CUSCO: A Customizable Solution for NFV Composition

Vinicius Fulber-Garcia, Marcelo C. Luizelli,
Carlos R. P. dos Santos, Elias P. Duarte Júnior

Summary

- **Introduction**
- **Related Work**
- **CUSCO: CUstomazable Service COmposing**
- **Case Study**
  - Setup
  - Experimentation
  - Results
- **Final Remarks**

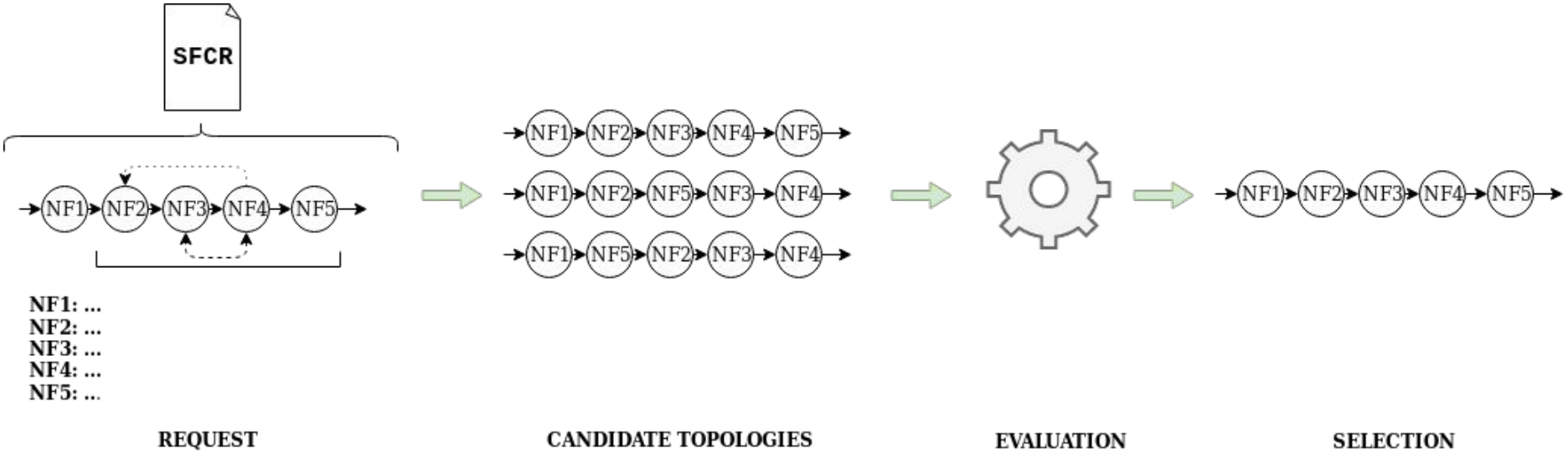
Introduction

- **Current Network Infrastructure**
  - Physical appliances
  - Core network ossification
- **Network Function Virtualization (NFV)**
  - Decoupling of network functions from their associated hardware
  - Current virtualization techniques
  - Network service and service topology
- **Network Service Deployment**
  - Acquisition, preparation and operationalization
  - Inter-related stages

Introduction

- **NFV Resource Allocation (NFV-RA)**
  - “Main part of the service deployment”
  - Constituted of three tasks
    - **Composition**
    - Embedding
    - Schedule
- **Composition**
  - Absolute positioning of network functions in a service topology
  - Operational model (service topology / relationship graphs)
  - Dependencies and policies
  - **Objective functions**
  - Service Function Chaining Request (SFCR)

Introduction



Related Work

- **Partially Ordered Service Topologies**
  - Solution by Mehraghdam
    - Traffic ratio
  - Solution by Draxler
    - Traffic ratio, Resources usage, topology size
- **Network Functions Relationship Graph**
  - Solution by Ocampo
    - Bandwidth requirements
  - Solution by Gil
    - Bandwidth requirements
  - Solution by Wang
    - Network function priority level

Service Composition

Problems?

SERVICE TOPOLOGY DESCRIPTION

- ➔ Generic branching structures support
- ➔ Network function dependencies

TOPOLOGY ANALYSIS

- ➔ Branching structures manipulation

OBJECTIVE FUNCTION

- ➔ **Static metrics**
- ➔ **Non-customizable processment**

CUSCO: CUstomizable Service COmposing

Network services composition solution that enables the user to configure and customize the entire composition process.

