Increasing Infrastructure Efficiency via Optimized NFV Placement in OpenStack Clouds

Yathiraj Udupi, Debo Dutta – Cisco
Ram (Ramki) Krishnan - Brocade

OpenStack Atlanta Summit, May 2014
Who and why?

Debo/Yathi - Cisco Cloud CTO office
Ramki - Brocade CTO office
Goal: Drive Innovative Open Source solutions for NFV with OpenStack
Our Thesis

• Toby Ford@AT&T’s NFV talk on Tue, May 13th
  • Worlds of IT and Telco are coming together
  • Telco Cloud - OpenStack as the infrastructure foundation

• Goal: Transform OpenStack to a Carrier-grade cloud solution
  • We deep dive into some high-level gaps Toby identified
  • We demo some initial progress
Agenda

• NFV Summary
• Cloud NFV Use Case
• Drive Innovation - Efficient Resource Placement Strategies
• Extensions to OpenStack scheduler
• Conclusion
Network Functions Virtualization (NFV)

NFV Vision

- Global movement by network operators - AT&T, Verizon, BT, CenturyLink, Deutsche Telekom, Telefonica, KDDI etc.
- General purpose hardware - OPEX and CAPEX savings
- Increased automation – OPEX savings, faster time to market
- New business models and value added services

Source: ETSI NFV White Paper
NFV Use Case - NFVlaas

Motivation

- Network Functions in the cloud
- Combined value – Infrastructure as a service (IaaS) – Compute/storage infra, Network as a service (Naas) – WAN network infra
- Leverage NFV Infra of another SP – increase resiliency, reduce latency (CDN), regulatory requirements

Where are we are today?

- Compute/storage is treated independent of network, no energy efficiency considerations
- Service value is not maximized

Source: ETSI NFV Use Cases
**Benefits**

- Use WAN bandwidth as needed, avoid fixed cost due to reservation (typically 1.5 times peak) – typically leverage MPLS technologies
- Popular use cases - Disaster Recovery, On-demand backup across WAN
Where do we want to get to?

- Beyond WAN Bandwidth savings
- Optimal resource placement across DCs - Increase Energy efficiency while maintaining multi-tenant fairness and improving performance – CAPEX/OPEX savings, Improve QoE, Address regulatory requirements
- Popular use cases - Disaster Recovery, On-demand backup across WAN, Virtualized CDN
NFVIaaS (IaaS+NaaS)

Energy Efficiency Issues

- Power usage in DCs → servers → heavy hitter
- Server power profiles typically non-linear; ~45% of peak power with ~20% of offered load; ~30% power in idle state
- Inefficient to keep servers powered on under low load conditions

<table>
<thead>
<tr>
<th>Target Load</th>
<th>Actual Load</th>
<th>$SSJ_{ops}$</th>
<th>Average Active Power (W)</th>
<th>Performance to Power Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>99.4%</td>
<td>1,217,238</td>
<td>324</td>
<td>3.752</td>
</tr>
<tr>
<td>90%</td>
<td>89.8%</td>
<td>1,099,468</td>
<td>311</td>
<td>3.539</td>
</tr>
<tr>
<td>80%</td>
<td>79.4%</td>
<td>971,386</td>
<td>293</td>
<td>3.319</td>
</tr>
<tr>
<td>70%</td>
<td>70.0%</td>
<td>856,359</td>
<td>275</td>
<td>3.108</td>
</tr>
<tr>
<td>60%</td>
<td>60.3%</td>
<td>738,030</td>
<td>245</td>
<td>3.010</td>
</tr>
<tr>
<td>50%</td>
<td>50.0%</td>
<td>611,914</td>
<td>202</td>
<td>3.026</td>
</tr>
<tr>
<td>40%</td>
<td>39.9%</td>
<td>487,913</td>
<td>174</td>
<td>2.797</td>
</tr>
<tr>
<td>30%</td>
<td>30.0%</td>
<td>367,489</td>
<td>152</td>
<td>2.415</td>
</tr>
<tr>
<td>20%</td>
<td>20.1%</td>
<td>245,999</td>
<td>138</td>
<td>1.784</td>
</tr>
<tr>
<td>10%</td>
<td>10.0%</td>
<td>121,941</td>
<td>128</td>
<td>0.949</td>
</tr>
<tr>
<td>Active Idle</td>
<td>0.4%</td>
<td>105</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

SPEC Benchmark results: HP ProLiant DL380p rack server

NFV – Huge opportunity for Openstack

How do we get there?

Energy aware joint scheduling of compute/storage/networking resources – example below

- NFV Customer submits job request, e.g. backup, with elasticity windows
- NFV Provider returns back information about time window to schedule backup
- Trigger other events e.g. Consolidate workloads; Finish one job and start the next; Power down resources (especially servers) after job completion

Our Solution: Smart Scheduler in Openstack

Adapted from ETSI NFV Architectural Framework
Smart Scheduling in OpenStack for Optimized NFV Resource Placements

Our Solution Smart Scheduler in Openstack
• Use analytics to determine current state of the Openstack deployment.
• Use resource management techniques to pick resources based on business constraints

Users:
Minimize costs… (Energy & Network Efficiency)
Maximize Performance...

