INTRIG INFORMATION & NETWORKING TECHNOLOGIES RESEARCH & INNOVATION GROUP

A VNF Testing Framework
Design, Implementation and Partial Results

NFVRG - IETF 97

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

November 14, 2016

FEEC/UNICAMP and Ericsson Research Hungary

FEEC Faculdade de Engenharia Elétrica e de Computação UNICAMP

ERICSSON

Motivation

- New paradigms of network services envisioned by NFV bring VNFs as software based entities, which can be deployed in virtualized environments

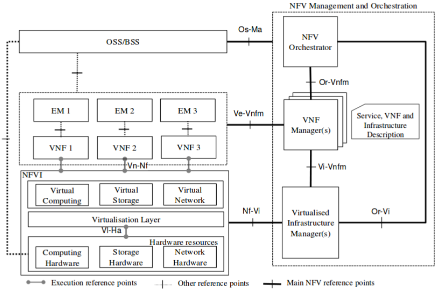
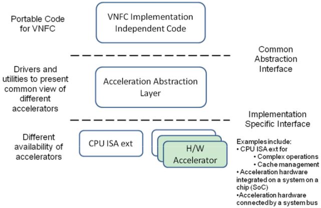


Figure: NFV Architectural Framework

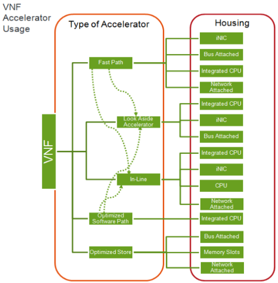
Motivation

- Virtualized environment (e.g., NFVI PoP) changes frequently in different places (e.g., platforms, hardware acceleration)


Portable Code for VNFC
VNFC Implementation Independent Code
Common Abstraction Interface
Drivers and utilities to present common view of different accelerators
Acceleration Abstraction Layer
Implementation Specific Interface
Different availability of accelerators
CPU ISA ext
H/W Accelerator

Examples include:
- CPU ISA ext for
  - Complex operations
  - Cache management
- Acceleration hardware integrated on a system on a chip (SoC)
- Acceleration hardware connected by a system bus

Figure: Use of acceleration abstraction layer (AAL) to enable fully portable VNFC code across servers with different accelerators



All accelerators shown are in the NFVI

Figure: VNF Usage of Accelerators

http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/001/01.01.01_60/gs_NFV-IFA001v010101p.pdf

Motivation

- VNFs need continuous development/integration
- VNF Descriptors can specify performance profiles containing metrics (e.g., throughput) associated with allocated resources (e.g., vCPU)

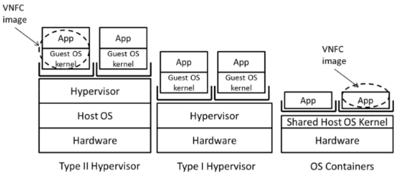
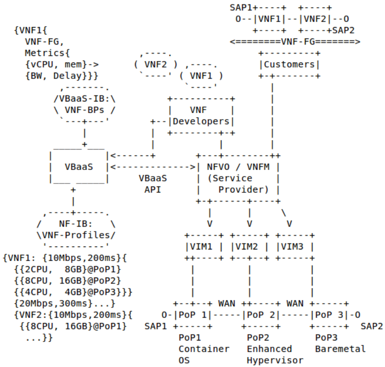


Figure: VNF Environment Examples

http://www.etsi.org/deliver/etsi_gs/NFV-EVE/001_099/004/01.01.01_60/gs_NFV-EVE004v010101p.pdf

