PLUGFEST REPORT

Results and Lessons from the First OPNFV Plugfest (May 2016)

A Linux Foundation Collaborative Project

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The Open Platform for Network Functions Virtualization (OPNFV) community was founded on September 30, 2014. This Linux Foundation collaborative project provides a carrier-grade, integrated, open source platform to accelerate the introduction of new NFV products and services.

The inaugural OPNFV Plugfest—hosted at CableLabs' facility in Louisville, Colorado, USA—focused on the deployment and integration of OPNFV over multiple hardware platforms, as well as deployment of Virtual Network Function (VNF) applications. The results indicated that the industry is taking major steps forward in the realization of NFV.

A key goal of the Plugfest was to fortify the unique capabilities of OPNFV's functional and performance testing projects and infrastructure across new use cases from OPNFV members and the NFV ecosystem. This goal and many others were achieved, cementing OPNFV’s role as the key testing and integration platform for NFV.

Forty-one participants from nineteen organizations—including three non-members—attended from around the world. They brought their key NFV use cases and the technologies needed to bring them to fruition.

The Plugfest tested functionality, and captured results accordingly. The Functest (functional testing) and Yardstick (system measurement) projects were heavily represented, along with the StorPerf (storage performance) and CPerf (controller performance) projects. In particular, the CPerf project is doing extensive SDN controller integration across all OPNFV installers.

Additionally, all of the installer groups (Apex, JOID, Fuel and Compass) were represented and participants used these installers on different platforms. Onsite hardware was provided by Huawei and Intel while Pharos labs were accessed remotely at ENEA Software in Sweden, Ericsson in Texas, and other locations.

The Plugfest focused on testing the OPNFV NFV infrastructure (NFVI) as well as Virtual Network Functions (VNFs) such as virtual Customer Premises Equipment (vCPE), virtual IP Multimedia Subsystem (vIMS) and virtual Evolved Packet Core (vEPC) that participants brought to advance the viability of NFV applications. Examples include:

- Deploying existing scenarios on new hardware
- Interoperability of new internal or external network elements
- Basic VNF lifecycle management
- NFV use cases or feature capabilities

Hardware, software and system vendors, partnering with service providers from many geographies, collaboratively used the local and remote Pharos resources to verify installation and testing with all the OPNFV installers from Brahmaputra. Service providers added new interworking tests (between OPNFV and new platforms and products) to their portfolios. All of the testing information is available to the participants for further analysis.

As an example, a system vendor verified new installers as well as an additional storage backend. A hardware vendor also tested a new installer with a number of scenarios, and another hardware vendor performed testing (using Functest and Yardstick) as well as integration with an installer (Apex) that had not been verified prior to the Plugfest.

In addition to all the real-time interoperability and testing of NFV infrastructure and applications, the Plugfest included several well-attended and valuable breakout sessions on OpenDaylight, the Functest and Yardstick testing projects, and a session on the entire OPNFV testing ecosystem. Many of the lessons from these sessions were applied in the lab.

INTRODUCTION

Multiple teams were formed in preparation for the Plugfest, and conference calls were regularly held to plan the details of the event. OPNFV made collaboration tools such as a private Wiki available prior to the event. Participants included software vendors building VNFs, and hardware or open source system vendors working on NFVI.

At the start of the Plugfest, all were using their preferred installers, and focused mainly on familiarizing themselves with, and setting up, the major testing projects: Functest, Yardstick, and StorPerf. Overview sessions for these testing projects were held at the Plugfest, as well as a multi-project session focusing on the overall approach that OPNFV takes to testing.

As the Plugfest progressed, new installers were explored, as well as alternatives on the NFVI—for instance, a freeware storage system was explored as an alternative to Ceph. VNF systems (such as vCPE, vIMS, and vEPC) were set up and connected to Functest for rudimentary functional testing. In addition, an NFVI monitoring and analytics engine was also verified on all of the hardware.

In order to make their solutions more general, several software vendors tested new combinations of installers with hardware. An embedded system vendor helped integrate these VNFs with hardware encryption. More generally, ways to automate all live VNF testing with OPNFV were explored.

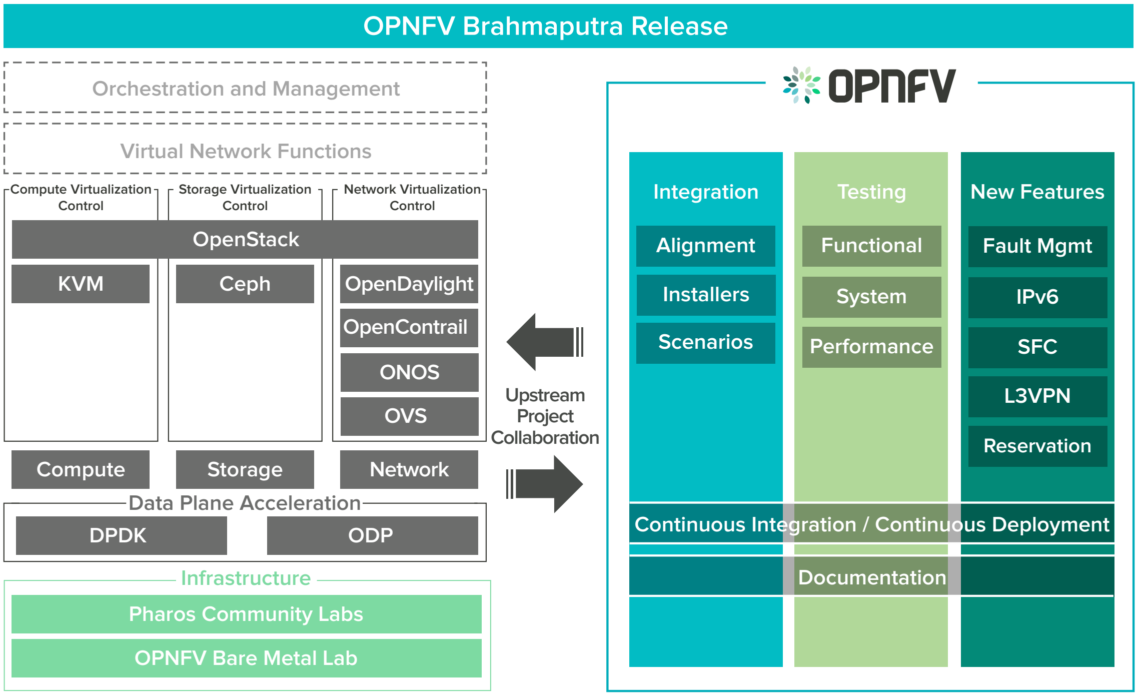
The projects that worked on the relevant installers (Apex, Fuel, JOID, Compass) all developed plans for upstream patches that will be required for the OPNFV Colorado release. Many of these were geared for OpenStack. Another major upstream open source project, the OpenDaylight SDN controller platform, was tested throughout the week with multiple installer projects to fine tune new scenarios.