SCAG: Service ChAin Grammar

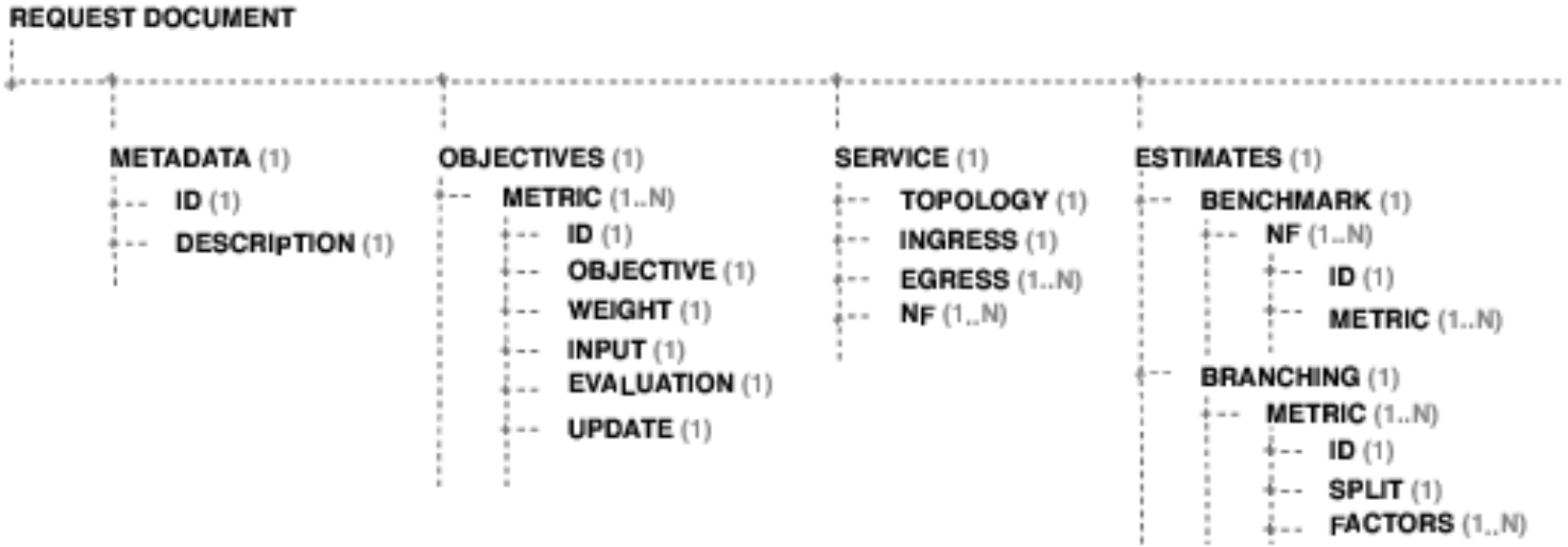
- **Service Topology Specification Model**
- **Context-free Grammar**
- **Support of:**
  - Partially ordered segments
  - Network function dependencies (ordering and coupling)
  - Infrastructure dependencies (domains)
  - Generic branching structures (terminal and non-terminal)

```
1  START → 'IN' MAIN
2  MAIN → TBRANCHING | NTBRANCHING |
       OPERATIONAL MAIN | OPERATIONAL EN
3  NTBMAIN → INTBRANCHING | OPERATIONAL
       NTBMAIN | OPERATIONAL
4  OPERATIONAL → PORDER | FUNCTION
5  PORDER → '[' FUNCTION NFUNCTION ']'
       EDEPENDENCY | '[' FUNCTION NFUNCTION ']'
6  EDEPENDENCY → EORDERING | ECOUPLING
7  EORDERING → '(' FUNCTION FUNCTION ')'
       EDEPENDENCY | '(' FUNCTION FUNCTION ')'
8  ECOUPLING → '(' FUNCTION FUNCTION '*' ')'
       EDEPENDENCY | '(' FUNCTION FUNCTION '*'
       ')'
9  TBRANCHING → OPERATIONAL '{' MAIN TBRANCH '
       }'
10 TBRANCH → '/' MAIN TBRANCH | '/' MAIN
11 NTBRANCHING → OPERATIONAL '{' NTBMAIN
       NTBRANCH '}' MAIN
12 NTBRANCH → '/' NTBMAIN NTBRANCH | '/'
       NTBMAIN
13 INTBRANCHING → OPERATIONAL '{' NTBMAIN
       NTBRANCH '}' NTBMAIN
14 FUNCTION → VNF | VNF ADDEPENDENCY
15 NFUNCTION → FUNCTION NFUNCTION | FUNCTION
16 ADDEPENDENCY → '<' ADMDOMAIN '>'
17 VNF → 'VNF#1', 'VNF#2', ..., 'VNF#n'
18 ADMDOMAIN → 'AD#1', 'AD#2', ..., 'AD#n'
19 EN → 'EN#1', 'EN#2', ..., 'EN#n'
```

