Recursive Monitoring Language in Network Function Virtualization (NFV) Infrastructure

draft-cai-nfvrg-recursive-monitor-00

Xuejun Cai
Catalin Meirosu
Gregory Mirsky

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement no. 619609 - the UNIFY project. 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

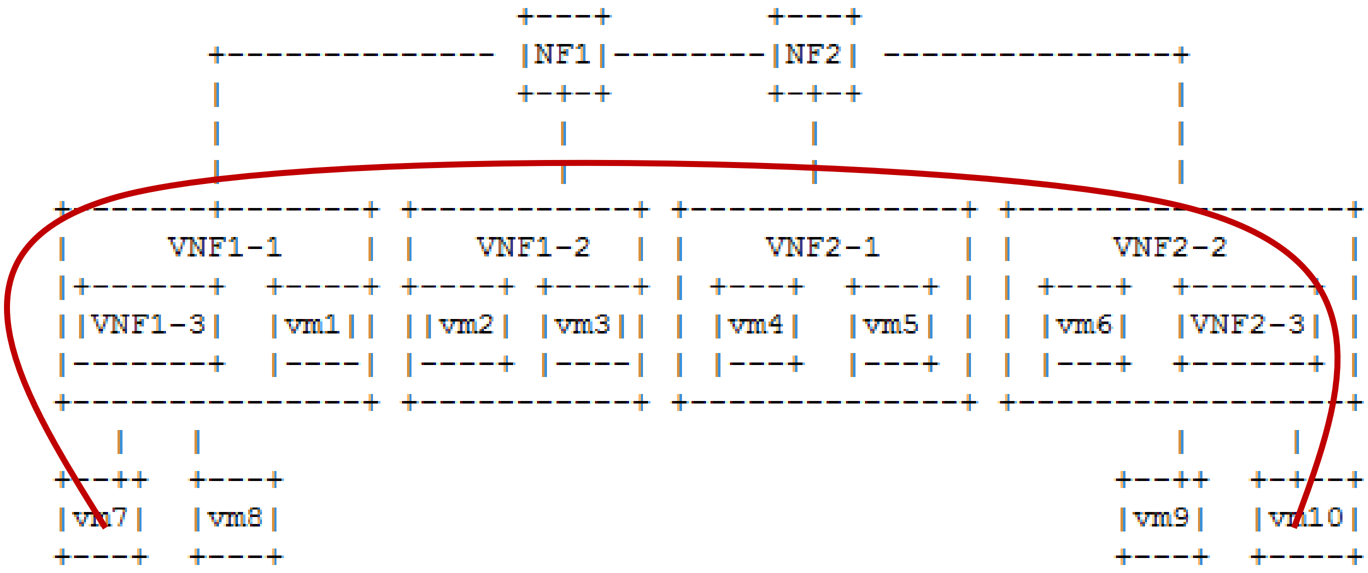
NFVRG interim meeting, Heidelberg, Dec 1st, 2015