ABOUT OPNFV SCENARIOS

The following diagram shows an architectural view of the OPNFV Brahmaputra release. The left side of the diagram highlights upstream components in Brahmaputra along with the community lab infrastructure, where users can test the platform in different environments and on different hardware.

The right side of the diagram shows representative capabilities in the areas of integration, testing, and adding features to services and applications. These capabilities became part of OPNFV via approved OPNFV projects.

Figure 1: An Architectural View of OPNFV (as of Brahmaputra)



OPNFV includes pre-built deployment scenarios that include SDN controllers, installers, deployment options, and carrier-grade features. Users can compose their own scenarios by selecting and configuring features and components relevant to a service, taking advantage of a strong OPNFV continuous integration pipeline.

Scope of the Testing

There were several key areas of testing for the Plugfest:

- ***OPNFV Deployment***: : Integration with different hardware platforms and installers (Apex, Compass, Fuel, JOID). Goals included new combinations of installers with NFV infrastructure.
- ***Network Integration and Testing***: Integration of OPNFV platform with SDN Controller or other networking elements (physical or virtual). Goals included the integration of functional and performance tests with NFVI and VIM components.
- ***VNF Applications***: Verification of application life cycle on OPNFV platform(s), including deployments created by different installers. Goals included new installation, deployment and testing methods applied for new VNFs such as vEPC and vIMS.
- ***End to End (E2E) Interoperability***: E2E integration test of reference points on NFVI and MANO.

Note: For an overview of ETSI's architectural framework, see https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf

Some extended scenarios were tested that included the interoperability of products from different vendors; these included OPNFV implementations, network elements, SDN controllers, and VNF applications.

LAB SETUP

OPNFV test infrastructure was provided on-site at CableLabs and through Pharos Community Labs hosted off-site.

The Pharos Project provides the OPNFV community with a lab infrastructure that is geographically and technically diverse. The Pharos Specification defines a compliant deployment and test environment known as a Pharos Pod.

The on-site hardware, provided by Huawei and Intel, is detailed in the following sections.

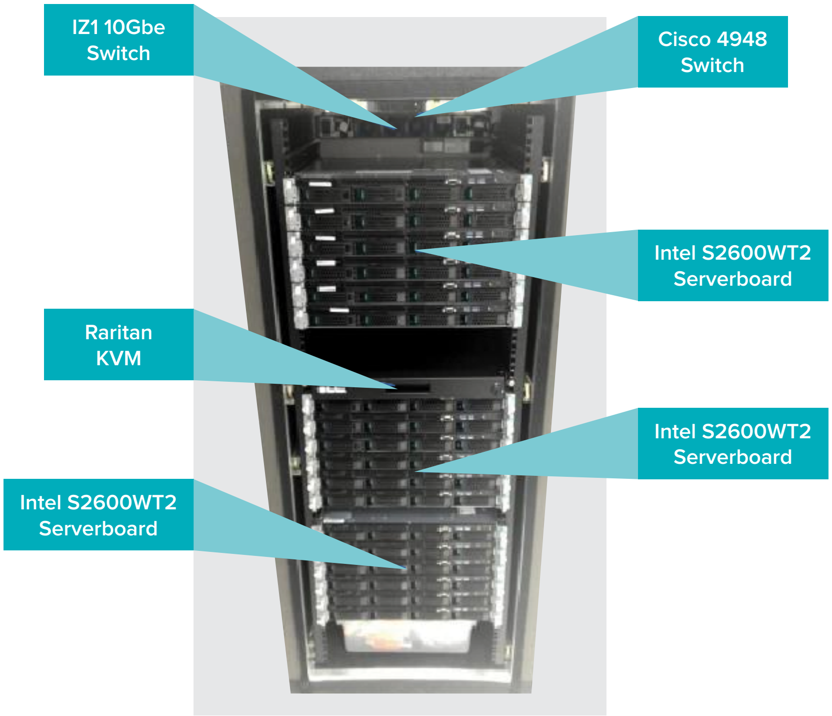
Intel On-Site Pharos Pods

Intel provided three complete Pharos pods on-site at CableLabs. Each pod consisted of six Intel® Xeon® based “white-box” rack-mount servers with the following configuration:

- Intel® Server Board “Wildcat Pass” 1U R1208WT2GSR
- Intel® Xeon® E5-2699 v4 LGA2011 2.2GHz 55MB 145W 22 cores (formerly known as Broadwell-EP) - two processors per server
- 64GB RAM (8x 8GB 2133 Reg ECC 1.2V DDR4 Kingston KVR21R15S4/8 Single Rank
- 480GB SSD 2.5in SATA 3.0 6Gb/s Intel® “Haleyville”
- 3TB SATA Hard Disk Drive
- Intel® 10 Gigabit Server Adapters
  - Intel® Ethernet Converged Network Adapter X710-DA2
  - Intel® Ethernet Converged Network Adapter X520-DA2

The rack containing the 3 Pharos pods (18 servers) is shown in Figure 2.

