VNF Benchmark-as-a-Service (VBaaS): A Teaser
(draft-rorosz-nfvrn-vbaas-00.txt)

Rosa, Raphael V.†‡
Rothenberg, Christian E.‡
Szabo, Robert†

‡FEEC/UNICAMP and †Ericsson Research Hungary

NFVRG, IETF 94
11/4/2015
**VNF Benchmarking**

- Different NFVI PoP/host may perform differently
- VNF development and NFVI upgrades are independent
- Orchestration needs to know how much resources (e.g., cpu, memory, storage) to allocate for given target KPI values (e.g., throughput, latency).

![Diagram showing the relationship between customers, developers, VNFs, SAPs, NFVO/VNFM, VIMs, and PoPs.](image)
VNF Benchmarking

- Different NFVI PoP/host may perform differently
- VNF development and NFVI upgrades are independent
- Orchestration needs to know how much resources (e.g., cpu, memory, storage) to allocate for given target KPI values (e.g., throughput, latency).

Problem to be solved:
- Gain information *autonomously* about VNFs’ benchmark metrics with given reserved resources at a “host” (execution environment).
Components

- VBaaS service function
- VBaaS Information Base for VNF
- Benchmark Profiles
  - structural
  - functional: manager, monitors and agents

Work-flows

- for ETSI NFVO and VIMs
- for recursive orchestration
VNF Benchmarking as a Service (VBaaS) aims at
- defining complementary functional components to ETSI NFV and other approaches;
- defining interfaces to the VBaaS service;
- defining possible VBaaS work-flows.

Acknowledgements
- This work is partially supported by FP7 UNIFY, a research project partially funded by the European Community under the Seventh Framework Program (grant agreement no. 619609). The views expressed here are those of the authors only. The European Commission is not liable for any use that may be made of the information in this document.
- This work is partially supported by Ericsson Brazil.
Outline

1. Teaser

2. Backup slides
How? VNF Benchmarking as a Service (VBaaS)

VBaaS objectives:
- "Black box" VNF benchmarking, with respect to the NVFO Benchmarking Service.

VBaaS uses NFVO (or VIMs) to deploy benchmark measurements, e.g., by VNF-FG.
How? VNF Benchmarking as a Service (VBaaS)

VBaaS objectives

- “Black box” VNF benchmarking, with respect to the
  - NVFO
  - Benchmarking Service
VBaaS objectives

- “Black box” VNF benchmarking, with respect to the
  - NVFO
  - Benchmarking Service
- VBaaS uses NFVO (or VIMs) to deploy benchmark measurements
  - e.g., by VNF-FG
VBaaS Process Walk-through

[VNF1: {10Mbps, 200ms}]
[VNF-FG, Metrics: {VCPU, mem} -> {BW, delay}]
[VNF1: {10Mbps, 200ms}]
[VNF2: ...]

{VNF1: {10Mbps, 200ms} [{2CPU, 8GB}@PoP1] [{4CPU, 4GB}@PoP3]} {20Mbps, 300ms}...

VNF Profiles

NFVI PoPs

VNF 1 VNF 2

VNF-FG

Customers

NFVO / VNFM

VIMs

Manager

Agents, Monitors and SUT VNF
Usage: Benchmarking, Dimensioning and Verification

Benchmarking
To measure VNF’s throughput, latency, jitter parameters for given cpu, memory, storage reservation at given VIM.
Usage: Benchmarking, Dimensioning and Verification

**Benchmarking**
To measure VNF’s throughput, latency, jitter parameters for given cpu, memory, storage reservation at given VIM.

**Dimensioning**
To determine cpu, memory, storage reservation parameters for given VNF at given VIM for target throughput, latency, jitter parameters.
## Usage: Benchmarking, Dimensioning and Verification

### Benchmarking
To measure VNF’s throughput, latency, jitter parameters for given cpu, memory, storage reservation at given VIM.

