The Future of AGL-JTA
the CIAT tool for AGL

September 8, 2016
AGL Member Meeting FALL

NuoHan Qiao
Fujitsu TEN
Self Introduction

- NuoHan Qiao (qiaonuohan@cn.fujitsu.com)

- Linux Software Engineer (2010 ~ )

- AGL FTE (2015 ~ )
  - engaging on CIAT for AGL
  - especially AGL-JTA
Agenda

- **WHAT** is CIAT
- **WHY** use AGL-JTA
- **HOW** to use AGL-JTA
- Future Work
WHAT is CIAT

Continuous Integration and Automated Test

- [https://wiki.automotivelinux.org/eg-ciat#explanation_of_ciat](https://wiki.automotivelinux.org/eg-ciat#explanation_of_ciat)

CIAT is supposed to include:

- CI pipeline which executes tests on user's demand or triggered automatically
- collection of source code from upstream
- automated instructions for building/deploying built distro
- ability to include binary artifacts
- automated test pipeline which executes sets of tests
- publishing of built distro/component and test results/logs
- mechanism for formal code review prior to merging of changes
- demonstration of license compliance
WHY use AGL-JTA

- based on Fuego
  - formerly called JTA
  - official automated test framework for LTSI project
  - A host/target script engine and a Jenkins front-end

- advantages
  - lots of plugins to extend features
  - highly customizable
  - flexible test configuration
  - running tests in batches
  - not imposing any demands on boards or distributions
  - easy yet flexible board setup
## WHY use AGL-JTA

- **check AGL CIAT’s goals**

<table>
<thead>
<tr>
<th>CIAT’s goals</th>
<th>AGL-JTA’s feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>flexible trigger</td>
<td>build triggers (like gerrit, cron job)</td>
</tr>
<tr>
<td>collection of source code</td>
<td>Source Code Management (SCM, like git)</td>
</tr>
<tr>
<td>instructions for building/deploying</td>
<td>host/target script engine</td>
</tr>
<tr>
<td>include binary artifacts</td>
<td>host/target script engine</td>
</tr>
<tr>
<td>executing sets of tests</td>
<td>job trigger of jenkins</td>
</tr>
<tr>
<td>publishing of distro and test results</td>
<td>plugins of jenkins, or use SCM to upload distro/result</td>
</tr>
<tr>
<td>code review</td>
<td>(offered by gerrit)</td>
</tr>
<tr>
<td>demonstration of license compliance</td>
<td>(offered by gerrit)</td>
</tr>
</tbody>
</table>
HOW to use AGL-JTA

➢ simple test
  • simply execute test on target board

➢ CIAT

➢ share test result
  • upload CIAT’s test result to public server

➢ display test result
  • display shared test result
  • easy to read
HOW to use AGL-JTA – simple test

simple test

- Jenkins will call script engine first, then gather test result/log from script engine
- Script engine will do the work
  - Cross-compile testsuite for target board
  - Load testsuite to target board then execute
  - Gather test logs

execute test script  run test on board

jenkins (frontend)  script engine (backend)

return result  return logs
HOW to use AGL-JTA – simple test

- have a glance – homepage
HOW to use AGL-JTA – simple test

➢ have a glance – execute tests
HOW to use AGL-JTA – simple test

- have a glance – execute tests
HOW to use AGL-JTA – CIAT

> role in AGL’s CIAT
  • for Automated Test mainly
  • also involves building/deploying images

> two instances
  • private server – inhouse CIAT (mid-term test)
  • public server – https://jta.automotivelinux.org/

> work flow
  • trigger
  • collect source code
  • build/deploy distro to target board
  • execute tests on target board
  • publish test result
HOW to use AGL-JTA – CIAT

Private server – inhouse CIAT

1. time trigger

AGL-JTA server

- CIAT.inhouse_mid
  - CIAT.inhouse_deploy
    - CIAT.batch_test
      - CIAT.upload
  - 2. collect source code
  - 3. build & deploy distro
  - 4. execute sets of tests, like Benchmark.bc, Functional.zlib
  - 5. publish test result
HOW to use AGL-JTA – CIAT

- **private server**
  - hardware setting

Diagram:
- Internet (AGL gerrit)
- Hub
- Porter board
- TFTP/NBD server
- AGL-JTA server
- Compile server
HOW to use AGL-JTA – CIAT