Motivation

- Process for VNF metrics extraction can be automated: VBaaS - https://datatracker.ietf.org/doc/draft-rorosz-nfvrg-vbaas/



```
                                        SAP1+----+  +----+
                                         O--|VNF1|--|VNF2|--O
  {VNF1{                                    +----+  +----+SAP2
    VNF-FG,                             <========VNF-FG=======>
    Metrics{            ,----.               +---------+
    {vCPU, mem}->      ( VNF2 ) ,----.       |Customers|
    {BW, Delay}}}       `----' ( VNF1 )      +-+-------+
          ,-------.                 `----'         |
         /VBaaS-IB:\         +----------+          |
         \ VNF-BPs /         |   VNF    |          |
          `---+---'       +--|Developers|          |
              |           |  +--------+-+          |
         _____+___        |           |            |
        |         |<------+       +---+--------++
        |  VBaaS  |<------------->| NFVO / VNFM |
        |___ _____|     VBaaS     | (Service    |
            +            API      |   Provider) |
            |                     +-+------+----+
       ,----+-----.                 |      |     \
      /   NF-IB:   \                V      V      V
      \VNF-Profiles/            +-----+ +-----+ +-----+
       '----------'             |VIM1 | |VIM2 | |VIM3 |
{VNF1: {10Mbps,200ms}{          ++----+ +--+--+ +-----+
  {{2CPU,  8GB}@PoP1}            |         |          |
  {{8CPU, 16GB}@PoP2}            |         |          |
  {{4CPU,  4GB}@PoP3}}}          |         |          |
  {20Mbps,300ms}...}          +--+--+ WAN ++----+ WAN +-----+
  {VNF2:{10Mbps,200ms}{     O-|PoP 1|-----|PoP 2|-----|PoP 3|-O
   {{8CPU, 16GB}@PoP1}   SAP1 +-----+     +-----+     +-----+  SAP2
    ...}}                      PoP1        PoP2        PoP3
                               Container   Enhanced    Baremetal
                               OS          Hypervisor
```

Figure: NFV MANO and VBaaS

VBaaS Generic Workflow

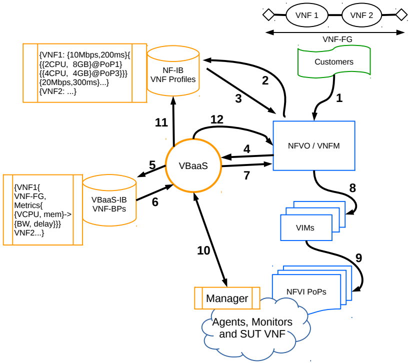


Figure: VBaaS Workflow

VBaaS Components

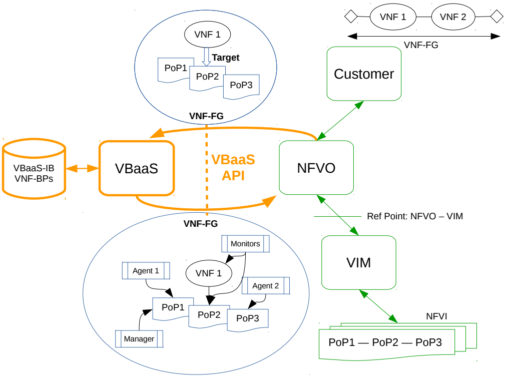


Figure: VBaaS API and Components

VBaaS Core Interactions

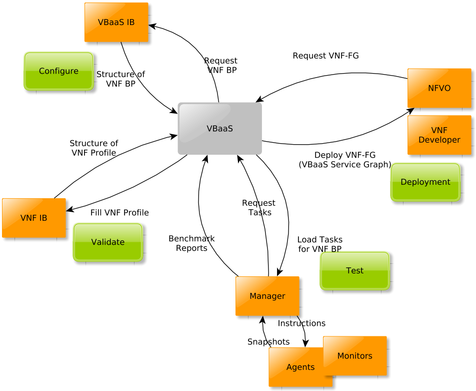


Figure: VBaaS Interactions

Request/Response Messages

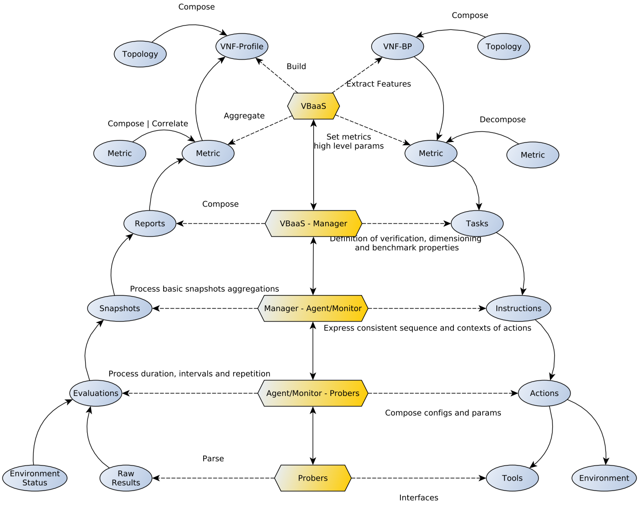
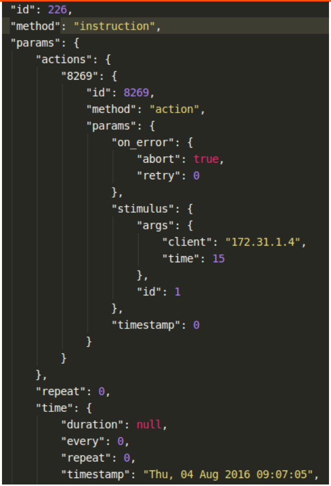


