Agenda

- Continuous Integration Status
- Proposal for CI improvement
- CI in Open Lab
- Open Lab CI dashboard
DPDK CI introduction

dpdk.org
(Patchwork
Repositories)

Build

Server

Function test

Server

Performance test

CI control center
(Jenkins)

1. send patch set to
2. pull
3. trigger tasks
4. report
5. report
6. report
DPDK CI current status

DPDK CI has two parts:
Before merge: Process of new patch set which haven’t merge into any tree
After merge: Process of main-tree (master)

<table>
<thead>
<tr>
<th></th>
<th>Before merge</th>
<th>After merge</th>
</tr>
</thead>
<tbody>
<tr>
<td>build</td>
<td>✔️ (Intel)</td>
<td>✔️ (Intel)</td>
</tr>
<tr>
<td>Unit test</td>
<td>✗</td>
<td>✔️ (Intel)</td>
</tr>
<tr>
<td>Function test</td>
<td>✗</td>
<td>✔️ (Intel)</td>
</tr>
<tr>
<td>Performance test</td>
<td>✔️ (Open lab)</td>
<td>✗</td>
</tr>
</tbody>
</table>

✔️ Implemented
✗ Gap
Proposal for CI improvement

Gaps

Before merge
- Need identify patch set to go to which tree
- Need cover unit test and function test
- Need more performance test

After merge
- Need cover performance test

Proposal
- Two options on tree-selected issue.
- Step by step cover the test gaps
- Before merge > after merge
Workflow - Before merge

Stage 1
- Apply
  - Pull main-tree *(master)*
  - Download patch set
  - Apply patchset

Stage 2
- Build
  - Centos/gcc/x86_64
  - Ubuntu/clang/x86_32
  - OS*/Compiler*/arch*

Stage 3
- Test
  - Unit test
  - Function test
  - Performance test

Whenever error or final pass

Report to dpdk.org

End
Workflow - Before merge

Start

Apply
- Pull main tree (master)
- Download patch set
- Apply patchset

Build
- Centos/gcc/x86_64
- Ubuntu/clang/x86_32
- OS*/Compiler*/arch*

Test
- Unit test
- Function test
- Performance test

Whenever error or final pass

Report to dpdk.org

End

Stage 1
Stage 2
Stage 3

......

......

......
Open: Which tree patch set should go?

Options
1. Go to next-master
2. Go to exact sub-tree
Option 1: Go to next-master

1, apply to master
- patch set
- master
- apply
- pass
- next: build/test

2, apply to next-master
- next-master
- apply
- pass
- conflict

Report to dpdk.org
- conflict
- pass
- next-master: Local tree which formed by merging all sub-trees into master
Option 2: Go to exact sub-tree

- Information in MAINTAINERS file
- Script to analyze the git history and gives a percentage of probability.
## Option comparison

<table>
<thead>
<tr>
<th>Go to</th>
<th>next-master (Preferred)</th>
<th>exact sub-tree</th>
</tr>
</thead>
</table>
| **Pros ✔** | 1. No dependency  
2. No extra script  
3. Already have weekly merge: sub-trees to master  
4. Early to find issues | 1. Timely reports |
| **Cons ✗** | 1. Will be delayed if merge conflict  
2. Maintainers need to manually fix conflict | 1. Extra script effort  
2. Still uncertainty |
Workflow – After merge

Stage 1
Download
Download main-tree (master)

Stage 2
Build
Centos/gcc/x86_64
Ubuntu/clang/x86_32
OS*/compiler*/arch*

Stage 3
Test
Unit test
Function test
Performance test

Whenever error or final pass
Report to dpdk.org
End
# DPDK CI in Open Lab

<table>
<thead>
<tr>
<th></th>
<th>Before merge</th>
<th>After merge</th>
</tr>
</thead>
<tbody>
<tr>
<td>build</td>
<td>✓ (Intel)</td>
<td>✓ (Intel)</td>
</tr>
<tr>
<td>Unit test</td>
<td>Open lab</td>
<td>✓ (Intel)</td>
</tr>
<tr>
<td>Function test</td>
<td>Open lab</td>
<td>✓ (Intel)</td>
</tr>
<tr>
<td>Performance test</td>
<td>✓ (Open lab)</td>
<td>Open lab</td>
</tr>
</tbody>
</table>

## Future plan (before merge > after merge)

1. CI before merge implementation deploy (option 1)
2. More basic & critical performance test
3. Expend to function test & unit test
Open Lab CI dashboard

• What is it?

