

VSPERF CI

Table of contents:

- [Introduction](#)
- [CI JOBS](#)
- [Where do VSPERF CI jobs run?](#)
- [FAQ](#)
 - [Q: Why VEFIY JOB has failed and my patch got -1 for verification?](#)
 - [Q: Why DAILY JOB has failed?](#)
 - [Q: Why DAILY JOB takes too long to finish \(e.g. 1 or 2 days\)?](#)
 - [Q: What to do if Jenkins slave appears to be offline?](#)
 - [Q: What to do if IxNework TCL Server is Down/Connection Failed ?](#)
- [Ideas](#)
 - [Configure 2nd jenkins slave for execution of VSPERF JOBS.](#)
 - [Reconfigure VERIFY & MERGE jobs to utilize ericsson PODs more often.](#)
 - [Minimize impact of jenkins health check application](#)

Introduction

VSPERF CI consists of several jobs, which are integrated into OPNFV infrastructure. It means that jobs are triggered by OPNFV jenkins (daily job) or OPNFV gerrit (verify and merge jobs). The comprehensive list of jobs, their status and history is visible in VSPERF specific dashboard at <https://build.opnfv.org/ci/view/vswitchperf/>

There are two versions of each job, one is created for current stable branch and second for the master branch.

In case of the daily job, which executes a set of performance tests, the results are available also in the graphical form at [VSPERF CI Results](#) and test results, reports and logs are stored inside OPNFV artifacts at http://artifacts.opnfv.org/logs_vswitchperf_intel-pod12.html.

OPNFV Jenkins is operated by releng team and the configuration of jobs is stored in releng git repository. VSPERF specific part can be found at YAML file [vswitchperf.yml](#). For more info on writing and using jjbs see [Jenkins Wow](#).

In order to have more flexible way of job configuration, VSPERF project stored detailed job configuration in VSPERF repository into [build-vsper.sh](#) script, which is invoked by generic YAML job configuration above.

Links summary:

CI Dashboard: <https://build.opnfv.org/ci/view/vswitchperf/>

Daily job results:

- VSPERF output: check "console output" of selected job type at jenkins, e.g. for a daily job <https://build.opnfv.org/ci/view/vswitchperf/job/vswitchperf-daily-master/lastSuccessfulBuild/consoleFull>
- report, results & logs: http://artifacts.opnfv.org/logs_vswitchperf_intel-pod12.html
- graphs: [VSPERF CI Results](#)

Job definition scripts:

- generic YAML (releng git repo): [vswitchperf.yml](#)
- job details (vsperf repo): [build-vsper.sh](#)

CI JOBS

The VSPERF CI jobs are broken down into:

1. Daily job:
 - a. It is executed at INTEL POD.
 - b. Requires a traffic generator (Ixia)
 - c. Runs everyday **in case that new change was merged into particular branch since the last daily job execution**; Daily job duration is about 14 hours, but it can take over a day in case that VM running IxNetwork is slow. Please see FAQ section below for details.
 - d. A set of performance tests is executed for OVS with DPDK support, Vanilla OVS, VPP and SRIOV. Ixia traffic generator is used to generate RFC2544 Throughput and Back2Back traffic.
2. Merge job (similar to verify job):
 - a. It is executed at INTEL POD or at Ericsson PODs.
 - b. Does not require a traffic generator.
 - c. Runs whenever patches are merged to the particular branch.
 - d. Runs a basic set of integration testcases for OVS with DPDK support, Vanilla OVS and VPP.
 - e. in case that documentation files were modified, then documentation is built.
3. Verify job (similar to merge job):
 - a. It is executed at INTEL POD or at Ericsson PODs.
 - b. Does not require a traffic generator.
 - c. Runs every time a patch is pushed to gerrit. On success, the patch will be marked as verified (+1 for verification).
 - d. Runs a basic set of integration testcases for OVS with DPDK support, Vanilla OVS and VPP.

- e. in case that documentation files were modified, then documentation is built

NOTE: The list of testcases to be executed for particular job type is configured inside [build-vsperf.sh](#). Please refer to configuration options TESTCASES_* and TESTPARAM_* for additional details.

Where do VSPERF CI jobs run?

VSPERF project has a dedicated POD hosted at Intel LAB. Please check [Intel POD12](#) and [VSPERF in Intel Pharos Lab - Pod 12](#) for details.

DAILY JOB:

It requires a traffic generator in order to execute the performance testcases. Thus this job is executed at POD12.

The status of Intel POD12 is visible in jenkins at: <https://build.opnfv.org/ci/computer/intel-pod12/>

VERIFY and MERGE JOB:

They are executed at POD12 or at Ericsson pods as they don't require a traffic generator. POD12 is used as a primary jenkins slave, because execution at Ericsson build machines was not reliable since other projects start to use it more extensively. It seems, that there is a clash on resources (hugepages). There was a attempt to avoid a parallel execution of VSPERF and other jobs, but it didn't help. Contact for the Ericsson Pod: _____

FAQ

Q: Why VEFIY JOB has failed and my patch got -1 for verification?