Figure: Request/Response Messages - JSON RPC

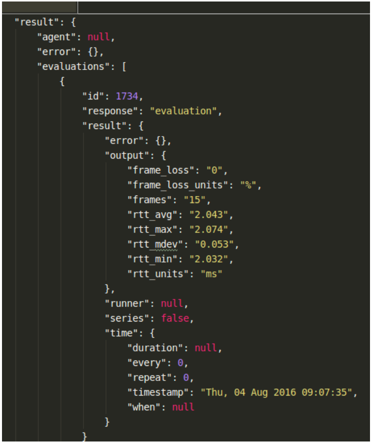
Message Example: Instruction



```
"id": 226,
"method": "instruction",
"params": {
    "actions": {
        "8269": {
            "id": 8269,
            "method": "action",
            "params": {
                "on_error": {
                    "abort": true,
                    "retry": 0
                },
                "stimulus": {
                    "args": {
                        "client": "172.31.1.4",
                        "time": 15
                    },
                    "id": 1
                },
                "timestamp": 0
            }
        }
    },
    "repeat": 0,
    "time": {
        "duration": null,
        "every": 0,
        "repeat": 0,
        "timestamp": "Thu, 04 Aug 2016 09:07:05",
```

Figure: Request: Instruction of ping prober

Message Example: Snapshot



```
"result": {
    "agent": null,
    "error": {},
    "evaluations": [
        {
            "id": 1734,
            "response": "evaluation",
            "result": {
                "error": {},
                "output": {
                    "frame_loss": "0",
                    "frame_loss_units": "%",
                    "frames": "15",
                    "rtt_avg": "2.043",
                    "rtt_max": "2.074",
                    "rtt_mdev": "0.053",
                    "rtt_min": "2.032",
                    "rtt_units": "ms"
                },
                "runner": null,
                "series": false,
                "time": {
                    "duration": null,
                    "every": 0,
                    "repeat": 0,
                    "timestamp": "Thu, 04 Aug 2016 09:07:35",
                    "when": null
                }
            }
```

Figure: Response: Snapshot of ping prober

VBaaS Prototyping - Ongoing work

- All independent components coded in python
- Defined by microservices - each component with stand-alone REST API
- Agents/Monitors with pluggable probers/listeners - easy to attach/install/load new tools
- Simple Workflows - Interactions among components - JSON RPC

Ongoing Analysis - Three Use Cases

Software Switch

Target: l2 traffic

vIMS

Target: decomposed compents (containers/VMs)

L4/L7 Custom VNF

Target: internal instrumentation

Related Work - Prototypes

- OPNFV Yardstick
- ToDD

Related work - IETF RFCs and drafts

Approach

- Information about Benchmarking Methodology for Network Interconnect Devices (RFC2544)
- IP Performance Metrics (IPPM) Framework (RFC2330)
- BMWG draft: VNF Benchmarking Methodology Considerations
- BMWG draft: Benchmarking Virtual Switches in OPNFV
- Initial Work Proposal - (draft-rosa-bmwg-vnfbench-00.txt)

NFVRG Intersection

VNF Performance Modeling

- Gain information about how to extract VNFs' performance metrics with given reserved resources at given VIM (NFVI PoP).

Analytics for Visibility and Orchestration

- Orchestration (e.g., NFVO) needs to know throughput, latency, among other metrics, performance values for a given resource allocation (cpu, memory, storage) of a VNF at a VIM.

Summary

Done

1. Prototype framework on VNF Testing - based on initial VBaaS proposal
2. Input for experiments and assist standardization

Next Steps

- Academic Publication - Analysis of Framework and Use Cases
- Open Sourcing
- Back to BMWG - VNF Benchmarking Methodology

This work is supported by Ericsson Research Brazil
This work is 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

Thanks!

Questions?