➤ public server

1. jenkins trigger

AGL-JTA server

CIAT. mid

CIAT.lava_deploy

2 & 3. call lava to deploy image

CIAT.batch_test

4. execute sets of tests, like Benchmark.bc, Functional.zlib

CIAT.upload

5. publish test result

CIAT.lava_done

6. stop the deployment in lava
HOW to use AGL-JTA – CIAT

- public server
  - lava deploy

AGL-JTA

CIAT.lava_deploy

- test & upload result

CIAT.lava_done

lava

- download image from snapshot server
- power up porter board
- set uboot env to boot porter board with downloaded image
- shutdown porter board
HOW to use AGL-JTA – share test result

- inhouse test as a supplement
- after community’s test, extra test of inhouse server will be executed

```
community’s AGL-JTA

public git repo

AGL-JTA for displaying

FTEN’s AGL-JTA

xxx’s AGL-JTA
```
HOW to use AGL-JTA – share test result

- CIAT.upload
  - one part of CIAT (publishing test result)
  - called by CIAT.mid, CIAT.inhouse_mid
  - test summary, test info, manifest, detailed results

- public git repo for test result
  - git:
    https://gerrit.automotivelinux.org/gerrit/gitweb?p=staging/agl-jta-results.git;a=summary
  - branch for each snapshot
  - 1st-level directories are for company & target board
HOW to use AGL-JTA – share test result

git layout example

```bash
# git branch -a
  remotes/origin/2016-08-17-b350
  remotes/origin/2016-08-18-b351
  remotes/origin/2016-08-19-b352
  remotes/origin/2016-08-24-b357
  remotes/origin/master
# git checkout -b 2016-08-18-b351 origin/2016-08-18-b351
# ls
agl  fnst.porter
# ls -R agl
agl:
detailed_results  info  manifest.xml  result
agl/detailed_results:
  Benchmark.IOzone.xml  CIAT.porter1.xml
# ls -R fnst.porter
fnst.porter:
detailed_results  info  manifest.xml  result
fnst.porter/detailed_results:
  CIAT.porter1.xml  test.porter1.xml
```
HOW to use AGL-JTA – display test result

- make shared test result easy to read and compare

<table>
<thead>
<tr>
<th>No.</th>
<th>Test Name</th>
<th>2016-08-25-b358 @ agl</th>
<th>2016-08-18-b351 @ fnst.porter</th>
<th>2016-08-18-b351 @ agl</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CIAT.common1</td>
<td>10</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>test.common1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>test.porter1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>test.common1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>test.porter1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SUM</td>
<td></td>
<td>13</td>
<td>13</td>
<td>0</td>
</tr>
</tbody>
</table>
# HOW to use AGL-JTA – display test result

- make shared test result easy to read and compare

<table>
<thead>
<tr>
<th>No.</th>
<th>Test Name</th>
<th>2016-08-25-b358 @ agl</th>
<th>2016-08-18-b351 @ fnst.porter</th>
<th>2016-08-18-b351 @ agl</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>DATE: 2016-08-25-b358 (manifest)</td>
<td>DATE: 2016-08-18-b351 (manifest)</td>
<td>DATE: 2016-08-18-b351 (manifest)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Test Dir: /home/a</td>
<td>Test Dir: /home/a</td>
<td>Test Dir: /home/a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Test Device: /dev/sda1</td>
<td>Test Device: /dev/sda1</td>
<td>Test Device: /dev/sda1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Filesystem for Test Device: ext4</td>
<td>Filesystem for Test Device: ext4</td>
<td>Filesystem for Test Device: ext4</td>
</tr>
<tr>
<td>1</td>
<td>Benchmark.IOzone (detail)</td>
<td>total: 10, pass: 10, fail: 0, untest: 0</td>
<td>total: 0, pass: 0, fail: 0, untest: 0</td>
<td>total: 10, pass: 10, fail: 0, untest: 0</td>
</tr>
<tr>
<td>2</td>
<td>CIAT.common1</td>
<td>total: 1, pass: 1, fail: 0, untest: 0</td>
<td>total: 0, pass: 0, fail: 0, untest: 0</td>
<td>total: 1, pass: 1, fail: 0, untest: 0</td>
</tr>
<tr>
<td>3</td>
<td>CIAT.porter1 (detail)</td>
<td>total: 1, pass: 1, fail: 0, untest: 0</td>
<td>total: 0, pass: 0, fail: 0, untest: 0</td>
<td>total: 1, pass: 1, fail: 0, untest: 0</td>
</tr>
<tr>
<td>4</td>
<td>CIAT.porter2</td>
<td>total: 1, pass: 1, fail: 0, untest: 0</td>
<td>total: 0, pass: 0, fail: 0, untest: 0</td>
<td>total: 1, pass: 1, fail: 0, untest: 0</td>
</tr>
<tr>
<td>5</td>
<td>test.common1</td>
<td>total: 0, pass: 0, fail: 0, untest: 0</td>
<td>total: 1, pass: 1, fail: 0, untest: 0</td>
<td>total: 0, pass: 0, fail: 0, untest: 0</td>
</tr>
<tr>
<td>6</td>
<td>test.porter1 (detail)</td>
<td>total: 0, pass: 0, fail: 0, untest: 0</td>
<td>total: 1, pass: 1, fail: 0, untest: 0</td>
<td>total: 0, pass: 0, fail: 0, untest: 0</td>
</tr>
<tr>
<td>7</td>
<td>test.porter2</td>
<td>total: 0, pass: 0, fail: 0, untest: 0</td>
<td>total: 1, pass: 1, fail: 0, untest: 0</td>
<td>total: 0, pass: 0, fail: 0, untest: 0</td>
</tr>
<tr>
<td>SUM</td>
<td></td>
<td>total: 13, pass: 13, fail: 0, untest: 0</td>
<td>total: 3, pass: 3, fail: 0, untest: 0</td>
<td>total: 13, pass: 13, fail: 0, untest: 0</td>
</tr>
</tbody>
</table>
HOW to use AGL-JTA – display test result

