FastDataStacks
Building a fast and flexible platform for high performance applications using FD.io

Frank Brockners, Distinguished Engineer, Cisco
Michael Weir, Vice President, CENGNI

September 26, 2016
Building NFV stacks

“The 20th century was about invention, the 21st is about mashups and integration” – Toby Ford, ATT

- OPNFV performs System Integration as an open community effort:
  - Create/Evolve Components (in lock-step with Upstream Communities)
  - Compose / Deploy / Test
  - Iterate (in a distributed, multi-vendor CI/CD system)
- Scenarios in OPNFV Arno and Brahmaputra focused on establishing the base infrastructure, system test, and system-control focused features
- Let’s add “networking” as another focus...
Foundational Assets For NFV Infrastructure: A stack is only as good as its foundation

- **Virtual Forwarder**
  - Feature rich, high performance, highly scalable virtual switch-router
  - Leverages hardware accelerators
  - Runs in user space
  - Modular and easy extensible

- **Forwarder Diversity: Hardware and Software**
  - Virtual Domains link and interact with physical domains

- **Domains and Policy**
  - Connectivity should reflect business logic instead of physical L2/L3 constructs

---

- **Service Model**
- **WorkFlow Topology**
- **App Intent**

- **Service/WF Life Cycle Manager**
- **Virtual Machine/Container Life Cycle Manager**

- **Network Controller Forwarder – Switch/Router**
Networking Foundation for NFV Infrastructure

Choices

• VPP
  • Highly scalable, high performance, extensible virtual forwarder

• OpenDaylight
  • Extensible controller platform
  • Decouple business logic from network constructs: Group Based Policy as mediator between business logic and network constructs
  • Support for a diverse set of network devices
  • Clustering for HA
Evolving The OPNFV Scenario Set

- OPNFV uses “scenarios” (i.e. compositions of features and their configuration) as key release vehicle
- OPNFV scenarios in the Brahmaputra release were focused on OVS as virtual forwarder
- Create a new stack which significantly evolves networking for NFV
- Introduce Scenarios with VPP for OPNFV
  - `os-odl_l2-fdio-noha`, `os-odl_l2-fdio-ha`
  - `os-odl_l3-fdio-noha`, `os-odl_l3-fdio-noha`

<table>
<thead>
<tr>
<th>Category</th>
<th>Components in OPNFV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Install Tools</td>
<td>Apex, Compass, Fuel, Juju</td>
</tr>
<tr>
<td>VM Control</td>
<td>OpenStack</td>
</tr>
<tr>
<td>Network Control</td>
<td>OpenDaylight, ONOS, OpenContrail</td>
</tr>
<tr>
<td>Hypervisor</td>
<td>KVM, KVM4NFV</td>
</tr>
<tr>
<td>Forwarder</td>
<td>OVS, OVS-DPDK + VPP</td>
</tr>
</tbody>
</table>
OPNFV FastDataStacks (FDS)

• Integrate VPP into existing OPNFV scenarios

• Initial scenarios
  - OpenStack – ODL (Layer2) – VPP
  - OpenStack – ODL (Layer3) – VPP
  - OpenStack – VPP
  - ...

• Diverse set of contributors:

• https://wiki.opnfv.org/display/fds
FastDataStacks Component Development

- **OpenDaylight**
  - GBP Neutron Mapper
  - GBP Renderer Manager enhancements
  - VPP Renderer
  - Virtual Bridge Domain Mgr / Topology Manager

- **FD.io**
  - HoneyComb – Enhancements
  - VPP – Enhancements
  - CSIT – VPP component tests

- **OPNFV**
  - Overall System Composition – Integration into CI/CD
  - Installer: Integration of VPP into APEX
  - System Test: FuncTest and Yardstick system test application to FDS

See also: [FDS Architecture](https://wiki.opnfv.org/display/fds/OpenStack-ODL-VPP+integration+design+and+architecture)
Example: Creating a Neutron vhostuser port on VPP

1. POST PORT (id=<uuid>, host_id=<vpp>, vif_type=vhostuser)
2. Update Port
3. Map Port to GBP Endpoint
4. Update/Create GBP Endpoint (L2 context, MAC,...)
5. Apply Policy
6. Update node(s), bridge-domain
7. Netconf Commit (bridge config, tunnel config)
8. Netconf Commit (vhostuser i/f config, bridge config)
9. Update device end point