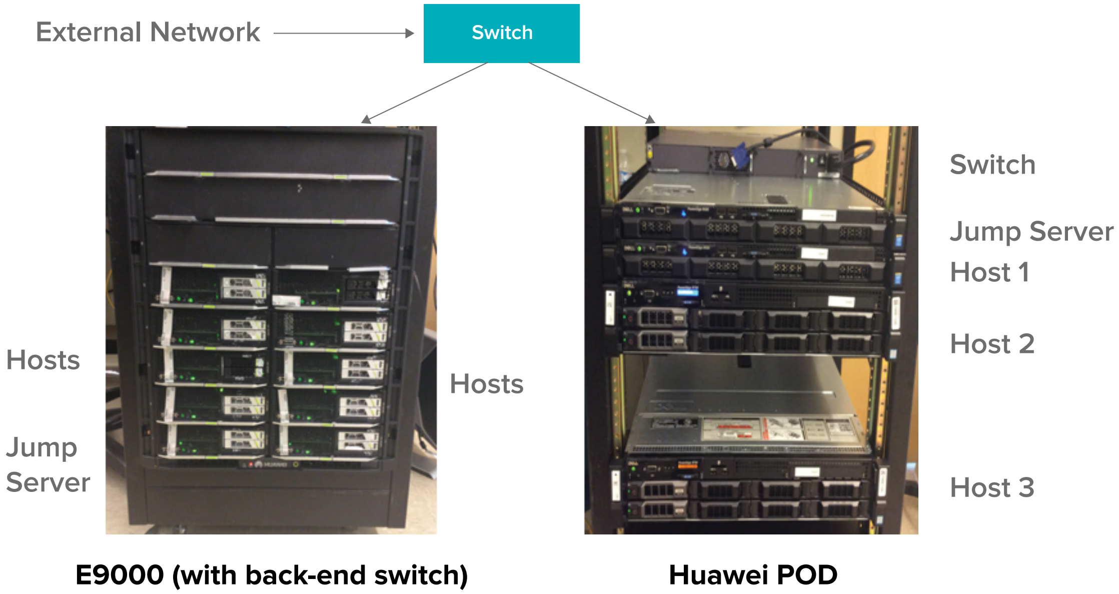
Figure 2: Intel Rack with three Pharos Pods with 1GbE and 10GbE switches


IZ1 10Gbe Switch
Cisco 4948 Switch
Intel S2600WT2 Serverboard
Raritan KVM
Intel S2600WT2 Serverboard
Intel S2600WT2 Serverboard

Huawei On-Site Pharos Pods

The testing infrastructure provided by Huawei is shown in the following diagram.

Figure 3: Huawei Testing Infrastructure



The FusionServer E9000 converged-architecture blade server enables the convergence of computing, storage, networking, and management. There are 16 slots in a 12U chassis that includes redundant Power Supply Units (PSUs), heat-dissipation modules, management modules, and switch modules.

Remote Pharos Usage

The Pharos Labs at ENEA Software in Sweden, the CENGN lab in Canada, and the Ericsson lab in Texas were used throughout to run test suites and work on interoperability issues.

The ENEA team collaborated on verifying, deploying (manual and then automated), and testing an open source vIMS VNF. The team worked through a series of hurdles to achieve 100% vIMS “live test” suites and successfully automated deployment steps that would be normally invoked in Functest by an independent cloud orchestrator.

Note: The execution of the cloud orchestrator itself could not be completed, however, due to a hard-coded test image pointer. Engagement with the author of this orchestrator configuration continues to address this issue.

Separately, a small test pod was also made available in parallel for collaboration between a JOID installer representative and an ENEA engineer to begin adding support for ARM in the JOID installer. This work continues post-Plugfest.

The ENEA Pharos lab pods utilized during the Plugfest represented multiple vendors including Applied Micro, SoftIron (AMD) and Cavium ThunderX server platforms.

The pod used for vIMS testing in ENEA Pharos lab is set up as described below and a network diagram can be seen at the following link: https://wiki.opnfv.org/display/pharos/Enea-pharos-lab.

- KVM Server (hosting Jump Servers): Intel Xeon E5-2620v3 @ 2.40GHz (2x6 cores), 64GB RAM, 2x160GB SSD (mirrored) for OS, 2x2TB HDD (mirrored) for storage, 2x10Gbps NICs (RJ45), IPMI interface (separate NIC).
- Controller nodes: 3 x Applied Micro X-Gene 2 ARMv8-64 8 cores @ 2.4GHz, 32GB RAM, 1x128GB SSD, 2x1TB HDD, 1x10Gbps SFP+ NICs, 2x1Gbps NICs.
- Compute nodes: 2 x Cavium Networks CN8890-CRB ThunderX ARMv8-64 48 cores @ 2.5GHz, 8x16GB RAM (128GB total), 1x500GB HDD, 1x40Gbps QSFP+ NIC, 2x10Gbps SFP+ NICs, 1x1Gpbs NIC (RJ45, IPMI interface).
- Extreme Networks x670 10GE L3 switch for Storage, Management, Private, Public (10 Gbps) and Lights-Out networks (1Gbps).
- HP 2530-48G 1GE L2 switch for Public, Admin and Lights-Out networks (1Gbps).

DATA COLLECTION OVERVIEW

During the Plugfest, testing data was collected via Functest, Yardstick, and StorPerf, with data being stored in a test database available to the event participants.

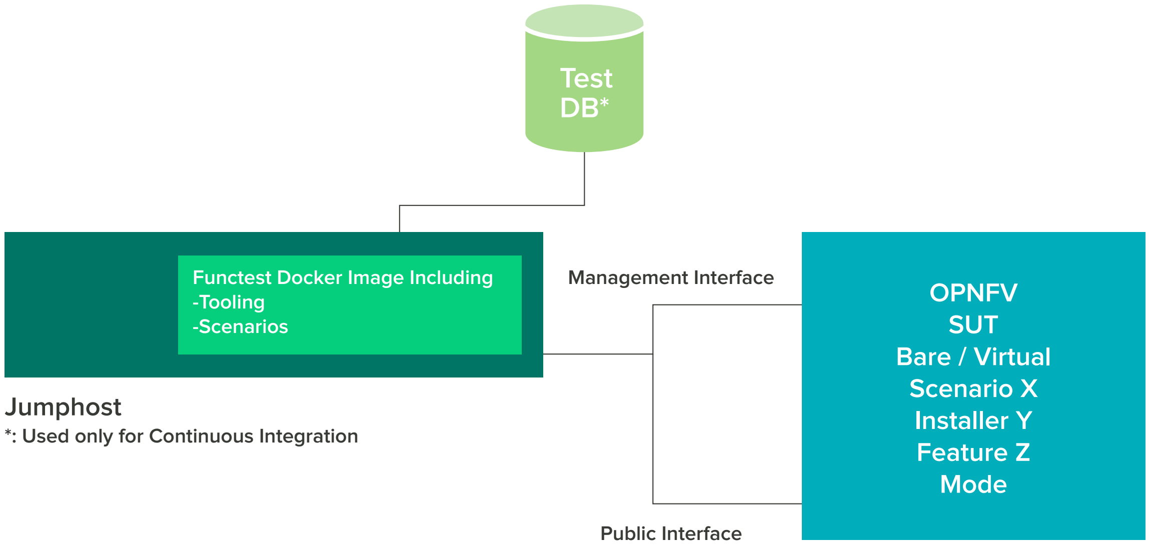
Optionally, some tests were run without writing the results up the central database, keeping things local to the platform where the test was running.

