> Introducing FunkLoad

Open Source HTTP testing tool

Stefane Fermigier <sf@nuxeo.com>

Tests

> “If it's not tested, it's broken”
>
> Bruce Eckel, Thinking in Java, 3rd edition

Where do we come from?

- FLOSS ISV (based on Zope, Eclipse, OOo)
- Active R&D in several FLOSS communities (Zope, CPS, Eclipse, OOo)
- FLOSS products / customers projects dichotomy
- Projects built on tens of different components, with complex configurations
- Work with partners and external FLOSS contributors
- We have assessed or tested, and sometimes put into production, many of the FLOSS tools relevant to our own needs over the past 4 years

Automated testing

- Mandatory for agile software development
  - Test-driven development [Beck]
  - Non-regression testing
- Mandatory for “agile” (aka dynamic) programming languages
  - “Duck typing” vs. static typing
  - Read “Strong Typing vs. Strong Testing” [Bruce Eckel]

Types of testing

- White box testing
  - Programmers tests
  - Our focus is on Python
- Black box testing
  - Testers tests

White box testing

- Unit testing
- Integration testing

Black box testing

- Functional testing
- Systems testing
- Load / stress testing

FunkLoad

- Functional + load testing
- Written and scripted in pure python
- Leverages several FLOSS libraries like PyUnit and WebUnit
- Developed as a first step towards the QSD portal after a review of the existing FLOSS solutions

State of the art

- FLOSS tools
- Proprietary tool
- References
  - opensourcetesting.org
  - agiletesting.blogspot.com
  - clif.forge.objectweb.org/load_tools_overview.pdf
  - freshmeat.net categories: Software Development :: Testing

Floss tools – Unit tests

- xUnit
  - Family of similar frameworks for Java, C++, C#, SmallTalk, Ruby, etc.
- DocTest
  - Zope, then Python, thing
  - Combines tests and documentation (= storytelling) in one place
  - Works with PyUnit et al.
  - Goal and test-driven development

FLOSS tools – HTTP Functional tests

- Puffin
  - Written in Python, but dead (we tried using it in 2003)
- MaxQ
  - Java project (scripted in Jython), has too many bugs currently, progressing slowly
- Selenium
  - New (2005) and very popular
  - Works in browser (JavaScript)
  - Not clear how to use it for CI tests (hard to automate)
- …

FLOSS tools – Perf tests (1/2)

- Apache Bench (Apache Foundation)
  - Single URL perf testing
  - Very useful for what it does
  - Not good for testing transactional scenarios
- Apache Jmeter (Apache Foundation)
  - Coded in Java
  - Not scriptable
- Deluge, Flood
  - Dead

FLOSS tools – Perf tests (2/2)

- IDX-Tsunami (IDEALX)
  - Distributed, coded in Erlang, very hard to install
- OpenSTA
  - Win32 platform only
- TestMaker (PushToTest)
  - Coded in Java / scripted with Jython, not based on PyUnit, was very buggy at the time we needed it
- …

Proprietary tools

- Mercury WinRunner / LoadRunner
  - So expensive that we haven't been able to test them
- Parasoft WebKing
  - Complex GUI for capturing user scenarios
  - Scripted in Jython
  - Optional distributed performance testing
- ...
- Usually very expensive (> 5 kE / seat), so we couldn't conduct extensive tests

Zope testing ecosystem (1/2)

- (Zope3 only) Pure unit tests
- (Zope2) Pseudo unit tests
  - Use ZopeTestCase
  - Need either mock objects, or complex test harness setup (ZopeTestCase / CPSTestCase)
- Unit doctests

Zope testing ecosystem (2/2)

- White box functional tests
  - Browser (external) request emulation from the test harness
  - Functional doctests
  - Mechanize-based functional tests
- Black box functional HTTP tests
  - Using an external tool, doesn't need to be written in Python

State of the arts conclusions

- Good or even excellent white box testing tools available for our purpose
  - We've already been using them, for 3 years, with great impact on quality
- No adequate black box FLOSS testing tools covering our needs

Why FunkLoad? (1/2)

- Existing solutions were inadequate for our needs
- Pure Python, easy to install and configure
- One tool for functional tests and several kinds of performance tests (load and stress testing)
- Same scripts for functional and performance tests

