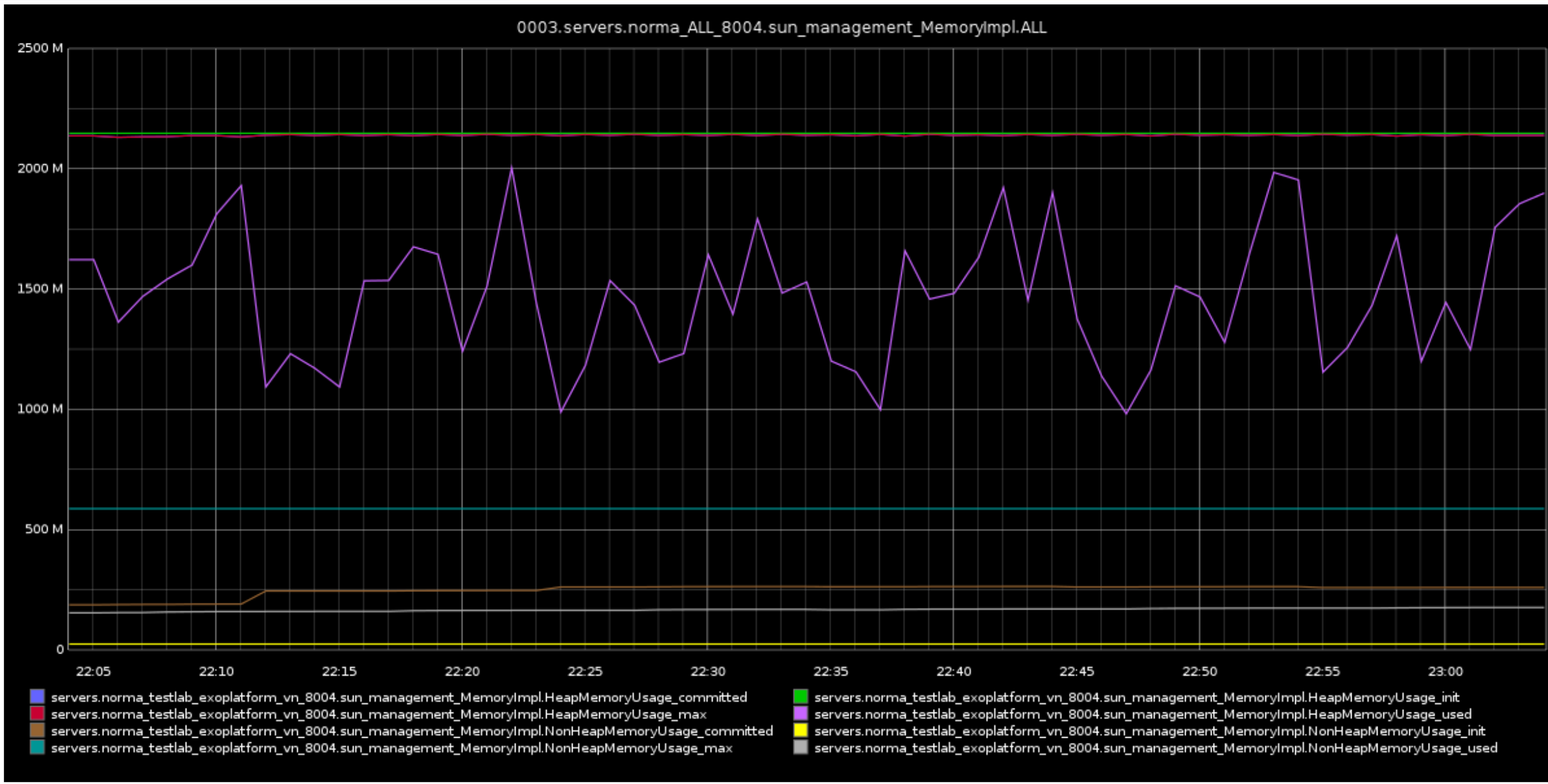
- ***Memory leak***
- ***Session leak***







Data analysis and reporting – resource utilization



What to focus in a performance test report

A real performance test report explanation

- ***TC-PLF-3.5.1 PLF_PERF_04_SocialRead explanation***
  - ***https://wiki-int.exoplatform.org/display/exoReleases/TC-PLF-3.5.1+PLF_PERF_04_SocialRead***

Tips & tricks

1. - A clear name for each performance test
2. - non-GUI mode is more stable than GUI mode
3. - Do not use listeners if not needed
4. - Rampup is needed for heavy load
5. - Assertion is needed to simulate a virtual user
6. - Shutdown.sh → save your effort while doing a long test
7. - Unstable tests: think of data while an user run its scenario
8. - If one user can not login, its later steps should not be counted
9. - Backup after every important step you made to your script easily by cloning the jmx file
10. - Speedup jmeter script modifying with text editors which support regex, e.g: kate

Links

1. Http://code.google.com/p/jmeter-plugins/
2. http://jmeter.apache.org/
3. http://code.google.com/p/jmxtrans/wiki/IntroductionToJmxTrans?tm=6
4. https://wiki-int.exoplatform.org/display/QAF/Graphite
5. https://wiki-int.exoplatform.org/display/TQA/Performance+test+suites+and+u
6. https://wiki-int.exoplatform.org/display/TQA/PLF_PERF_04_SocialRead-3.0.x
7. https://wiki-int.exoplatform.org/display/TQA/PLF_PERF_04_SocialRead-3.5.x
8. https://wiki-int.exoplatform.org/display/exoReleases/TC-PLF-3.5.1_TOMCAT
9. https://wiki-int.exoplatform.org/display/exoReleases/TC-PLF-3.5.1+PLF_PER

Q & A

Thank you!