Functest

Functest provides a comprehensive testing methodology, test suites and test cases to verify OPNFV Platform functionality that covers the VIM and NFVI components of ETSI NFV. The Functest repository includes detailed functional test cases, test methodologies and platform configurations. Test cases include virtual pings, the testing of OpenStack interfaces, and interaction with SDN controllers.

The following figure shows a high level view of how Functest is used to gather data from a scenario and write it to a database.

Figure 4: Functest High Level View



Functest runs on a jump host, and is connected via a public interface and/or a management interface to a system under test (SUT). The SUT could be on bare metal or in a hypervisor, and contains an installer and some combination of features.

A total of 62 Functest test results were collected during the Plugfest. These results correspond to five different systems on three different hardware solutions, including three scenarios (os-onos-nofeature-ha, os-onos-nofeature-ha, and os-odl_l2-nofeature-ha).

Note: For an overview of scenario naming standards, see http://artifacts.opnfv.org/opnfvdocs/brahmaputra/docs/userguide/introduction.html.

The test cases were vPing, vPing_userdata, Tempest, Rally, ONOS, Promise (resource management), vIMS (see Brahamputra documentation for details): http://artifacts.opnfv.org/functest/brahmaputra/docs/userguide/index.html)

Some Tempest results were close to the same as those achieved with Functest after we integrated the solutions; they matched the success criteria (> 90%). Logs and a first level of troubleshooting was provided by the Functest project during the week. Most of the errors were due to misconfiguration or the absence of some OpenStack modules (on non-OPNFV systems).

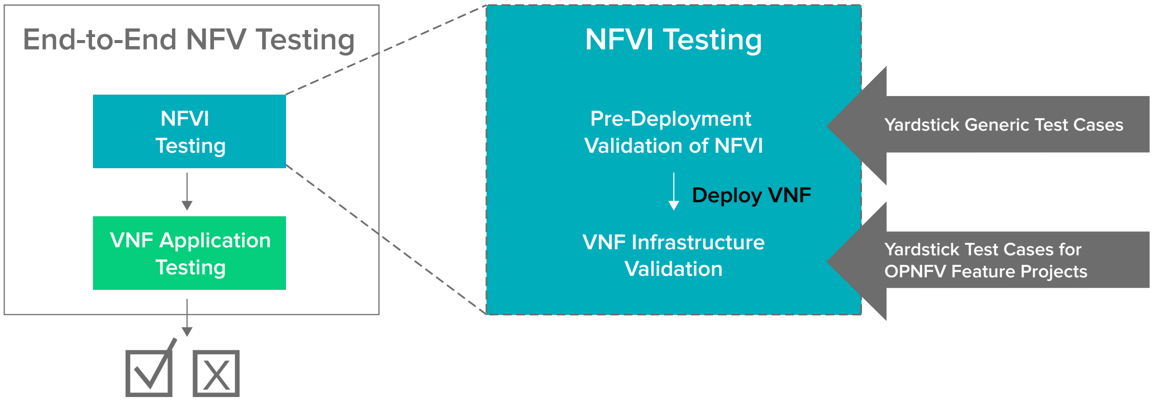
The Plugfest was a place to exchange ideas with the other test projects and to help lab owners finalize their integration. The first results on ARM pods have been pushed into the CI pods during the week and this was a significant milestone for OPNFV as it was the first time CI jobs became "multi-architecture" in terms of supporting multiple instruction set architectures (ISA). It was also an occasion to test Functest towards an OpenStack deploy in a non-OPNFV solution.

Note: The list of the systems (as well as logs) can be found on https://wiki.opnfv.org/display/EVNT/Testing+sub+page+restriction and the results can be retrieved from http://plugfest.opnfv.org/testapi/results.

Yardstick

The goal of the Yardstick Project is to verify infrastructure compliance when running VNF applications. NFV Use Cases described in ETSI GS NFV 001 show a large variety of applications, each defining specific requirements and complex configuration on the underlying infrastructure and test tools. The Yardstick concept decomposes typical VNF workload performance metrics into a number of characteristics and/or performance vectors, each of which can be represented by distinct test-cases.

Figure 5: Yardstick Generic and Infrastructure Test Cases


End-to-End NFV Testing
NFVI Testing
VNF Application Testing
☑ ☒
NFVI Testing
Pre-Deployment Validation of NFVI
Deploy VNF
VNF Infrastructure Validation
Yardstick Generic Test Cases
Yardstick Test Cases for OPNFV Feature Projects

The tests that Yardstick implemented for the Plugfest include:

- Network latency with Ping
- Memory and storage latency
- Variations in packet delay (jitter)
- Memory read and write bandwidth
- Processing speed
- Networking throughput and packet loss

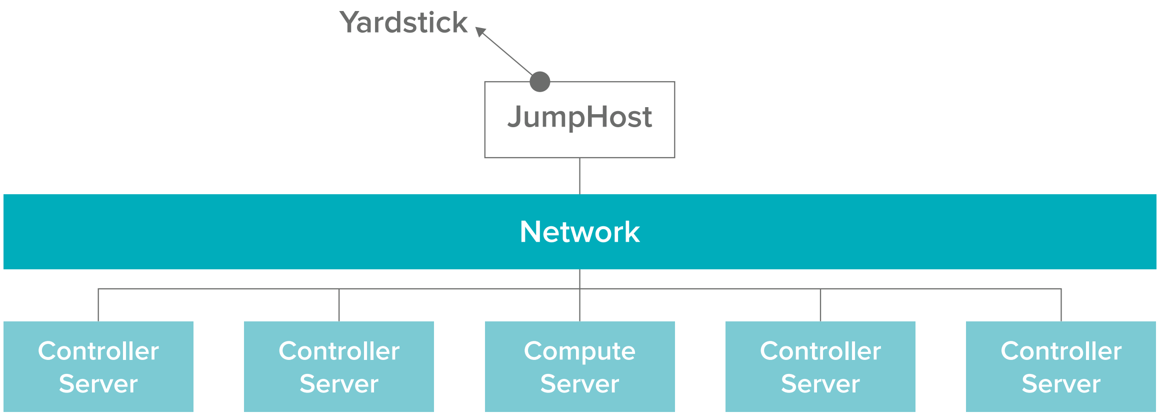
See the Brahmaputra documentation for test case details: http://artifacts.opnfv.org/yardstick/brahmaputra/docs/userguide/userguide.pdf.

Yardstick can be deployed on a virtual machine, a bare metal machine, or a container. To run Yardstick test cases, the only requirement is network connectivity between Yardstick and the system under test (SUT).