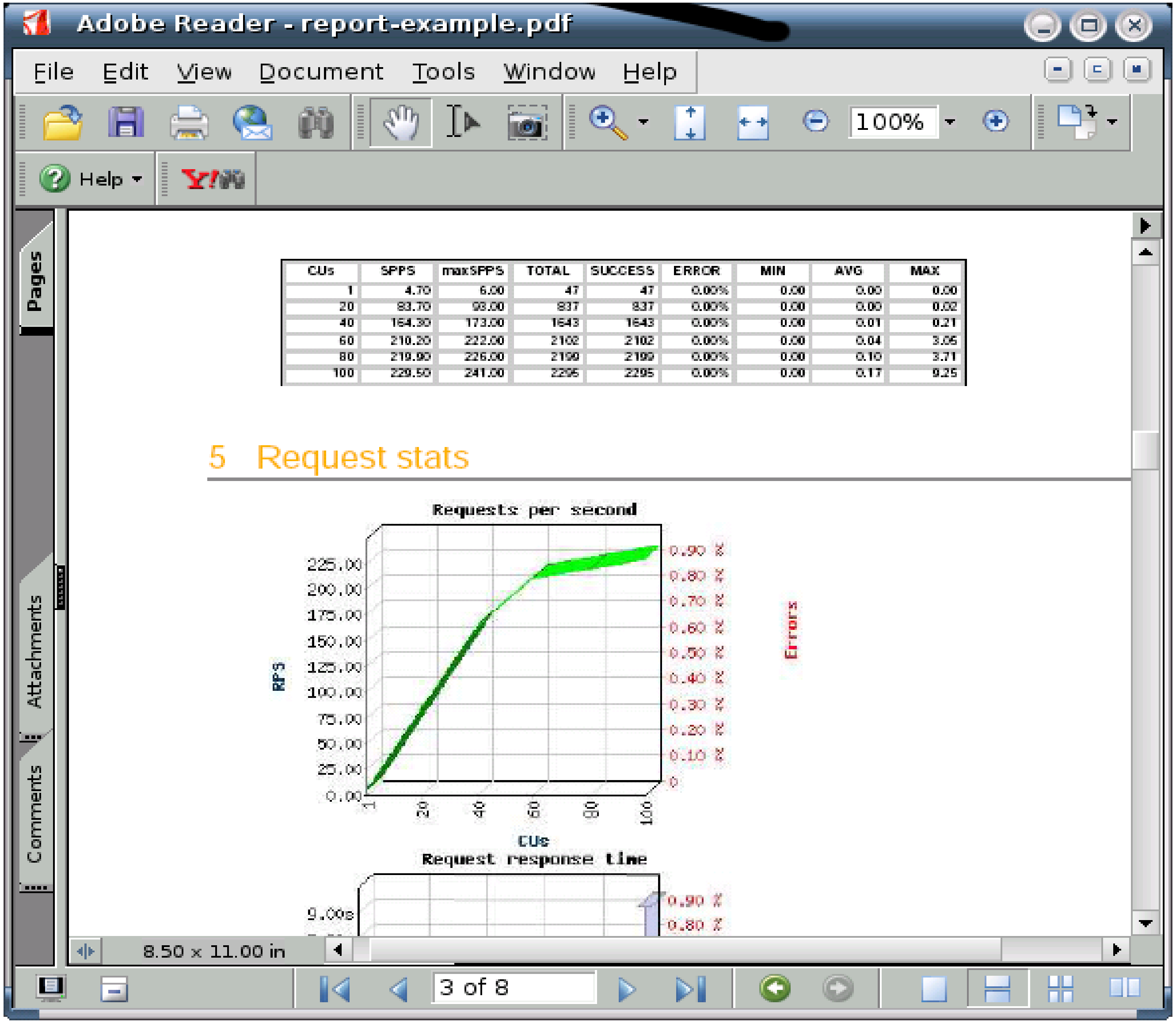
Why FunkLoad? (2/2)

- Same programming paradigm as the popular PyUnit / xUnit framework
- Tests scenarios import from tcpwatch recorder (proxy-based)
- Detailed, readable output (charts)
- Provides basic services like
  - Linux monitoring
  - Credential server
  - Random data generator

FunkLoad TODO

- Recording sessions with the new, browser-based, more user-friendly, but immature, TestGen4Web [SpikeSource R&D]
- Plug into PerfBase [NEC R&D] for statistical data collection
- HTML/CSS/... validation (W3C tools like Tidy)

FunkLoad - “Screenshot”



CUs	SPPS	maxSPPS	TOTAL	SUCCESS	ERROR	MIN	AVG	MAX
1	4.70	6.00	47	47	0.00%	0.00	0.00	0.00
20	83.70	93.00	837	837	0.00%	0.00	0.00	0.02
40	164.30	173.00	1643	1643	0.00%	0.00	0.01	0.21
60	210.20	222.00	2102	2102	0.00%	0.00	0.04	3.05
80	219.90	226.00	2199	2199	0.00%	0.00	0.10	3.71
100	229.50	241.00	2295	2295	0.00%	0.00	0.17	9.25

5 Request stats

FunkLoad – Not in scope (yet?)

- Distributed test clients
- Non-HTTP testing

Applications of FunkLoad (1/2)

- Nuxeo+Capgemini: load testing of a large-scale application for the French Ministry of Justice
  - Helped identify bottlenecks in the application, and achieve the targeted performance level for the application
- Nuxeo/CPS Project: CPS performance enhancement
  - Helped remove performance bottlenecks in the transaction machinery

Applications of FunkLoad (2/2)

- Zope project: Zope3 load testing
  - Initial tests carried out on the SVN trunk of the Zope3 application server
- Nuxeo/CPS Project: CPS automatic testing (buildbot)
  - TBD RSN
- Nexedi: ERP5 testing
  - TBD RSN

The end

- More info:
  - About FunkLoad:
    - http://funkload.nuxeo.org/
  - About Nuxeo:
    - http://www.nuxeo.com/
    - contact@nuxeo.com
  - About CPS:
    - http://www.cps-project.org/
    - Mailing lists: http://lists.nuxeo.com/