Automated testing: Past, Present and Future

Julio Merino <jmmv@NetBSD.org> for BSDCan 2011
Goals

- NetBSD is where testing is happening
- Show what's coming in the near future
Goals

- NetBSD is where testing is happening
- Show what's coming in the near future
- Get other BSD systems involved!
## Contents

<table>
<thead>
<tr>
<th>Past</th>
<th>What led to ATF?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present</td>
<td>What is ATF?</td>
</tr>
<tr>
<td></td>
<td>Other testing technologies</td>
</tr>
<tr>
<td>Future</td>
<td>A critique of ATF</td>
</tr>
<tr>
<td></td>
<td>Meet Kyua</td>
</tr>
</tbody>
</table>
Why testing?

- Software has bugs
- Automated testing helps reducing bugs
• No consistency
• Not used; instead, ad-hoc testing
• What about release engineering?
What is ATF?

• Simply put: a testing framework
  • Libraries: C, C++, Shell
  • Tools: run, report
• Since 2007 (SoC) until now
ATF in NetBSD

• For developers:
  • Maintain stable current
  • Help release engineering

• For users:
  • Confidence
Related Technologies

- rump
- puffs, pud and others
- anita

- Development policies
- Dashboards
Continuous Builds

May 2011 [details]

April 2011 [details]

March 2011 [details]
Continuous Builds
## Releng Dashboard

### The NetBSD Project

"Of course it runs NetBSD"

#### Test Run Logs

Disclaimer: The logs are provided by individuals and are not an official service of The NetBSD Project. Frequencies are approximate and subject to the availability of the people running the tests.

<table>
<thead>
<tr>
<th>Port/version</th>
<th>NetBSD login</th>
<th>Main page</th>
<th>Latest Results</th>
<th>Frequency</th>
<th>Platform</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha/-current</td>
<td>njoly</td>
<td>link</td>
<td>Test log</td>
<td>manual</td>
<td>Digital PWS500au</td>
<td>QEMU clock is not accurate and causes sporadic failures in some tests</td>
</tr>
<tr>
<td>amd64/-current</td>
<td>pogoyette</td>
<td>link</td>
<td>Test log</td>
<td>daily</td>
<td>QEMU/Anita</td>
<td></td>
</tr>
<tr>
<td>amd64/-current</td>
<td>jun</td>
<td>link</td>
<td>see page</td>
<td>biweekly</td>
<td>Thinkpad X61</td>
<td></td>
</tr>
<tr>
<td>hpcmips/-current</td>
<td>jun</td>
<td>link</td>
<td>see page</td>
<td>biweekly</td>
<td>W2000/WS011SH</td>
<td></td>
</tr>
<tr>
<td>hps/st/-current</td>
<td>jun</td>
<td>link</td>
<td>see page</td>
<td>biweekly</td>
<td>sigmarion</td>
<td></td>
</tr>
<tr>
<td>i386/-current</td>
<td>gson</td>
<td>link</td>
<td>top of page</td>
<td>daily</td>
<td>QEMU/Anita</td>
<td></td>
</tr>
<tr>
<td>macppc/-current</td>
<td>riz</td>
<td>link</td>
<td>Test log</td>
<td>Manually</td>
<td>Power Mac G4/400</td>
<td>pthread mutex problems and sporadic failures in tests using them</td>
</tr>
<tr>
<td>sparc64/-current</td>
<td>martin</td>
<td>link</td>
<td>Test log</td>
<td>weekly</td>
<td>Sun Blade 1000 (US1/1+)</td>
<td></td>
</tr>
</tbody>
</table>

This page is maintained by pooiak@NetBSD.org. Send email if you have regularly updated build/test results you wish to share.
A critique of ATF

• Ugly codebase
• Desired features are hard to implement
  • Parallel execution
  • Integrated dashboards
• High entry barrier for contributors
Refactor vs. Rewrite

• The eternal dilemma
Refactor vs. Rewrite

• The eternal dilemma

• Decision:
  • Rewrite user tools
  • Reuse bindings (atf-{c,c++,sh}) for now
What is Kyua?

- *Think of it as ATF v2*
- Component separation
- “Better” code
- Easier entry for contributors
- New features!
From ATF to Kyua

- Smooth migration plan
- Rewrite the tools (easy!)
- Later: rewrite the libraries (hard)
Kyua: Components

Test program (atf-c)

Test program (atf-c++)

Test program (atf-sh)

Kyua Runtime engine

Kyua Store

Kyua Inspection tools

Reports

Dashboards
Kyua: Easy entry

http://code.google.com/p/kyua/
# Test coverage

<table>
<thead>
<tr>
<th></th>
<th>Test cases</th>
<th>Run time</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATF</td>
<td>542</td>
<td>683</td>
</tr>
<tr>
<td>All components</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kyua</td>
<td>48s real</td>
<td>23s real</td>
</tr>
<tr>
<td>Runtime engine</td>
<td>19s user 4s user</td>
<td>16s sys 5s sys</td>
</tr>
</tbody>
</table>

$ cd /usr/tests

$ kyua test --session_id=bsdcan_session

$ kyua debug fs/tmpfs/t_mount

$ kyua inspect fs/tmpfs/t_mount

$ kyua export --session_id=bsdcan_session -o html
The short-term future

- Kyua 0.1: run-time engine, no store
- Kyua 0.2: with store, exporting of results

- Lua binding for test programs?
- Adoption by other BSDs?
Conclusions

- Tests make NetBSD more robust
- NetBSD has many exciting technologies
- Kyua is the evolution of ATF

- Other BSDs should join the party!
Kyua
http://code.google.com/p/kyua/
<kuya-discuss@googlegroups.com>

Contact
Julio Merino <jmmv@NetBSD.org>