make shared test result easy to read and compare

Benchmark IOzone Test Result

- Latest Total: 10
- Latest Pass: 10
- Latest Fail: 0
- Latest Untest: 0

<table>
<thead>
<tr>
<th>No.</th>
<th>Benchmark IOzone</th>
<th>Average</th>
<th>Unit</th>
<th>Criterion</th>
<th>Start time</th>
<th>End time</th>
<th>Board version</th>
<th>Test dir</th>
<th>Test Device</th>
<th>Command line</th>
<th>Result</th>
<th>Output</th>
<th>Unit</th>
<th>Rate (Output/Average)</th>
<th>Result</th>
<th>Output</th>
<th>Unit</th>
<th>Rate (Output/Average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2048 Kb_Record_Write.Random_write</td>
<td>72548.50</td>
<td>KB/s</td>
<td>0.00 – 100.00</td>
<td>2016-08-25 07:26:10</td>
<td>2016-08-25 07:27:46</td>
<td>'Porter Rev 1.0'</td>
<td>/home/a/work</td>
<td>/dev/sda1</td>
<td>ext4</td>
<td>PASS</td>
<td>45608</td>
<td>KB/s</td>
<td>0.63</td>
<td>PASS</td>
<td>46670</td>
<td>KB/s</td>
<td>0.64</td>
</tr>
<tr>
<td>2</td>
<td>2048 Kb_Record_Read.ReRead</td>
<td>111210.60</td>
<td>KB/s</td>
<td>0.00 – 100.00</td>
<td>2016-08-25 07:26:10</td>
<td>2016-08-25 07:27:46</td>
<td>'Porter Rev 1.0'</td>
<td>/home/a/work</td>
<td>/dev/sda1</td>
<td>ext4</td>
<td>PASS</td>
<td>98087</td>
<td>KB/s</td>
<td>0.88</td>
<td>PASS</td>
<td>94671</td>
<td>KB/s</td>
<td>0.85</td>
</tr>
<tr>
<td>3</td>
<td>2048 Kb_Record_Write.ReWrite</td>
<td>71567.80</td>
<td>KB/s</td>
<td>0.00 – 100.00</td>
<td>2016-08-25 07:26:10</td>
<td>2016-08-25 07:27:46</td>
<td>'Porter Rev 1.0'</td>
<td>/home/a/work</td>
<td>/dev/sda1</td>
<td>ext4</td>
<td>PASS</td>
<td>45189</td>
<td>KB/s</td>
<td>0.63</td>
<td>PASS</td>
<td>44664</td>
<td>KB/s</td>
<td>0.63</td>
</tr>
<tr>
<td>4</td>
<td>2048 Kb_Record_Read.ReRead</td>
<td>131265.30</td>
<td>KB/s</td>
<td>0.00 – 100.00</td>
<td>2016-08-25 07:26:10</td>
<td>2016-08-25 07:27:46</td>
<td>'Porter Rev 1.0'</td>
<td>/home/a/work</td>
<td>/dev/sda1</td>
<td>ext4</td>
<td>PASS</td>
<td>94222</td>
<td>KB/s</td>
<td>0.72</td>
<td>PASS</td>
<td>87404</td>
<td>KB/s</td>
<td>0.67</td>
</tr>
<tr>
<td>5</td>
<td>2048 Kb_Record_Write.Write</td>
<td>65798.60</td>
<td>KB/s</td>
<td>0.00 – 100.00</td>
<td>2016-08-25 07:26:10</td>
<td>2016-08-25 07:27:46</td>
<td>'Porter Rev 1.0'</td>
<td>/home/a/work</td>
<td>/dev/sda1</td>
<td>ext4</td>
<td>PASS</td>
<td>44663</td>
<td>KB/s</td>
<td>0.68</td>
<td>PASS</td>
<td>44661</td>
<td>KB/s</td>
<td>0.68</td>
</tr>
<tr>