For those who wanted to push the test results to the influxdb, Yardstick had access to the influxdb server. Yardstick was run in containers hosted on each jump host during the Plugfest. The test results were pushed to the influxdb server (a virtual machine hosted on a Google Compute Engine).

A deployment view of Yardstick is shown in the following diagram.

Figure 6: Yardstick Deployment View



To run Yardstick test cases, participants ran Docker containers and exported the pod environment, as well as the OpenStack credential environment. We then created the Yardstick configuration file, built and uploaded it as a guest image to OpenStack Glance, and ran the final Plugfest test suite.

Note: To make Yardstick easier to use, we created a wiki page at https://wiki.opnfv.org/display/yardstick//Notes+on+installing+Yardstick+with+VM during the plugfest.

Over the course of the week, Yardstick was used to test four different systems on three different hardware solutions, using two scenarios (os-onos-nofeature-ha, os-odl_l2-nofeature-ha), and three installers (Compass, Fuel, commercial). Three of the systems pushed their test results to the result database.

Note: For an overview of scenario naming standards, see: http://artifacts.opnfv.org/opnfvdocs/brahmaputra/docs/userguide/introduction.html.

Just through one simple ping (network latency test case), we found a misconfiguration of the OpenStack Nova Compute module. This reflects the value of Yardstick, which will eventually evolve with service level agreements (SLA) for defined benchmark metrics. This will make it much easier to verify if a system can run a VNF.

StorPerf

The StorPerf project provides a SNIA-like performance report of Cinder block storage, using various Cinder backend drivers. See http://www.snia.org/sites/default/files/SSS_PTS_Enterprise_v1.1.pdf.

StorPerf measures latency, throughput and IOPs across a matrix of block sizes, queue depths, breadth of application VMs, and read/write workloads. The workloads are executed on logical volumes that are managed by Cinder.

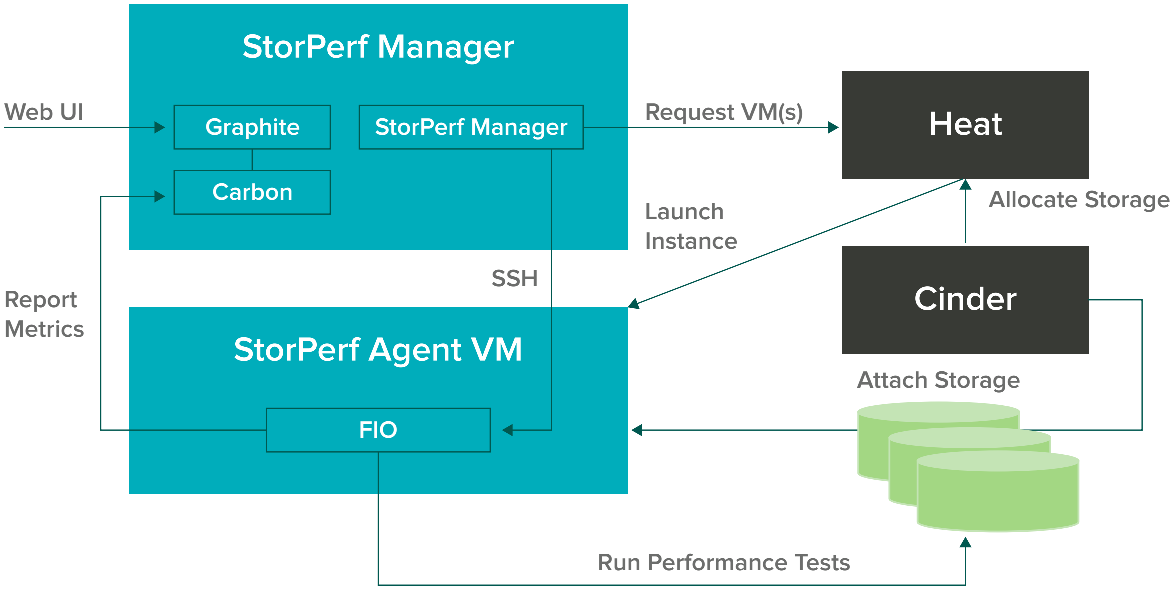
As a result, raw disk performance is not being measured, and thus there is no control over physical disk caching or disk initialization. The reality of spinning disk versus solid state is also abstracted away, presenting additional challenges in determining statistically valid performance metrics. The concept of steady state, where metrics are observed to be within established parameters, is used to qualify the statistics and discard results that are known to be invalid according to the guidelines published by SNIA.

At the Plugfest, the matrix was constrained to:

- Queue depths: 1, 4, 8
- Block sizes: 4k, 16k
- Volume size: 100 GB
- Volume/agent count: 8
- Workload: Mix of 70% Read / 30% Write

StorPerf is deployed in a Docker container that can run either in a VM or directly on a jump host in a pod:

Figure 7: StorPerf Architecture



Both Pharos Intel pod 9 and a service provider's pod 2 were targeted for use. Intel pod 9 was dedicated to integration with freeware software defined storage, and the service provider's pod 2 was dedicated to Ceph with LVM backend. In both pods, StorPerf was installed as a container right on the jump host of the pod.

Over the course of the week, StorPerf integrated test results database reporting and generated its first official test result from a run against pod 2.

Integration of Functional and Performance Testing

A key purpose of OPNFV is to coordinate continuous integration and testing of NFV solutions as an open community effort. At the Plugfest, we put together a broad and complete view of OPNFV's entire suite of testing options. This section shows how the projects described above are integrated.

OPNFV approaches testing as "function first, then measure." Example: Functest for functional testing of the platform, then Yardstick (metric testing) for system level performance testing. Individual performance projects for components feed into Yardstick.