### Dimensioning
To determine cpu, memory, storage reservation parameters for given VNF at given VIM for target throughput, latency, jitter parameters.

### Verification
To assess if given throughput, latency, jitter parameters of a VNF is met with given cpu, memory, storage reservation at given VIM.
Usage: Benchmarking, Dimensioning and Verification

**Benchmarking**
To measure VNF’s throughput, latency, jitter parameters for given cpu, memory, storage reservation at given VIM.

**Dimensioning**
To determine cpu, memory, storage reservation parameters for given VNF at given VIM for target throughput, latency, jitter parameters.

**Verification**
To assess if given throughput, latency, jitter parameters of a VNF is met with given cpu, memory, storage reservation at given VIM.

**Observation**
Dimensioning and verification boil down to benchmarking operation(s).
Recursive Orchestration with VBaaS

UNIFYing carrier network and cloud resources

- Recurrent joint software and networking control API\(^a\)

\(^a\)draft-unify-nfvr-g-recursive-programming-02
Recursive Orchestration with VBaaS

**UNIFYing carrier network and cloud resources**

- Recurrent joint software and networking control API
- Flexibility in resource virtualization with Big Switch with Big Software

---


draft-unify-nfvr-g-recursive-programming-02
Recursive Orchestration with VBaaS

UNIFYing carrier network and cloud resources

- Recurrent joint software and networking control API
- Flexibility in resource virtualization with Big Switch with Big Software

\[\text{^a draft-unify-nfvg-recursive-programming-02}\]

VBaaS

is part of each orchestration component; options:
Recursive Orchestration with VBaaS

UNIFYing carrier network and cloud resources

- Recurrent joint software and networking control API
- Flexibility in resource virtualization with Big Switch with Big Software

*a draft-unify-nfvrg-recursive-programming-02

VBaaS

is part of each orchestration component; options:
- request VNF profiling from the virtualization provider;
Recursive Orchestration with VBaaS

UNIFYing carrier network and cloud resources

- Recurrent joint software and networking control API\(^a\)
- Flexibility in resource virtualization with Big Switch with Big Software

\(^a\)draft-unify-nfvg-recursion-programming-02

VBaaS

is part of each orchestration component; options:

- request VNF profiling from the virtualization provider;
- do it on your own “transparently” over the underlying substrates.
VBaaS Request and Reporting as Capabilities

**Capability reporting**

- Based on a virtualization Yang model, each execution environment can report VNF benchmarking results.

```
+--------------+ VBaaS +--------------+
|Orchestration | API-1 |Orchestration |
 | (NFVO) |<---------->| (NFVO) |
 | Consumer) | | Producer) |
+------+-------+ +------+-------+

1) get-config()
2) rpc-reply(virt)
3) edit-config(NF1@cap)
4) rpc-ok
5) <notification>
6) get-config(NF1@cap)
7) rpc-reply(NF1@cap)
```
### VBaaS Request and Reporting as Capabilities

#### Capability reporting
- Based on a virtualization yang model, each execution environment can report VNF benchmarking results.
- Consumer can request a capability report by defining partial capability entries:
  - Providing only performance KPI means a dimensioning request
  - Providing only resource allocation means a benchmarking request
  - Providing both mean verification

```plaintext
+------------------------+ VBaaS +------------------------+
| Orchestration | API-1 | Orchestration |
| (NFVO |<---------->| (NFVO |
| Consumer) | | Producer) |
+------------------------+ +------------------------+
 1) get-config() |
 2) rpc-reply(virt) |
 3) edit-config(NF1@cap) |
 4) rpc-ok |
 5) <notification> |
 6) get-config(NF1@cap) |
 7) rpc-reply(NF1@cap) |
```
Acknowledgements

- This work is partially supported by FP7 UNIFY, a research project partially funded by the European Community under the Seventh Framework Program (grant agreement no. 619609). The views expressed here are those of the authors only. The European Commission is not liable for any use that may be made of the information in this document.

- This work is partially supported by Ericsson Brazil.