Figura 1. SCAG Production Rules

YAMLR: YAML Request

- **Extensible Request Model**



CUSCO: CUstomizable Service COmposing

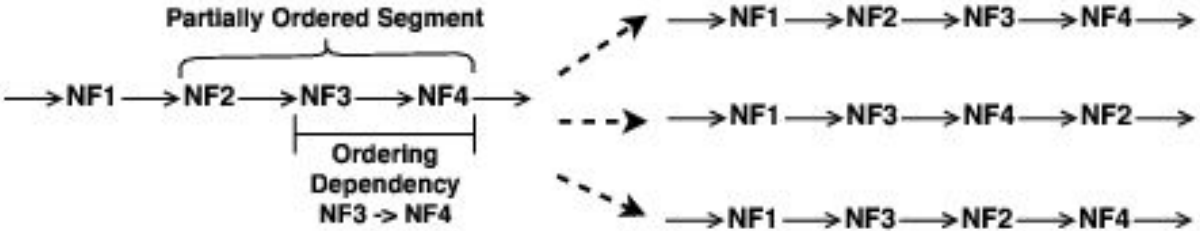
- **Configuration Flexibility**
  - Turn on/off some specific steps executed by the solution
- **Evaluation Flexibility**
  - Multi-criteria evaluation
    - Multiple metrics with different granularities
    - Evaluation metrics defined by the users
  - Weighting
    - Each metric has a particular weight defined by the user
  - Indexing
    - A single value is returned (Suitability Index - SI)
- **Two procedures**
  - (I) Topologies expansion; (II) Topologies evaluation

CUSCO: CUstomizable Service COmposing

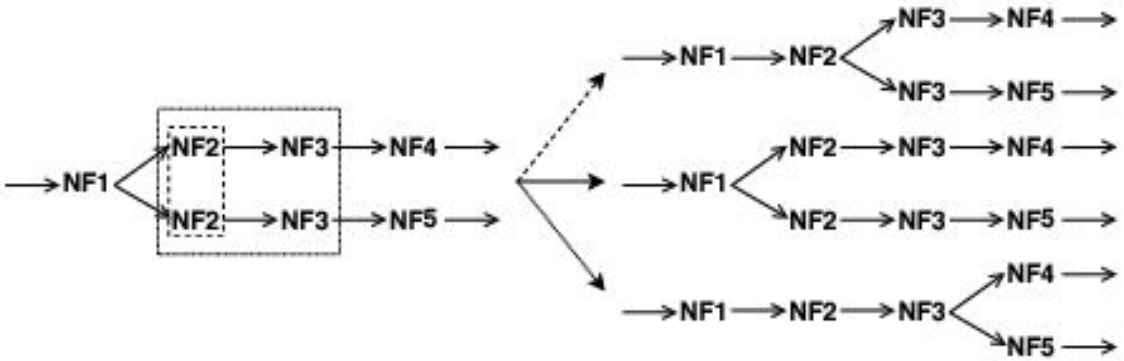
- **Topology Expansion**
  - **Proc. #1**: partial ordering permutations
  - **Proc. #2**: branching structures remodelling
- **Partial Ordering Permutations**
  - Permutation with constraints
    - Permutation -> Partially ordered segments
    - Constraints -> NF dependencies (ordering and coupling)
  - Exhaustive -> Generates all possible permutations
- **Branching Structures Remodelling**
  - Identical segments in every branch of the same branching structure
  - Reduction to single instance in a common segment of the service topology:
    - Initial segment (terminal and non-terminal branching structures)
    - Final segment (non-terminal branching structures)

CUSCO: CUstomizable Service COmposing

TE-P1)