Network components:
- Neutron
- Neutron NorthBound
- GBP Neutron Mapper
- GBP Renderers Manager
- VPP Renderers
- Topology Manager (vBD)
- VXLAN Tunnel
- VPP 1
- VPP 2
- VM
- Netconf/YANG
FastDataStacks: OS – ODL(L2) – FD.io
Example: 3 node setup: 1 x Controller, 2 x Compute
Running a FastDataStack (1/2)

1. **Get an image**
   
   ```
   wget -O /tmp/cirros-0.3.4-x86_64-disk.img http://download.cirros-cloud.net/0.3.4/cirros-0.3.4-x86_64-disk.img
   ```

2. **Upload the image into Glance and make it support hugepages**
   
   ```
   glance image-create --name "cirros-0.3.4-x86_64" --file /tmp/cirros-0.3.4-x86_64-disk.img --disk-format qcow2
   --container-format bare --visibility public --progress
   glance image-update --property hw_mem_page_size=large ad74564d-fd22-414b-9fa3-619e87f781a9
   ```

3. **Create a flavor with hugepages enabled**
   
   ```
   nova flavor-create --is-public true opnfv 42 768 1 1
   nova flavor-key opnfv set hw:mem_page_size=large
   ```

4. **Create a Neutron network**
   
   ```
   neutron net-create vxlannet --provider:network_type vxlan
   ```
Running a FastDataStack (1/2)

5. **Assign a Subnet to your network**
   
   ```bash
   neutron subnet-create vxlanet 10.11.12.0/24 --name vxlansubnet
   ```

6. **Create a Neutron Router**
   
   ```bash
   neutron router-create r1
   neutron router-interface-add r1 subnet=vxlansubnet
   neutron router-gateway-set r1 external
   ```

7. **(Optional) Create Ports for your VMs**
   
   ```bash
   neutron port-create vxlanet --name vm-compute-0-1-port
   neutron port-create vxlanet --name vm-compute-1-1-port
   ```
   
   (booting the VMs will do this for you)

8. **Boot your VMs**
   
   ```bash
   nova boot --image cirros-0.3.4-x86_64 --flavor opnfv
   --nic port-id=' neutron port-list | grep vm-compute-0-1-port | cut -f 2 -d " " vm-compute-0-1
   --availability-zone nova:overcloud-novacompute-0.localdomain.com
   nova boot --image cirros-0.3.4-x86_64 --flavor opnfv
   --nic port-id=' neutron port-list | grep vm-compute-1-1-port | cut -f 2 -d " " vm-compute-1-1
   --availability-zone nova:overcloud-novacompute-1.localdomain.com
   ```
Network Setup Complete:
Example OpenStack Perspective: Check your ports

```
[root@overcloud-controller-0 ~]# neutron port-list

+-----------------+-----------------+-----------------+-----------------+
| id              | name            | mac_address     | fixed_ips       |
+-----------------+-----------------+-----------------+-----------------+
| 8904fbf0-5a9b-  |                | fa:16:3e:8c:8f:f | "subnet_id": "1e9a |
| 48f2-b88f-      | 2               | 5a26-478e-4837-a659 |
| e115093d2594    |                 | -ca7f6105a7e3", | "ip_address": |
+-----------------+-----------------+-----------------+-----------------+
| 9bf477cb-a016   | vm-             | fa:16:3e:db:e5:0 | "subnet_id": "1e9a |
| -407e-a97a-     | 2               | 5a26-478e-4837-a659 |
| 7f15b35c3bad    |                 | -ca7f6105a7e3", | "ip_address": |
+-----------------+-----------------+-----------------+-----------------+
| b504c2a4-6c6c-41| vm-             | fa:16:3e:99:24:4 | "subnet_id": "1e9a |
| c0-b011-a3e747c6| compute-1-1-port | 1               | 5a26-478e-4837-a659 |
| 3568            |                 | -ca7f6105a7e3", | "ip_address": |
+-----------------+-----------------+-----------------+-----------------+
| f70a5d22-e6cd-  | vm-             | fa:16:3e:3f:50:e | "subnet_id": "1e9a |
| 4c36-8b7c-      | 4               | 5a26-478e-4837-a659 |
| 66ddd14c469f    |                 | -ca7f6105a7e3", | "ip_address": |
+-----------------+-----------------+-----------------+-----------------+
```