UNIFY is co-funded by the European Commission
DG CONNECT in FP7

Motivation

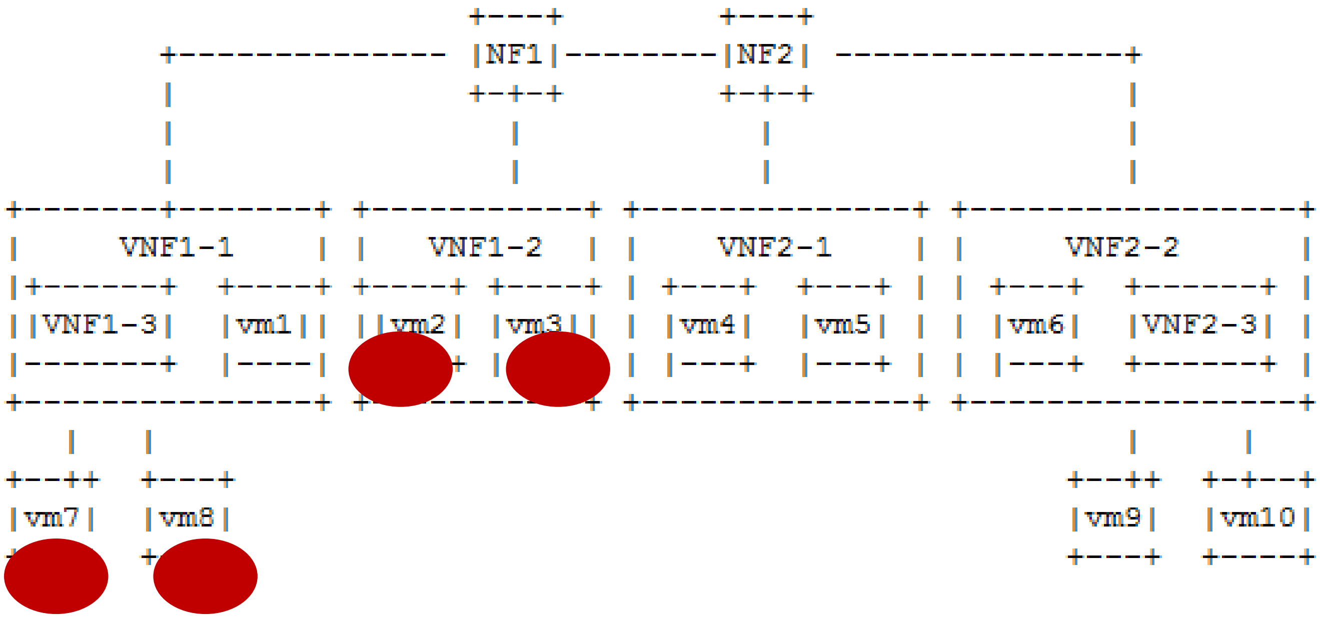
- provide an automatic way to decompose/aggregate monitoring data in different infrastructure layers
- provide a way for developers and operators to easily access monitoring data collected from resources in a software-defined telecom infrastructure that contains a hierarchy of abstraction levels
  - several NFVRG drafts describe such infrastructure

Example – e2e delay



```
                        +---+        +---+
        +-------------- |NF1|--------|NF2| ---------------+
        |               +-+-+        +-+-+                |
        |                 |            |                  |
        |                 |            |                  |
+-------+-------+ +-----------+ +---------------+ +--------+--------+
|     VNF1-1    | |  VNF1-2   | |    VNF2-1     | |     VNF2-2      |
|+-------+  +---+ +----+ +----+ | +---+   +---+ | | +---+  +-------+ |
||VNF1-3|   |vm1|| |vm2| |vm3|| | |vm4|   |vm5| | | |vm6|  |VNF2-3| |
|--------+  |---| |----+ |----| | |---+   |---+ | | |---+  +-------+ |
+---------------+ +-----------+ +---------------+ +-----------------+
   |   |                                                  |     |
+--++ +---+                                            +--++ +-+--+
|vm7| |vm8|                                            |vm9| |vm10|
+---+ +---+                                            +---+ +-----+
```

Example – aggregated CPU usage



Datalog – brief intro

- Subset of Prolog
- A program consists of declarative rules and a query
  - Rules: h <= p1, p2, ..., pn
    - p(x1, ..., xi, ..., xn)" are either predicates applied to arguments "xi" (variables and constants), or function symbols applied to arguments
  - Queries: q (m, y1, ..., yn)“
    - "q" is a predicate, contains arguments "m" (a function) and "yi" (arguments for that function)

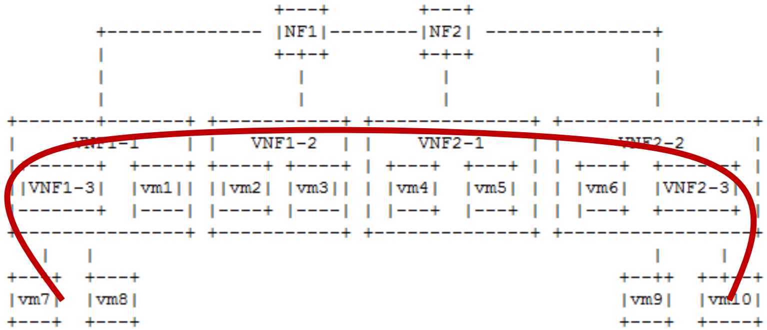
Simple NF-FG representation

- sub(x, y): 'y' is an element of the directly descend sub-layer of 'x';
- link(x, y): direct link between elements 'x' and 'y';
- node(z): node in NF-FG
- The NF-FG representation is “ground facts” for Datalog

