RECURSIVE MONITORING LANGUAGE IN NETWORK FUNCTION VIRTUALIZATION (NFV) INFRASTRUCTURE

DRAFT-CAI-NFVRG-RECURSIVE-MONITOR-00

NFVRG @ IETF 94 YOKOHAMA

Xuejun Cai
Catalin Meirosu
Gregory Mirsky
OVERVIEW

› Motivation:
  – provides 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

› Solution proposal:
  – Define a query language based on an extended Datalog syntax
  – Include pre-defined templates for initial metrics examples
EXAMPLE

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(X, Y)
R5: out_leaf(X, Y) <= leaf(X, Y) & ~link(Y, M)
R6: e2e_delay(S, D, F) <= link(S, D), F = f_e2e_delay(in_leaf(S, Y), out_leaf(D, Y))

query(e2e_delay, NF1, NF2)

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)