DHCP tap port
Port for VM #1
Port for VM #2
qrouter tap port
Network Setup Complete:
Example ODL Perspective: Active Network Topology

{
  "network-topology": {
    "topology": [
      {
        "link": [
          {
            "destination": {
              "dest-node": "overcloud-novacompute-0.opnfvapex.com",
              "dest-tp": "vxlan_tunnel5"
            },
            "link-id": "overcloud-novacompute-0.opnfvapex.com-1-overcloud-novacompute-0.opnfvapex.com",
            "source": {
              "source-node": "overcloud-novacompute-1.opnfvapex.com",
              "source-tp": "vxlan_tunnel4"
            },
            "vbridge-topology:tunnel": "vxlan_tunnel4"
          }
        ]
      },
      {
        "destination": {
          "dest-node": "overcloud-novacompute-0.opnfvapex.com",
          "dest-tp": "vxlan_tunnel4"
        },
        "link-id": "overcloud-novacompute-0.opnfvapex.com-1-overcloud-controller-0.opnfvapex.com",
        "source": {
          "source-node": "overcloud-novacompute-0.opnfvapex.com",
          "source-tp": "vxlan_tunnel4"
        },
        "vbridge-topology:tunnel": "vxlan_tunnel4"
      }
    ]
  }
}
Network Setup Complete: Example HoneyComb Perspective: Interfaces

```bash
[root@overcloud-controller-0 ~]# curl -X GET -v -u admin:admin http://localhost:8182/restconf/config/ietf-interfaces:interfaces/ | python -m json.tool
```

```
% Total    % Received % Xferd Average Speed Time Time Time Current
          0     0     0     0     0     0     0     0          0 --:--:-- --:--:-- --:--:--   0* About to connect() to localhost port 8182 (#0)
* Trying ::1...
* Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8182 (#0)
* Server auth using Basic with user 'admin'
> GET /restconf/config/ietf-interfaces/ HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.29.0
> Host: localhost:8182
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Sun, 18 Sep 2016 10:29:10 GMT
< Content-Type: application/yang.data+json
< Transfer-Encoding: chunked
< Server: Jetty(9.3.11.v20160721)
<
{
  "interfaces": [
    {
      "description": "neutron port",
      "enabled": true,
      "link-up-down-trap-enable": "enabled",
      "name": "neutron_port_f70a5d22-e6cd-4c36-8b7c-66ddd14c469f",
      "type": "v3po:tap",
      "v3po:tap": {
        "mac": "fa:16:3e:3f:50:e4",
        "tap-name": "qr-f70a5d22-e6"
      }
    }
  ]
}
```

C0

C1

Ctl

QR

DHCP
Network Setup Complete:
Example VPP perspective - vppctl show int addr

[root@overcloud-controller-0 ~]# vppctl show int addr
TenGigabitEthernet7/0/0 (up):
    192.168.0.23/24
local0 (dn):
tap-0 (up):
    l2 bridge bd_id 1 shg 0
tap-1 (up):
    l2 bridge bd_id 1 shg 0
vxlan_tunnel0 (up):
    l2 bridge bd_id 1 shg 1
vxlan_tunnel1 (up):
    l2 bridge bd_id 1 shg 1
VMs up and running – Ready to Ping 😊

```
[root@overcloud-controller-0 ~]# nova list

+---------------------------------+----------+--------+-------------+----------------+------------------+
| ID                              | Name     | Status | Task State | Power State   | Networks         |
+---------------------------------+----------+--------+-------------+----------------+------------------+
| 78a2a66d-b73c-470d-9359-5bd0a076e61 | vm-compute-0-1 | ACTIVE | -           | Running        | vxlan net=10.11.12.3 |
| 05413042-1446-4cf0-ac93-9f2ba3dfc984 | vm-compute-1-1 | ACTIVE | -           | Running        | vxlan net=10.11.12.4 |
```
FastDataStacks: Status and near term roadmap

Colorado 1.0
- Base O/S-ODL(L2)-VPP stack (Infra: Neutron / GBP Mapper / GBP Renderer / VBD / Honeycomb / VPP)
  - Automatic Install
  - Basic system-level testing
  - L2 networking using ODL (no east-west security groups), L3 networking uses qrouter/OVS
  - Overlays: VXLAN, VLAN