Requirements for a query engine

- MUST provide the capability to parse and interpret the query scripts which are written with the language
- MUST be able to retrieve the NF-FG created by NFV infrastructure and translate them into Datalog based ground facts
- MUST be able to query the database in which the monitoring results of primitive metric are stored
- An interface between query engine and the users of the language (e.g., developer or network service operator) MUST be defined to exchange the query scripts and query results.

Example – e2e delay



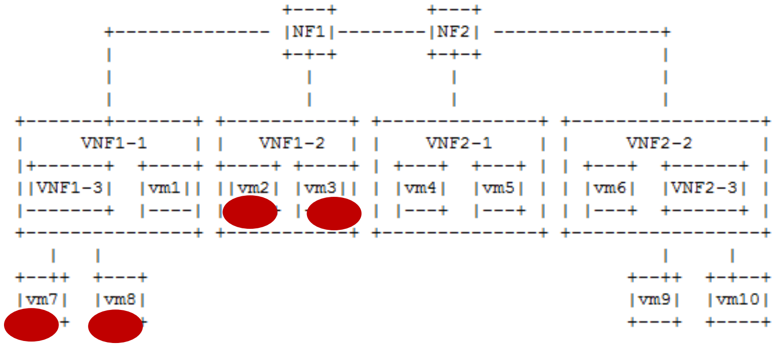
Ground rules

```
F1: sub(NF1, VNF1-3, vm1, vm2, vm3), sub(NF2, vm4, vm5, vm6, VNF1-3),
sub(VNF1-3, vm7, vm8), sub(VNF1-3, vm9, vm10)
F2: node(NF1, NF2, VNF1-3, vm1, vm2, vm3, vm4, vm5, vm6, VNF1-3,
vm7, vm8, vm9, vm10)
F3: link(NF1, NF2), link (VNF1-3, vm1), link(vm2, vm3), link(vm3, vm4),
link(vm4,vm5), link(vm5,vm6), link(vm6, VNF1-3), link(vm7, vm8),
link(vm9, vm10)
R1: child(X,Y) <= sub(X,Z), child(Z,Y)
R2: child(X,Y) <= sub(X,Y)
R3: leaf(X,Y) <= child(X,Y), ~sub(Y,Z)
R4: in_leaf(X, Y) <= leaf(X, Y) & ~link(M, Y)
R5: out_leaf(X, Y) <= leaf(X, Y) & ~link(Y, M)
R6: e2e_delay(S,D,P) <= link(S,D), P == f_e2e_delay(in_leaf(S,Y),
out_leaf(D,Z))
query(e2e_delay, NF1, NF2)
```

Rule template(s)

Query

Example – e2e delay



Ground rules

```
F1: sub(NF1, VNF1-3, vm1, vm2, vm3), sub(NF2, vm4, vm5, vm6, VNF1-3),
sub(VNF1-3, vm7, vm8), sub(VNF1-3, vm9, vm10)
F2: node(NF1, NF2, VNF1-3, vm1, vm2, vm3, vm4, vm5, vm6, VNF1-3, vm7,
vm8, vm9, vm10)
R1: child(X,Y) <= sub(X,Z), child(Z,Y)
R2: child(X,Y) <= sub(X,Y)
R3: leaf(X,Y) <= child(X,Y), ~sub(Y,Z)
R4: max_cpu(X,C) <= leaf(X,Y), C == f_max_cpu(leaf(X,Y))
R5: mean_cpu (X,C) <= leaf(X, Y), C == f_mean_cpu(leaf(X,Y))
Query(max_cpu, NF1)
```

Rule template(s)

Query

Conclusion

- We presented a proposal on using a Datalog-derived language for automatically aggregating monitoring data in NFV environments
- Next steps:
  - Receive feedback from the community
  - Provide additional templates
  - Enhance the NF-FG description to align with NFVRG drafts evolution