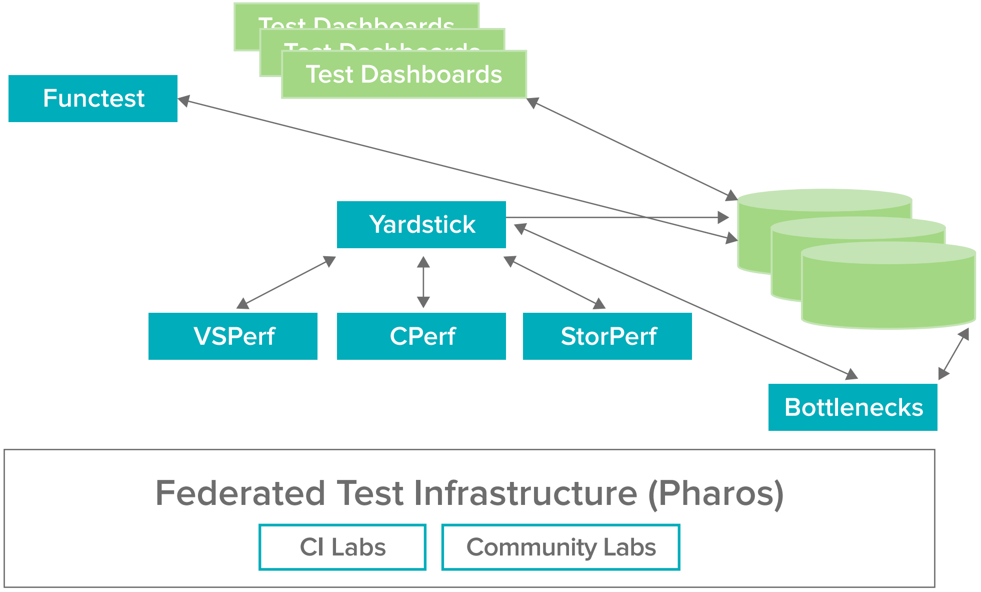
Functional testing is performed by:

- *Functest*: base system functionality, component testing, increase in number and types of tests – functionality of the VIM (when a VNF is running)

Performance testing is performed by:

- *Yardstick*: end to end system level based on ETSI TST001
- *Bottlenecks*: verify OPNFV boundaries in staging before bringing VNFs
- *StorPerf*: Storage performance for any VNF
- *VSPerf*: OVS characterization
- *CPerf*: controller performance

Figure 8: The Recommended OPNFV Approach to Testing



For example, both Yardstick and Functest feed data into a database that offers analytics for Bottlenecks, which feeds back to Yardstick.

For more information, see https://wiki.opnfv.org/pages/viewpage.action?spaceKey=EVNT&title=Testing+sub+page+restriction.

PLUGFEST TEST SCENARIOS

Major Plugfest test scenarios are discussed in the following sections.

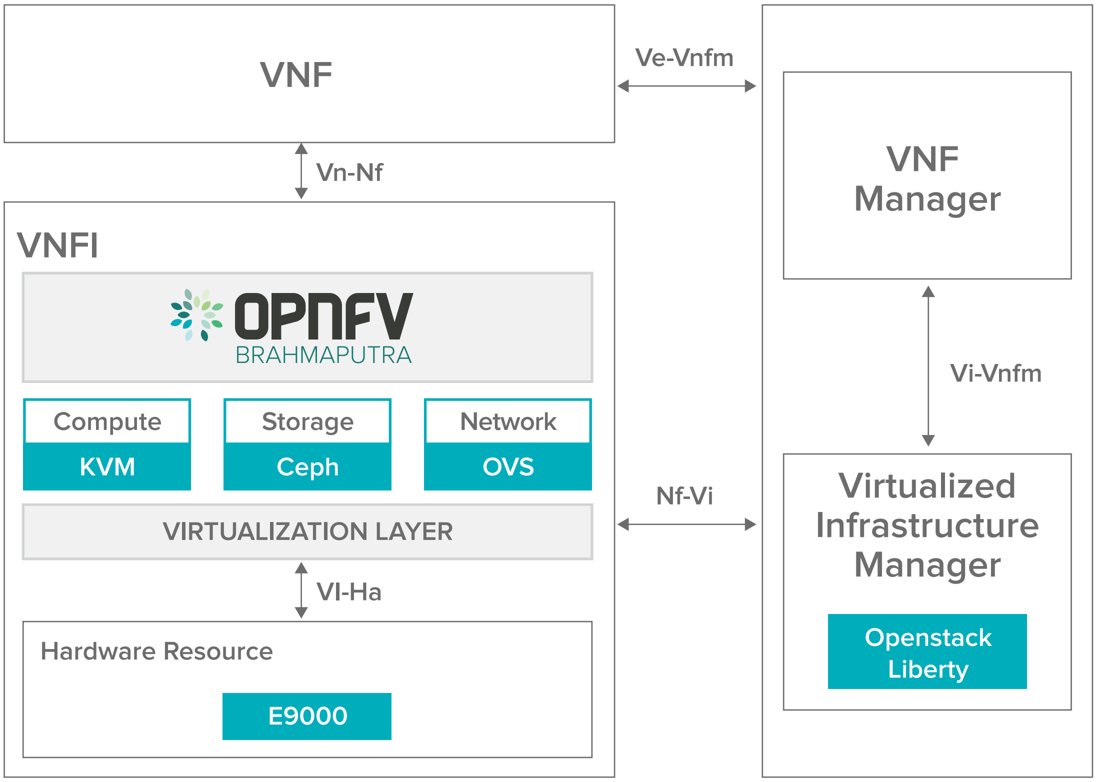
E2E Interoperability Test

The virtual EPC application was a successful example collaboration between a major APAC service provider, multiple system vendors and a software VNF vendor. It consisted of the following activities:

- Provide E2E integration including common, off the shelf (COTS) hardware, VNF Manager (VNFM), and a VNF (vEPC) based on OPNFV Brahmaputra
- Use the Compass installer
- Contribute 50 integration test cases to OPNFV based on interface
- Execute test activity on a remote OPNFV pod

The test is shown in the following figure.