A: Please check "console output" of failed job to find out a cause of failure. Them most common failures are:

1. DPDK, OVS, QEMU or VPP can't be cloned from it's repository and thus job fails. Example of console output in that case:

```
Cloning into 'dpdk'...
error: RPC failed; result=18, HTTP code = 200
fatal: The remote end hung up unexpectedly
fatal: early EOF
fatal: index-pack failed
make[1]: *** [dpdk] Error 128
```

This is often a temporary case and it is enough to re-trigger the job, e.g. by **inserting a comment "reverify" into gerrit review** in question. If problem will persist, please get in touch with admins responsible for particular server to verify, that connection to the failing site is not blocked by firewall.

2. **PYLINT execution failed**. Please note, that all files have to pass with score "10" from pylint. Please check a console output for pylint verification details. Correct values are "OK" (i.e. score 10), "NA" (not a pylint code, e.g. a configuration file) or EXCLUDED (e.g. python 2.7. library). In case of pylint error, you will see a score (e.g. 9.64) and a list of detected pylint errors. It is essential to **use the same version of pylint at your server**. This is ensured by installation of vsperf requirements into your virtual environment by vsperf installation scripts or by execution of "pip install -r requirements.txt" from vsperf repository when your vsperf python virtual environment is active.
3. **Jenkins slave went offline** during job execution. Example of a console output in that case:

```
FATAL: command execution failed
java.nio.channels.ClosedChannelException
    at org.jenkinsci.remoting.protocol.NetworkLayer.onRecvClosed(NetworkLayer.java:154)
    at org.jenkinsci.remoting.protocol.impl.NIONetworkLayer.ready(NIONetworkLayer.java:179)
    at org.jenkinsci.remoting.protocol.IOHub$OnReady.run(IOHub.java:721)
    at jenkins.util.ContextResettingExecutorService$1.run(ContextResettingExecutorService.java:28)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
Caused: java.io.IOException: Backing channel 'JNLP4-connect connection from 10.30.0.3/10.30.0.3:34322'
is disconnected.
...
```

There are two common causes:

- a. A connection between Jenkins server and jenkins slave (a server where tests were physically executed) was terminated. Please **check Jenkins GUI to find out the status of server** in question, e.g. at <https://build.opnfv.org/ci/computer/>. If it is "offline" then it is always good to check a status of other servers in the same lab. For example in case that Intel POD12 is offline and other intel PODxx are offline too, then it is some generic connection issue between Intel and OPNFV LAB. If such problem persists, then you should raise a INFRA ticket (if it is not there already). On the other hand if server is online ("idle"), then it was a temporary problem and job can be re-triggered again. In case that particular jenkins slave is offline for a long duration, then contact responsible administrator (in case of Intel-POD12 vsperf community can reboot it or access a console over web) for help.

- b. A server was rebooted. It means, that shortly (up to 10 minutes) after the job failure the server is up and running (a status "idle" is visible in jenkins GUI). In that case, re-trigger the job. In case that job fails again with another reboot, then go ahead with inspection of "console output". In the past a reboots were observed at Intel-POD12 during execution of OVS Vanilla testcases. If this is the case, then please login at Intel-POD12 and **check what kernel version is running**. In case that CentOS specific kernel 3.10.xxx is active (uname -a), then you should update grub to **use kernel 4.4**, installed from epel repo. As of now (Apr 2018), recent versions of OVS and especially openvswitch.ko kernel module are having issues with recent modifications of 3.10.xxx kernels modified by RHT. Thus kernel 4.4. is used by default at Intel-POD12, however in case of OS update (yum update), the default kernel can be updated and selected by default by GRUB. It is often enough to update grub config to use kernel 4.4. by default and reboot the server. In case that regular reboots are observed at other PODs (e.g. ericsson), then you should get in touch with responsible admins. Hint: In case that default version of tool causing reboot (e.g. OVS) was changed, then you could try to push a (temporary?) patch to gerrit with older or newer version of tools to find out the version, which is "compatible" with OS at given jenkins slave. This information will be helpful during debugging and discussion with responsible administrators.

Q: Why DAILY JOB has failed?