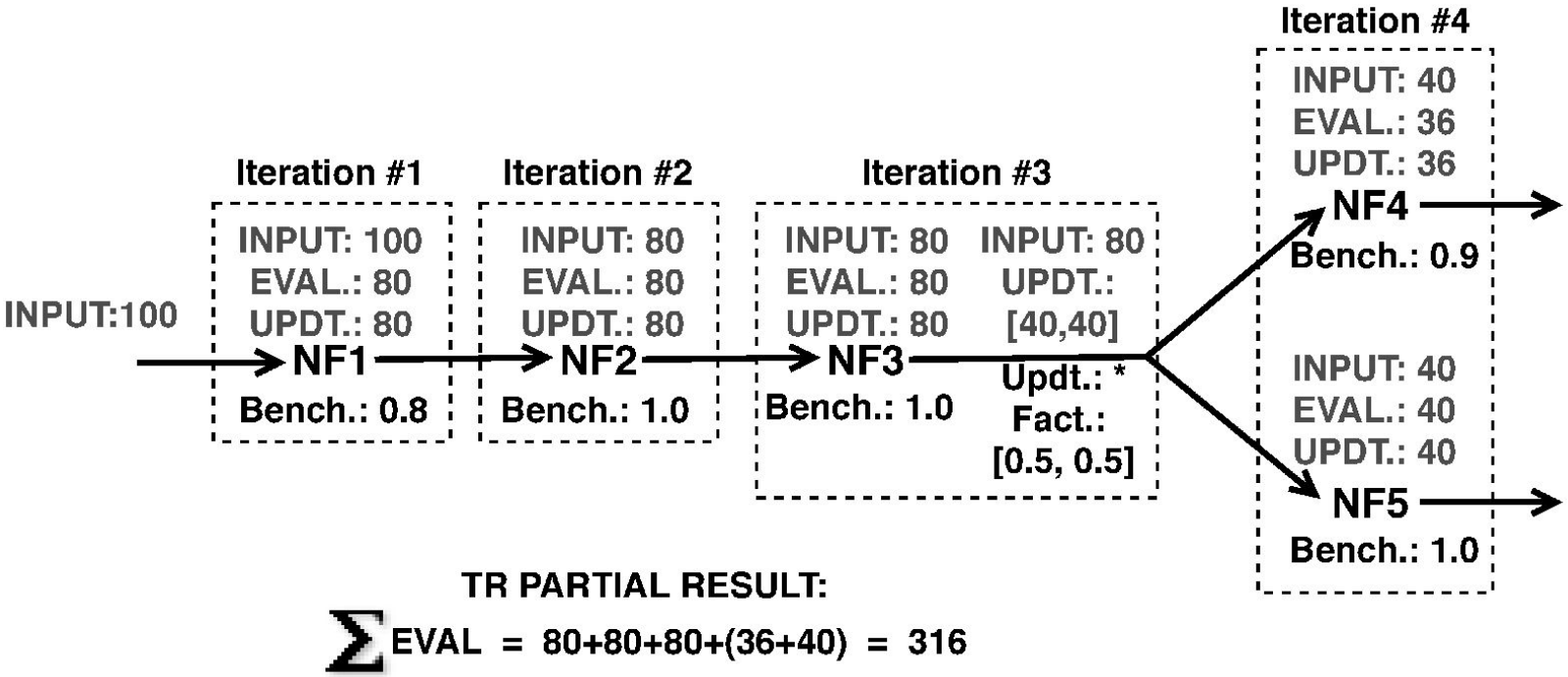
TE-P2)



CUSCO: CUstomizable Service COmposing

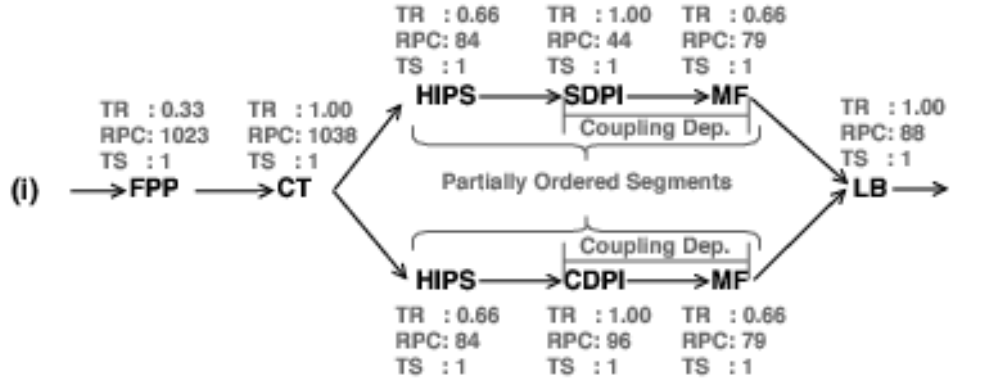
- **Topologies Evaluation**
  - **Proc. #1**: partial results generation
  - **Proc. #2**: candidates evaluation
- **Partial Results Generation**
  - Partial function
  - **Iteration** (partial function evaluation + input update)
  - Partial results
  - Mapping + Normalization + Complementation
- **Candidates Evaluation**
  - Weighting + Summing (= Suitability Index)
  - Ranking

