Performance Testing and Monitoring

A Guided Tour Through CLIF Tools and Features

Bruno Dillenseger
Orange Labs
Performance testing and monitoring

→ **Performance testing**
  - emulate real user workload with load injectors and
  - check the behavior of a system under test through selected metrics

→ **Performance monitoring**
  - generate a light traffic on an on-line system with real users
  - check metrics about user Quality of Experience.
CLIF in a nutshell

- A generic and adaptable Java framework for distributed performance testing and monitoring
  - traffic generators measuring response times and throughput
    - supported protocols: TCP, UDP, DNS, FTP, GIT, HTTP, IMAP, JDBC, JMS, LDAP, MQTT, RTP, SIP, SVN, TCP
    - write/wrap your own protocols
    - write/wrap your own data set providers
  - probes measuring resources usage
    - CPU, disk, RAM, network, JVM, RTP
    - write/wrap your own probes

- High power
  - up to 1000 load injectors in parallel X millions of virtual users...
  - ... but also OK with 1 virtual user on a single load injector
Happy 15th birthday to CLIF!

- CLIF has been developed, used and adapted following trends of computing science R&D
  - Component-Based Software Engineering
    - component-based distributed architecture based on OW2 Fractal model
  - Java technologies
    - CLIF is pure Java, CLIF's main GUI is based on Eclipse IDE
  - Autonomic computing
    - control loops for self-driven test campaigns (CLIF/Selfbench module)
  - Cloud Computing for on-demand testing infrastructures
    - OpenStack, OW2 ProActive Scheduling & Workflows
  - Automation and containers
    - CLIF plug-in for Jenkins
    - CLIF Docker image
Main CLIF modules/distributions

- CLIF server
  - basic CLIF runtime with full-fledged command line interface
  - required wherever you intend to run a load injector or a probe

- CLIF swingGui
  - CLIF server enhanced with a simplified GUI for running tests and building custom performance reports

- CLIF Eclipse-based console (main GUI)
  - full-fledged GUI for writing and running complex test scenarios
  - IDE for extending CLIF itself
    - e.g. Eclipse wizard for writing custom plug-ins for load injection

- CLIF plug-in for Jenkins
  - automates test runs, performance reporting, QoE alerting
Installing and using CLIF

➔ Download a distribution from clif.ow2.org
  ▪ unzip
  ▪ dependency: java 8+ runtime

➔ Get CLIF Performance Testing plug-in for Jenkins
  ▪ Manage Jenkins > Manage Plugins > Available

➔ Get container image from Docker hub
  ▪ docker pull dillense/clif

➔ Get support from clif.ow2.org
  ▪ user guide, javadoc, tutorials, videos, examples, presentations...
  ▪ e-mail clif@ow2.org
Command line interface

clifcmd ...

- analyze
- change testplan_name id param_name param_value
- collect testplan_name [id1:id2:...idN]
- config [registry_host[:registry_port] [codeserver_host[:codeserver_port]]]
- deploy testplan_name testplan_file
- gui*
- help
- init testplan_name testrun_id
- join testplan_name [id1:id2:...idN]
- launch testplan_name testplan_file testrun_id

* this command is available only with distribution clif-swingGui

- listservers [test plan file names...]
- params testplan_name id
- probehelp probe_type
- quickstats [report_directory]
- registry
- resume testplan_name [id1:id2:...idN]
- run testplan_name testrun_id [id1:id2:...idN]
- server [name]
- start testplan_name [id1:id2:...idN]
- stop testplan_name [id1:id2:...idN]
- suspend testplan_name [id1:id2:...idN]
- version
- waitservers [testplan_file]

Available for Windows, MacOSX, Linux, etc.
Defining and running tests
Custom performance report
Build Performance Report
calcudp20x2 - Tue Jan 24 15:33:21 CET 2017

Server - clif2

**Probe - jvm ( Argument=1000 60 Comment= )**

<table>
<thead>
<tr>
<th>Label</th>
<th>Samples</th>
<th>Average</th>
<th>Median</th>
<th>Min</th>
<th>Max</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>free memory (MB)</td>
<td>61</td>
<td>131</td>
<td>131</td>
<td>119</td>
<td>146</td>
<td>7.75</td>
</tr>
<tr>
<td>used memory %</td>
<td>61</td>
<td>15</td>
<td>16</td>
<td>4</td>
<td>23</td>
<td>5.14</td>
</tr>
<tr>
<td>free usable memory %</td>
<td>61</td>
<td>89</td>
<td>90</td>
<td>84</td>
<td>97</td>
<td>3.38</td>
</tr>
</tbody>
</table>

**Injector - Inj2 - IsacRunner ( Argument=calcudp20.xis Comment= )**

<table>
<thead>
<tr>
<th>Type</th>
<th>Requests</th>
<th>Success</th>
<th>Errors</th>
<th>Average</th>
<th>Median</th>
<th>Min</th>
<th>Max</th>
<th>Standard deviation</th>
<th>Throughput</th>
<th>Error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP CONNECT</td>
<td>20</td>
<td>20</td>
<td>0</td>
<td>36</td>
<td>36</td>
<td>23</td>
<td>66</td>
<td>11.38</td>
<td>0.32</td>
<td>0%</td>
</tr>
<tr>
<td>UDP RECEIVE</td>
<td>8427</td>
<td>8,265</td>
<td>162</td>
<td>1429</td>
<td>1060</td>
<td>8</td>
<td>136096</td>
<td>3238.21</td>
<td>133.64</td>
<td>1.92%</td>
</tr>
<tr>
<td>computation error</td>
<td>883</td>
<td>0</td>
<td>883</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>UDP SEND</td>
<td>8427</td>
<td>8,427</td>
<td>0</td>
<td>36</td>
<td>35</td>
<td>7</td>
<td>1206</td>
<td>35.26</td>
<td>136.26</td>
<td>0%</td>
</tr>
</tbody>
</table>
CLIF plug-in for Jenkins: performance report graphs
CLIF plug-in for Jenkins: performance trend

Clif Performance Trend Report

UDP CONNECT

Percentage of errors

UDP RECEIVE

Percentage of errors
What's coming next?

➔ More automation for users
  - OW2 ProActive-based deployment over Metal as a Service
    - collaboration with ActiveEon
  - Ansible roles and playbooks
  - OpenStack environment

➔ Evolutions for developers
  - move to OW2's Gitlab instance...
  - ... coming with big code base reorganization
  - Continuous Integration: Bamboo to be replaced by Gitlab CI
  - move from "legacy" CLIF to "ProActive CLIF"

➔ Your Questions?