• Who is it hosted by?

• What are the plans for the future?

• How to get involved?
What is it?

https://lab.dpdk.org/

Below is the current CI performance testing status for the patches currently active in the DPDK Patchwork instance. Possible statuses are Pending, Waiting, Apply Error, Incomplete, Possible Regression, and Pass. NIC vendors may log in to view detailed performance results for their hardware.

<table>
<thead>
<tr>
<th>ID</th>
<th>Ver</th>
<th>Title</th>
<th>Submitter</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>43792</td>
<td>2</td>
<td>[v2] port: add sym crypto port</td>
<td>Zhang, Roy Fan</td>
<td>Pass</td>
</tr>
<tr>
<td>43791</td>
<td>1</td>
<td>[RFC] mkix: flow counters support on the linux-rdma v19 base</td>
<td>vlacheslavio</td>
<td>Pass</td>
</tr>
<tr>
<td>43790</td>
<td>1</td>
<td>port: add sym crypto port</td>
<td>Zhang, Roy Fan</td>
<td>Pass</td>
</tr>
<tr>
<td>43789</td>
<td>1</td>
<td>[RFC] ethdev: flow counters batch query</td>
<td>vlacheslavio</td>
<td>Pass</td>
</tr>
<tr>
<td>43788</td>
<td>1</td>
<td>bus/ismc: fix the undefined ref of rte_dpaas2 memsegs</td>
<td>Hemant Agrawal</td>
<td>Pass</td>
</tr>
<tr>
<td>43787</td>
<td>2</td>
<td>[v2] netbonding: fix but corruption in merging un-transmitted packets</td>
<td>Jia Yu</td>
<td>Pass</td>
</tr>
<tr>
<td>43786</td>
<td>1</td>
<td>netbonding: fix but corruption in merging un-transmitted packets</td>
<td>Jia Yu</td>
<td>Pass</td>
</tr>
<tr>
<td>43785</td>
<td>1</td>
<td>net/octeonix: fix packet corruption on Tx</td>
<td>Jerin Jacob</td>
<td>Possible Regression</td>
</tr>
<tr>
<td>43784</td>
<td>1</td>
<td>vhost: initializing ioctl memory only when IOMMU feature is enabled.</td>
<td>Nilesh Katiyar</td>
<td>Pass</td>
</tr>
<tr>
<td>43783</td>
<td>1</td>
<td>app/testpmd: add commands for TM to mark pfts</td>
<td>Krzysztof Kanas</td>
<td>Pass</td>
</tr>
<tr>
<td>43782</td>
<td>1</td>
<td>vhost: fix crash if set vring num handling failed</td>
<td>Ilya Maximets</td>
<td>Possible Regression</td>
</tr>
<tr>
<td>43777–43781</td>
<td></td>
<td>Enable hotplug in vfio</td>
<td>Jeff Guo</td>
<td>Apply Error</td>
</tr>
</tbody>
</table>
Detailed View

- View relative performance numbers
- View hardware configuration information
- Download artifacts for logged in members
Who is it hosted by?

- University of New Hampshire InterOperability Laboratory (UNH IOL)
- Provides a neutral environment for members
- Established in 1988 with over 100 currently participating companies
- Current Open Lab Participants: Intel, Mellanox, NXP
What are the plans for the future?

- Running different test cases
- User management
- Graphing Integration
- Continuous Integration
Getting Involved

Dashboard: https://lab.dpdk.org/

Issue reporting: https://bugs.dpdk.org/

CI Mailing List: http://mails.dpdk.org/listinfo/ci

Submitting Equipment: https://www.iol.unh.edu/testing/hpc/dpdk

Contact: dpdklab@iol.unh.edu
Questions?

Lijuan Tu
lijuan.tu@intel.com

Jeremy Plsek
jplsek@iol.unh.edu