CUSCO: CUstomizable Service COmposing



Case Study [Setup]

- **HTTP/S-based Network Service**
  - Filtering of ports and anomalous/suspicious packets
  - Inspection of signatures (HTTPS) and key-words (HTTP)
  - Load balancing between HTTP/S servers
- **Seven Network Functions**
  - Python 3
  - Click Modular Router
- **Objective Function**
  - Traffic Ratio (%)
  - HTTP/S request response ratio (req/s)
  - Topology size



Case Study [Experimentation]

- **Nine candidates returned from the topologies expansion**
  - All of them evaluated and ranked by their SI
- **Three candidates were selected to be deployed and tested**

#1: SI 0.666

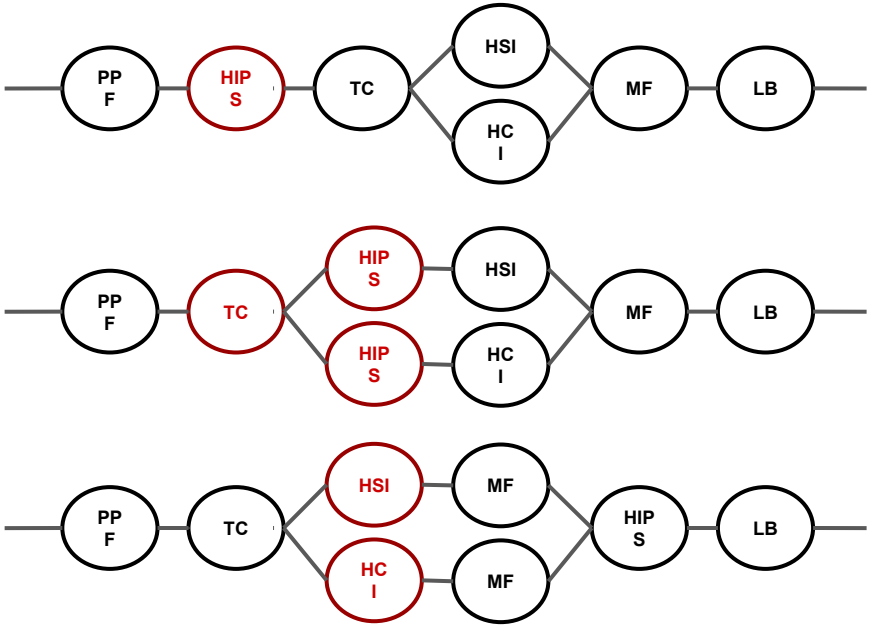
Best SI
Partial TR: 0.333
Partial RR: 0
Partial Size: 0.333

#2: SI 0.500

Best SI with maximum RR
Partial TR: 0.167
Partial RR: 0.333
Partial Size: 0

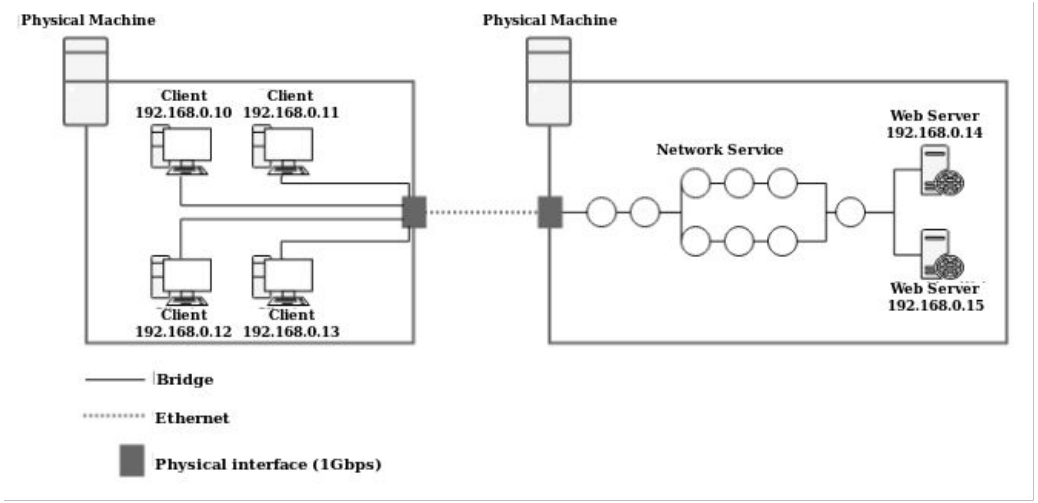
#3: SI 0.327

Worst SI
Partial TR: 0
Partial RR: 0.161
Partial Size: 0.166



Case Study [Experimentation]