Figure 9: E2E Interoperability test

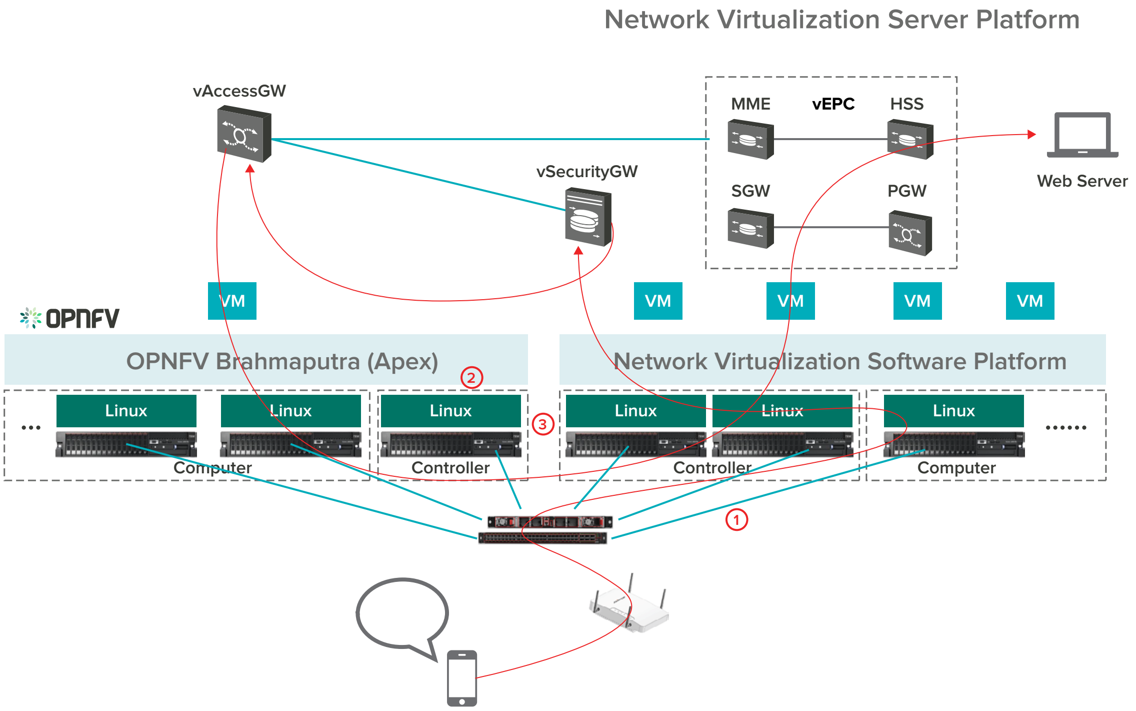


Virtual EPC Proven with a Femtocell Base Station

The following activities were involved in implementing a vEPC application:

- An Integrated and feature-complete network virtualization software platform with OPNFV Brahmaputra installed by Apex and running on x86 servers as NFVI
- VNF (vSecurityGW/vApplicationGW/vEPC) is running on the same hardware with OPNFV Brahmaputra.
- Through hardware, NFVI and VNFs, a mobile phone connected to the Internet through a Femtocell Base Station

Figure 10: Virtual EPC Test Case



Software-Defined Scalable Storage on Multiple Installers

Going into the Plugfest, a goal was set to integrate a freeware storage subsystem into three of the OPNFV installers: Apex, Fuel and JOID.

Due to the commercial support nature of the software, the installation path was not clearly defined when it came to pure open source projects. The Apex team was able to provide modifications to the upstream puppet scripts and successfully deploy by the end of the week.

Both Fuel and JOID encountered challenges arising from the use of the latest OpenStack releases against commercialized software certified against older OpenStack releases (i.e., Mitaka vs. Liberty). As this is an OPNFV plugfest, Mitaka was used, but this version of OpenStack is not officially supported by the freeware storage subsystem. The work on installing this software with Fuel and JOID continued past the end of the Plugfest.

vIMS on Multiple HW with Multiple Installers

As mentioned in the Pharos Lab section above, the Armband team performed extensive testing of vIMS and successfully automated deployment steps that would be normally invoked in Functest by an independent cloud orchestrator. Work continues beyond the Plugfest to fix the cloud orchestrator to invoke the correct image in Functest, and upstreaming all patches identified during the porting effort.

The Armband team began to work with the JOID representative, setting up a dedicated two node pod in the ENEA Sweden Pharos Lab, and beginning the porting of the JOID Installer for ARM-based servers. Work continues beyond the Plugfest.

In addition, an Apex Installer representative began to scope out the work with the Armband team on adding that Installer to ARM-based pods. Credentials for access to the Pharos Lab were provided by the ENEA team and work continues beyond the Plugfest. The Plugfest was instrumental in fostering collaboration between teams, fostering new scenarios being prepared for future CI testing.

VNF on Multiple Hardware

An NFVI monitoring and analytics engine was successfully integrated and deployed on an OpenStack-based NFV server as a VNF. This was accomplished on the Intel community pod NFVI platform. The engine was packaged and tested as both a KVM-based VM image as well as a Docker-based container VM.

OpenDaylight with New Installers and Test Tools

An OPNFV CPerf project representative presented at the Plugfest and met with service providers and installer teams. The use of OpenDaylight and other controllers in cloud and NFV use cases was a major topic. For OpenDaylight, upstream work was started for Fuel and JOID installers, and CPerf testing has evolved.

CENGN Consortium Progresses its SDN/NFV Initiatives

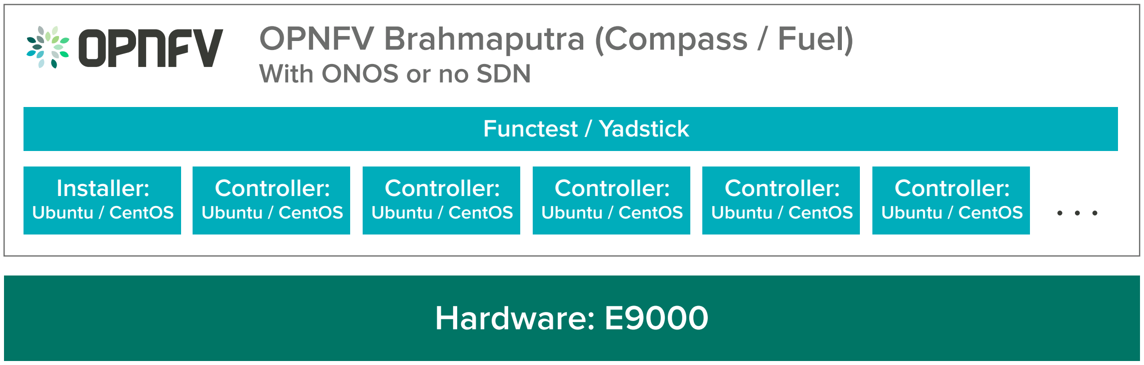
CENGN set out to build NFVI on a private remote pod containing nodes with mixed hardware configurations that introduce some new challenges to the OPNFV's NFVI deployers. The JOID project and the OpenStack Charm team worked with CENGN to identify issues and needed solutions to deploy a successful NFVI on this pod. Functest was run on the pod, and Yardstick was also run with the assistance of the Yardstick team.

CENGN continues to work with the JOID team beyond the Plugfest to test and improve the deployer for the Colorado Release as well as put lessons learned into CENGN's OPNFV Community Pharos Lab (cengnpod1).