Colorado 2.0
- Enhanced O/S-ODL(L2)-VPP stack (Infra complete: Neutron / GBP Mapper / GBP Renderer / VBD / Honeycomb / VPP)
  - Enhanced system-level testing
  - High-availability (OpenStack and ODL)
  - L2 networking using ODL (incl. east-west security groups), L3 networking uses qrouter/OVS
  - O/S-VPP (Infra: Neutron ML2-VPP / Networking-vpp-agent / VPP)
    - Automatic Install
    - Overlays: VLAN

Colorado 3.0
- Enhanced O/S-ODL(L3)-VPP stack (Infra complete: Neutron / GBP Mapper / GBP Renderer / VBD / Honeycomb / VPP)
  - L2 and L3 networking using ODL (incl. east-west security groups)
  - L3 networking via VPP (replacing qrouter), incl. NAT/floating-ips
FastDataStacks Validation

- As part of the effort towards increasing adoption, FDS is being validated on 3rd party hardware
  - OPNFV Linux Foundation reference lab (Cisco UCS-B)
  - Cisco OPNFV labs (UCS-B and UCS-C)
  - CENGN: OPNFV Pharos (Kontron)
- Thanks to CENGN, FDS is now being validated on a Kontron SymKloud server.
  - SymKloud MS2910
  - 10G internal networking
  - 6 blades:
    - Xeon 8C D-1548
    - 32GB RDIMM memory
    - 128GB SSD
    - 2x1TB HDD
OPNFV Functional Testing: Project FuncTest

<table>
<thead>
<tr>
<th>vPing SSH*</th>
<th>ODL suite*</th>
<th>Promise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify private &amp; public connectivity</td>
<td>Robot framework, ODL functional testing</td>
<td>Resource reservation and management project</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>vPing userdata*</th>
<th>ONOS suite</th>
<th>Doctor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify nova-metadata service and private connectivity</td>
<td>TestON framework</td>
<td>Fault management and maintenance project</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tempest test*</th>
<th>vIMS</th>
<th>BGPVPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenStack native tests (200+ tests)</td>
<td>OpenSource solution by Clearwater</td>
<td>Neutron BGPVPN project integration</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rally bench tests*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmark the OpenStack deployment</td>
</tr>
</tbody>
</table>

*FDS currently only runs the highlighted (dark black) test suites
FDS: FuncTest Test Results Summary

- os-odl_l2-fdio-noha

<table>
<thead>
<tr>
<th>vPing (ssh)</th>
<th>vPing (userdata)</th>
<th>Tempest (smoke)</th>
<th>Rally (smoke)</th>
<th>ODL</th>
</tr>
</thead>
<tbody>
<tr>
<td>☀</td>
<td>☀</td>
<td>☁</td>
<td>☀</td>
<td>☀</td>
</tr>
</tbody>
</table>
FastDataStacks OPNFV CI

• As FDS increases its automated test coverage, it will be able to utilize the CENGN Pharos pod to run its CI jobs using OPNFV’s Jenkins CI tool.

• This will facilitate testing of continued development as FD.io moves forward in its development, allowing FDS to keep pace.
CENGN and VPP

• Prior to contributing to FDS, CENGN presented a PoC at the 2016 OPNFV conference that featured VPP.

• Using a customized ODL distribution provided by Inocybe, CENGN integrated a virtual data path underlay provided by VPP and ODL with Juniper’s Contrail Cloud to provide VNFs to tenants over a shared infrastructure.
FastDataStacks Summary

• Create a new stack which significantly evolves networking for NFV: Introduce Scenarios with VPP for OPNFV

• OPNFV Colorado 1.0 integrates
  • OpenStack Mitaka (uses v2 of ML2 ODL driver)
  • OpenDaylight Boron GBP additions & enhancements (GBP Neutron Mapper, VPP Renderer)
  • HoneyComb 16.09
  • VPP 16.09
  • OPNFV APEX/TripleO Installer integration
  • OPNFV System-level testing

• Colorado 2.0/3.0 will add HA, more security, Layer 3 with VPP
https://wiki.opnfv.org/display/fds
#opnfv-fds
fds-dev@lists.opnfv.org