A: Please firstly check answer to "VERIFY JOB has failed" above for causes common for all jobs. Please note then in case of DAILY job, INTEL POD12 is used as a jenkins slave (a job executor) and VSPERF community does OS administration of this server themselves. So you can login and investigate issues directly. In case of daily job, it is possible to re-trigger it from jenkins GUI, but only in case that jenkins user is logged in and he has appropriate privileges. Get in touch with VSPERF PTL and Linux Fundation helpdesk in order to get these privileges. In case that generic issues above didn't occur, then following DAILY job specific issues can occur:

1. If all testcases listed in console output have status "FAILED", then it is possible that Ixia traffic generator is causing issues. This can be detected in "console output". In case that all TCs (OVS with DPDK, Vanilla OVS and VPP) have FAILED, then you should look for part of Ixia execution starting with log entry "Connecting to IxNetwork machine...". If there are messages like "Can't connect to socket", "couldn't open socket: host is unreachable" or "Failed to parse output", then you should check following:
 - a. that **VM where IxNetwork GUI is executed is up & running** (try remote desktop connection to 10.10.120.6). If it is not reachable, then you should file a jira ticket to INFRA
 - b. after logging into VM (above) as "vsperf_ci" user, please **check that IxNetwork is up & running**; If not (e.g. due to VM reboot) please execute it by appropriate icon and verify that it runs at correct port (see [Intel POD12](#) page for port assignment. In case of CI DAILY job it should be port 9126)
 - c. after execution of IxNetwork (above), please **check connection to IXIA chassis** (as of now 10.10.50.6 is possible); If not, please file a jira ticket to INFRA. Note: if don't know, how to check that chassis is up, then you could try to execute a vsperf TC (e.g. manually from Intel POD3) and check "ports" after TC will finish (i.e. fail). In case that ports can't be reserved (they are red) you could inspect their details. In case that you'll see unsuccessful connections to chassis (with many retries), then chassis is not reachable from IxNetwork GUI app (i.e. chassis down or connection between VM and chassis is broken).
 - d. after connection to chassis is verified and OK (above), you should **check IxNetwork GUI for any error indication**. For example after TC execution, there can be "license" related errors listed in Ixia log file (visible in GUI) or errors and warnings indications (see icons at IxNetwork GUI status bar). In case of license related errors (they expires every 6 months), please file a jira ticket to INFRA.
 - e. if all above is correct but "Failed to parse output" is still observed, then **shared samba folder** between VM and node3 should be verified. Please check vsperf documentation for details about shared folder between Win machine running IxNetwork GUI and DUT. Folder should be properly mounted at node3 and its content must be readable by the jenkins user account.
2. if only a few testcases have status FAILED (e.g. pvpv_tpvt or pvpv_back2back), then it can be caused by TC timeout. It has been observed several times, that server used for daily job execution gets slower during execution of daily job. In case that the same TC is executed manually it mostly works correctly, but execution of a lot of TCs at once can slow down the machine. It is not clear, what is the root cause. It can be caused by the lack of resources (please check, that enough hugepages are allocated, but at the same time enough memory is available at both nodes to execute processes). There is also a suspicion, that java application, which is used as node health check by OPNFV Jenkins (slave-agent.jnlp) is consuming a lot of resources and causing a machine slowness. The same application was executed at Intel-POD3 without any issues. However at Intel-POD12, it sometimes runs many threads and consumes significant amount of memory. It might be caused by some connection issues between OPNFV and Intel lab. In case that, that it happens, please use "monit" to restart "jenkins" servis or "systemctl" to restart "monit".

Q: Why DAILY JOB takes too long to finish (e.g. 1 or 2 days)?

A: This is caused by VM where IxNetwork GUI application is executed. In the past, VSPERF used Intel-POD3, where execution of DAILY job was stable. It means, that performance results were stable among Daily job executions and the execution always took about 12 hours. After the move to a different Intel LAB and to Intel-POD12, the performance started to fluctuate and daily job execution takes more time by each execution. Several attempts to fix these issues were made, but issues still persists. In order to shorten DAILY job execution, it is required to login into VM as "vsperf_ci" user via remote desktop and to **restart IxNetwork GUI application**.

Q: What to do if Jenkins slave appears to be offline?

A: Check if Jenkins slave process is running:

```
[root@pod12-node3 ~]# ps -ef | grep jenkins
jenkins    12995      1  0 Feb13 ?        00:09:40 java -jar slave.jar -jnlpUrl http://build.opnfv.org/ci/computer/intel-pod12/slave-agent.jnlp -secret <secret> -noCertificateCheck
root       17681  17647  0 15:23 pts/0    00:00:00 grep --color=auto jenk
```