<td>6</td>
<td>2048 Kb_Record_Read.Read</td>
<td>128735.60</td>
<td>KB/s</td>
<td>0.00 – 100.00</td>
<td>2016-08-25 07:26:10</td>
<td>2016-08-25 07:27:46</td>
<td>'Porter Rev 1.0'</td>
<td>/home/a/work</td>
<td>/dev/sda1</td>
<td>ext4</td>
<td>PASS</td>
<td>90842</td>
<td>KB/s</td>
<td>0.71</td>
<td>PASS</td>
<td>89884</td>
<td>KB/s</td>
<td>0.70</td>
</tr>
<tr>
<td>7</td>
<td>2048 Kb_Record_Write.Write</td>
<td>28588.40</td>
<td>KB/s</td>
<td>0.00 – 100.00</td>
<td>2016-08-25 07:26:10</td>
<td>2016-08-25 07:27:46</td>
<td>'Porter Rev 1.0'</td>
<td>/home/a/work</td>
<td>/dev/sda1</td>
<td>ext4</td>
<td>PASS</td>
<td>36844</td>
<td>KB/s</td>
<td>1.39</td>
<td>PASS</td>
<td>37005</td>
<td>KB/s</td>
<td>1.39</td>
</tr>
<tr>
<td>8</td>
<td>2048 Kb_Record_Read.Random_read</td>
<td>102667.10</td>
<td>KB/s</td>
<td>0.00 – 100.00</td>
<td>2016-08-25 07:26:10</td>
<td>2016-08-25 07:27:46</td>
<td>'Porter Rev 1.0'</td>
<td>/home/a/work</td>
<td>/dev/sda1</td>
<td>ext4</td>
<td>PASS</td>
<td>85245</td>
<td>KB/s</td>
<td>0.83</td>
<td>PASS</td>
<td>80885</td>
<td>KB/s</td>
<td>0.79</td>
</tr>
</tbody>
</table>
HOW to use AGL-JTA

➢ simple test
  ● REAME
  ● docs/jta-docs.pdf
  ● docs/How-to-Add-Test-Cases-on-JTA

➢ CIAT
  ● docs/How-to-Configure-CIAT-on-AGL-JTA.pdf

➢ AGL’s test framework
  ● combination of AGL-JTA and LAVA
  ● https://wiki.automotivelinux.org/agl-testframework
Future Work

➢ board supporting
  • current
    • porter, MinnowBoard, Dragon board (underway)
  • future work
    • improving deployment
    • parallelize the test

➢ trigger
  • current
    • gerrit trigger
    • time trigger
    • manual trigger
  • future work
    • for short/mid/long term
    • for different types of changes uploaded to gerrit
Future Work

- test cases
  - define tests for short/mid/long term tests
  - short -> change/commit
  - mid -> snapshot
  - long -> release

- display
  - current
    - CIAT.display for displaying
  - future work
    - UI, daily mail for test report?
    - similar to meta-isafw report, same solution?

- cooperate with fuego
  - fix gap between fuego and share features and test cases
Future Work

➢ Try AGL-JTA
   - try to setup an instance, use CIAT.inhouse_mid to do inhouse test for snapshot

➢ share test
   - add tests to AGL-JTA
   - test extra cases and share the test result

➢ participate in improving AGL-JTA and AGL’s CIAT
Thank you!

qiaonuohanan@cn.fujitsu.com