Validation with Multiple Installers (Compass/Fuel)

In the Plugfest, relevant scenarios from the Brahmaputra release were deployed with two installers (Compass and Fuel) and two operating systems (Ubuntu and CentOS). in these scenarios, Functest and Yardstick were executed as shown below.

Figure 11: Scenarios with Multiple Installers and Controllers


OPNFV
OPNFV Brahmaputra (Compass / Fuel)
With ONOS or no SDN
Functest / Yadstick
Installer: Ubuntu / CentOS
Controller: Ubuntu / CentOS
Controller: Ubuntu / CentOS
Controller: Ubuntu / CentOS
Controller: Ubuntu / CentOS
Controller: Ubuntu / CentOS
. . .
Hardware: E9000

Each scenario provides specific capabilities and/or components; these were used to solve specific problems for the deployment of VNFs. In the Plugfest, two basic scenarios (os-onos-ha and os-nosdn-ha) were executed and validated successfully.

Note: For an overview of scenario naming standards, see: http://artifacts.opnfv.org/opnfvdocs/brahmaputra/docs/userguide/introduction.html.

Further tests, which include Functest and Yardstick, were executed to verify basic NFV functions, including hardware compatibilities and virtualization capabilities. In these two scenarios, two different OS versions (Ubuntu 14.04 and CentOS 7) were also tested and all results were uploaded into the Plugfest results database.

CONCLUSION

The inaugural OPNFV Plugfest was an important step in demonstrating the progress of NFV and illustrating the plans of vendors and operators to build virtualized networks.

Participants verified many new scenarios on OPNFV Brahmaputra, and gained some insights into new directions that might be taken for the Colorado release.

Many Plugfest participants stated enthusiastically that the face-to-face collaboration was a major benefit of the event. All in all, the Plugfest provided a chance to test and interconnect NFV solutions based on OPNFV, verified the latest OPNFV implementations, facilitated communication between the industry and the OPNFV community, and improved the OPNFV reference platform itself, constributing to a healthy and diverse ecosystem.

The major findings are summarized in the following section.

Summary of Major Findings

A summary of the accomplishments for each type of participant is given in the following table.

Table 1: Summary of Major Findings

Key Testing Area	Accomplishments
OPNFV Deployments: hardware, platforms, installers	Additional installer support was provided for new platforms such as a carrier-grade NFV platform and other NFVI implementations. Both local pods and Pharos labs were utilized for this purpose. Upstream enhancements to installers were recommended and planned. Alternate storage subsystems were installed and validated on all four major installers (Apex, Compass, Fuel and JOID), including their interaction with OpenStack, and upstream needs were also identified. Semiconductor vendors installed and tested OPNFV on their own “white boxes” to gather usability information. All major installers and testing projects were utilized for this effort.
Network Integration: testing scenarios for components in different configurations	Functional and performance tests were integrated in valuable ways and the data was stored offline for further analysis. In particular, Yardstick measurements were run on multiple platforms, installers and applications. Controller integration was achieved for both OpenDaylight and ONOS. For ODL, upstream work was started for Fuel and JOID, and CPerf testing was evolved. ONOS was installed with Fuel using Ubuntu as the host OS, and was tested with both Functest and Yardstick.
VNF Applications in new environments	A vIMS and a vEPC system were successfully deployed on new combinations of hardware and software. Many application VNFs were tested on all four installers. A virtual performance and analytics was also installed and tested in a multivendor environment.
E2E Interoperability	A collaborative team provided E2E integration including COTS hardware, VNFM, VNF (vEPC) based on the OPNFV Brahmaputra release. The team also used the Compass installer tool, and contributed 50 integration test cases to OPNFV based on discoveries in testing NFV reference points.

Thoughts for Next Plugfest and Future Planning

For future Plugfests, participants recommended that we set up dedicated hardware resources for each installer team, and that we increase parallel work sessions—such as ones that were set up for Yardstick and OpenDaylight. Along these lines, the participants were quick to identify onsite hardware as key to the event's successes.

Suggestions were also made to deploy the hardware ahead of time and keep the labs open later into the evening, possibly even allowing remote access (by the participants) during closed hours to monitor tests running overnight, etc. The breakout sessions were considered extremely valuable and more breakouts will be scheduled in advance for the next Plugfest. There was also a discussion on breaking the Plugfest into two parts, with the first part focused on install tools and testing, and the second part more geared toward VNF vendors and interoperability.

The next Plugfest is tentatively planned for the second half of 2016, following the OPNFV Colorado Release. Registration and information pages will be created and shared with the community. The entire community is invited to participate in the Plugfest planning meetings (https://wiki.opnfv.org/display/EVNT/Plugfest).

Anyone interested in advancing open source NFV is welcome to participate in the OPNFV technical community, and technical meetings are open to all, including non-members. For more information on OPNFV technical projects, please visit wiki.opnfv.org.

APPENDIX: BRAHMAPUTRA PLUGFEST PARTICIPANTS

The following companies participated in the Plugfest. Many thanks to all the participants who helped make the event such a success!

Table 2: Plugfest Participants

Name	Description	URL
ARM	Semiconductor company	https://www.arm.com/
BII Group	Public service platform for internet infrastructure	http://www.biigroup.com/en/
CableLabs	Nonprofit R&D consortium for cable providers.	http://www.cablelabs.com/
Canonical	Producer of Ubuntu and associated commercial services.	http://www.canonical.com/
Cavium	Semiconductor company	http://www.cavium.com/
CENGN	CENGN: Research consortium for next gen solutions	http://www.cengn.ca/
China Mobile	Telecommunication company; service provider	http://www.chinamobileltd.com/en/global/home.php
Ericsson	Network equipment vendor providing communication technology and services	https://www.ericsson.com/
EMC	Information management company providing data storage and other products	http://www.emc.com/en-us/index.htm
ENEA	Information technology company	http://www.enea.com/
Huawei	Networking and telecom equipment vendor	http://www.huawei.com/en/
Intel	Semiconductor and computing vendor	http://www.intel.com/content/www/us/en/homepage.html
Mirantis	OpenStack software vendor	https://www.mirantis.com/
Orange	Telecommunication company; service provider	http://www.orange.com/en/home
Raisecom	Networking solutions vendor	https://www.raisecom.com/index
Red Hat	Provider of open source solutions	https://www.redhat.com/en
Sytech	Networking solutions; NFVI/VNF vendor	http://www.sytech-solutions.com/
Wind River	Embedded and open source software	http://www.windriver.com/
UNH-IOL	Independent interoperability testing lab at University of New Hampshire	https://www.iol.unh.edu/