You can also restart it if needed using "monit stop" and "monit start" commands. Example output:

```
[root@pod12-node3 ~]# monit
status
```

```
Monit 5.25.1 uptime: 73d 5h  
29m
```

```
Directory 'jenkins_piddir'  
  status          OK  
  monitoring status Monitored  
  monitoring mode active  
  on reboot       start  
  permission      755  
  uid             1001  
  gid             1001  
  access timestamp Mon, 03 Dec 2018 09:54:12  
  change timestamp Wed, 13 Feb 2019 14:35:01  
  modify timestamp Wed, 13 Feb 2019 14:35:01  
  data collected   Thu, 14 Feb 2019 15:23:51  
  
Process 'jenkins'  
  status          OK  
  monitoring status Monitored  
  monitoring mode active  
  on reboot       start  
  pid             12995  
  parent pid      1  
  uid             1001  
  effective uid   1001  
  gid             1001  
  uptime          1d 0h 48m  
  threads         53  
  children        0  
  cpu              0.0%  
  cpu total       0.0%  
  memory          0.7% [443.8 MB]  
  memory total    0.7% [443.8 MB]  
  security attribute (null)  
  disk read       0 B/s [81.8 MB total]  
  disk write      0 B/s [6.8 GB total]  
  data collected   Thu, 14 Feb 2019 15:23:51  
  
System 'pod12-node3.opnfv.local'  
  status          OK  
  monitoring status Monitored  
  monitoring mode active  
  on reboot       start  
  load average    [0.00] [0.00] [0.00]  
  cpu              0.0%us 0.0%sy 0.0%wa  
  memory usage    15.2 GB [24.1%]  
  swap usage      0 B [0.0%]  
  uptime          73d 5h 30m  
  boot time       Mon, 03 Dec 2018 09:53:25  
  data collected   Thu, 14 Feb 2019 15:23:51
```

Q: What to do if IxNework TCL Server is Down/Connection Failed ?

A: Currently there are 3 vsperf user accounts for IxNetwork in Ixia VM. Follow the below procedure to overcome the issue. Basically, all IxNetwork port numbers are pre-configured. You would just need to restart it.

1. Connect the Ixia VM (Remote Desktop) using 'vsperf_ci' login and password. Once its connected and VM is launched, system should automatically start IxNetwork service on TCL port 9126. Open the Hidden icon's arrow button in task bar and place the mouse pointer on the IxNetwork icon to see whether it shows the TCL Port Configuration. If it's not started automatically, then double click on the IxNetwork icon and it will start the service at port 9126.

2. Connect the Ixia VM (Remote Desktop) using 'vsperf_sandbox' login and password. Once its connected and VM is launched, system should automatically start IxNetwork service ion TCL port 9127. Open the Hidden icon's arrow button in task bar and place the mouse pointer on the IxNetwork icon to see whether it shows the TCL Port Configuration. If it's not started automatically, then double click on the IxNetwork icon and it will start the service at port 9127.

3. Connect the Ixia VM (Remote Desktop) using 'vsperf_sandbox2' login and password. Once its connected and VM is launched, system should automatically start IxNetwork service ion TCL port 9128. Open the Hidden icon's arrow button in task bar and place the mouse pointer on the IxNetwork icon to see whether it shows the TCL Port Configuration. If it's not started automatically, then double click on the IxNetwork icon and it will start the service at port 9128.

If above three IxNetwork TCL services are running fine, then you are good to go.

Ideas

Configure 2nd jenkins slave for execution of VSPERF JOBS.

There are several nodes available at Intel-POD12 (see [Intel POD12](#)). Currently there are two sandboxes, where second sandbox using node1 and node2 was created recently. It would be possible to reconfigure 2nd sandbox to be used as another (or even two) jenkins slave. This would speed up execution of VSPERF Jobs. However releng team must be consulted regarding the proper naming as two different jenkins slaves will be hosted at the same Intel POD12.

Reconfigure VERIFY & MERGE jobs to utilize ericsson PODs more often.

In the past, VERIFY & MERGE jobs, were executed at [opnfv-build-ubuntu](#) groups of slaves, which consists of ericsson-build3 and ericsson-build4 machine. The execution was reliable at both of these servers for several months, but later it started to fail. There were several issues, some of them were related to hugepages allocations and usage and to VPP. In case of VPP, it happened several times, that it stopped to work at all at one of ericsson servers. Responsible admins were asked for help, but they were not able to find a root cause. The only solution was to reboot affected server and it worked for some time again. There is a suspicious, that both hugepages and VPP issues are caused by parallel execution of jobs for vsperf and other projects. As debugging of such race condition at server without any access is hardly possible, both VERIFY & MERGE jobs are primarily executed at Intel-POD12. Idea was to execute VERIFY & MERGE jobs at POD12 if it is not occupied by DAILY job and if so, then to move to ericsson POD. However current YAML file definition doesn't work that way. It switches to ericsson POD only in case that INTEL POD12 is offline. Releng engineers can help us with YAML file definition to achieve better utilization of available PODs.

Minimize impact of jenkins health check application

Consider pinning of jenkins health check application at second numa slot, which is not used for performance tests execution. Even better would be a move of that application to a jumphost. However one had to solve, how to execute vsperf "remotely" and how configure multiple slaves at the same pod (probably not possible to run multiple healthchecks at the same machine - may be container would help).

This won't be needed if we will configure more jenkins slave at Intel POD12.