Infrastructure:
State (BigData?)
(Storage/Network/Compute state, Energy Profiles, Policy/constraints etc.)
Candidate Solution: Unified Constraints-based Scheduling

A Smart Resource Placement Engine

- Unified constraints involving network, storage, compute, energy, etc.
- Global state + analytics
- Blazing fast implementations via Apache licensed third-party Solver libraries

Sources:
- https://docs.google.com/document/d/1liPI0sfawb1bdYiMWzAAx0HYR6UgqOan_Utgml5W1HI/edit
- https://github.com/CiscoSystems/nova-solver-scheduler
Solver Scheduler: Smart Scheduling in OS

**Users:**
- Minimize costs… (Energy & Network Efficiency)
- Maximize Performance...

**Cost Functions** → **Plug in** → **Intelligent Placement Engine** → **Plug in** → **Constraint Functions**

**Scheduling Decision**

**Infrastructure:**
- Server State...
- Energy Profiles...
- Network Link Capacities...
- System Capacity...

**Sources:**
- [https://docs.google.com/document/d/1IlPl0sfaWb1bdYiMWzAAx0HYR6UgzoAn_Utgml5W1HI/edit](https://docs.google.com/document/d/1IlPl0sfaWb1bdYiMWzAAx0HYR6UgzoAn_Utgml5W1HI/edit)
- [https://github.com/CiscoSystems/nova-solver-scheduler](https://github.com/CiscoSystems/nova-solver-scheduler)
An Example LP Problem Formulation

Consider there are 2 hosts $H_1$ and $H_2$, and two VM requests $V_1$ and $V_2$, and let us consider the constraint resources of usable disk (mb) and usable memory (mb) and use the free ram (mb) as a cost metric. The Host Supply Matrix $S$, the VM demand matrix $D$, the variables matrix $X$, and the cost matrix $C$ are as follows:

$$S = \begin{pmatrix} 2048 & 2048 \\ 4096 & 1536 \end{pmatrix} \quad D = \begin{pmatrix} 1024 & 512 \\ 1024 & 512 \end{pmatrix} \quad C = \begin{pmatrix} 2048 & 1536 \\ 2048 & 1536 \end{pmatrix} \quad X = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix}$$

The mathematical LP formulation of this example is as follows:

$$\text{Minimize}(2048 \cdot x_{11} + 2048 \cdot x_{21} + 1536 \cdot x_{12} + 1536 \cdot x_{22})$$

subject to constraints:

- $x_{11} \cdot 1024 + x_{21} \cdot 1024 \leq 2048$ (disk maximum supply for $H_1$)
- $x_{11} \cdot 512 + x_{21} \cdot 512 \leq 2048$ (memory maximum supply for $H_1$)
- $x_{12} \cdot 1024 + x_{22} \cdot 1024 \leq 4096$ (disk maximum supply for $H_2$)
- $x_{12} \cdot 512 + x_{22} \cdot 512 \leq 1536$ (memory maximum supply for $H_2$)
- $x_{11} \cdot 1024 + x_{12} \cdot 1024 \geq 1024$ (disk minimum demand for $V_1$)
- $x_{11} \cdot 512 + x_{12} \cdot 512 \geq 512$ (memory minimum demand for $V_1$)
- $x_{21} \cdot 1024 + x_{22} \cdot 1024 \geq 1024$ (disk minimum demand for $V_2$)
- $x_{21} \cdot 512 + x_{22} \cdot 512 \geq 512$ (memory minimum demand for $V_2$)

$x_{11} + x_{12} = 1$ ($V_1$ can run in only 1 Host)

$x_{21} + x_{22} = 1$ ($V_2$ can run in only 1 Host)

$x_{11} = 0, x_{12} = 1, x_{21} = 0, and x_{22} = 1$ happens to be the optimal solution, implying, both VMs $V_1$ and $V_2$ will be hosted in $H_2$. 

Scheduling can be Complex
DEMO: Smart Scheduling for NFV Service VMs with Compute/Storage Affinity Constraints

Applicable Scenarios:

1. CDN NFV Service VMs that need data on certain storage volumes, on physical servers that are on or closest to the data.
2. Backup NFV Service VMs placement.

Multinode devstack setup:

- Host-1: (Controller, Compute node)
- Host-2: (Compute node with demo_vol_1 Volume)
- Host-3: (Compute node with demo_vol_2 Volume)

Boot 2 VMs specifying the requested volumes to be close in proximity

Results: Optimal placement by picking the two physical volume hosts: Host-2 and Host-3.
Demo: Smart Scheduling with Compute-Storage Affinity
Conclusion

• NFV Value Proposition
  • NVF is a killer use-case for Openstack

• Call for community action
  • Scheduler Gap and a candidate solution [e.g. SolverScheduler, blueprint exists, code pushed for review in Icehouse]
  • Cross-Scheduler API w. constraints [e.g. augment server-groups API released in Icehouse]
  • Neutron hooks for Virtual Network Services (and API)