- **Start:** simultaneous for ll clients
- **End:** conclusion of 5000 legitimate requests (each client)
- **Three Scenarios:**
  - Normal traffic
  - UDP overload
  - DDoS low rate attack



Case Study [Results]

	Maximization	Minimization	
	HTTP/S request response ratio (req/s)	Average Traffic Ratio (Mbps)	Topology Size and Computational Resource Usage
SI 0.666 **Best SI** Partial TR: 0.333 Partial RR: 0 Partial Size: 0.333	**Sce. #1:** 30.238 **Sce. #2:** 47.779 **Sce. #3:** 4.875 **Avg: 27.631**	**Sce. #1:** 2.002 **Sce. #2:** 20.645 **Sce. #3:** 37.929 **Avg: 20.192**	**7** **3584 MB RAM** **7 Virtual Cores**
SI 0.500 **Best SI with maximum RR** Partial TR: 0.167 Partial RR: 0.333 Partial Size: 0	**Sce. #1:** 28.839 **Sce. #2:** 46.534 **Sce. #3:** 33.462 **Avg: 36.278**	**Sce. #1:** 1.935 **Sce. #2:** 21.976 **Sce. #3:** 59.631 **Avg: 27.847**	9 4608 MB RAM 9 Virtual Cores
SI 0.327 **Worst SI** Partial TR: 0 Partial RR: 0.161 Partial Size: 0.166	**Sce. #1:** 29.381 **Sce. #2:** 47.226 **Sce. #3:** 0 **Avg: 25.535**	**Sce. #1:** 1.966 **Sce. #2:** 20.864 **Sce. #3:** * **Avg: ***	8 4096 MB RAM 8 Virtual Cores

Final Remarks

- **The Construction of Topologies Was Correct**
  - All the generated topologies provide the same service
- **The Composition Method Works Correctly**
  - Conciliated multiple metrics with different granularities
  - Weighted the partial results
  - Generated the suitability index and the ranking
- **Future Work**
  - Generalization of the composition method
    - Composition, embedding and Scheduling
  - Integration to marketplaces/providers
    - “Deployment-as-a-Service”

CUSCO: A Customizable Solution for NFV Composition

Thanks!!

Vinicius Fulber-Garcia
vfgarcia@inf.ufpr.br

https://github.com/ViniGarcia/ViNeFuR
https://github.com/ViniGarcia/NFV-FLERAS

Appendix - Topologies Evaluation

	Objective	Weight	Candidate #1	Candidate #2
Metric 01	Maximization	0,3	100	30
Metric 02	Minimization	0,7	1	0,6

	Maximum Absolute Distance	Mapping		Normalization		Complementation		Weighting		**Index**	
		C. #1	C. #2	C. #1	C. #2	C. #1	C. #2	C. #1	C. #2	C. #1	C. #2
Mtc. 01											
Mtc. 02											

Appendix - Candidates Evaluation

	Candidate #1	Candidate #2
Metric 01	100	30
Metric 02	1	0,6

	Maximum Absolute Distance	Mapping		Normalization		Complementation		Weighting		**Index**	
		C. #1	C. #2	C. #1	C. #2	C. #1	C. #2	C. #1	C. #2	C. #1	C. #2
Mtc. 01											
Mtc. 02											

Appendix - Topologies Evaluation

	Candidate #1	Candidate #2
Metric 01	100	30
Metric 02	1	0,6

	Maximum Absolute Distance	Mapping		Normalization		Complementation		Weighting		**Index**	
		C. #1	C. #2	C. #1	C. #2	C. #1	C. #2	C. #1	C. #2	C. #1	C. #2
Mtc. 01	[0, 70]										
Mtc. 02	[0, 0,4]										

Appendix - Topologies Evaluation

	Maximum Absolute Distance	Mapping		Normalization		Complementation		Weighting		**Index**	
		C. #1	C. #2	C. #1	C. #2	C. #1	C. #2	C. #1	C. #2	C. #1	C. #2
Mtc. 01	[0, 70]	70	0								
Mtc. 02	[0, 0,4]	0,4	0								

Appendix - Topologies Evaluation

	Maximum Absolute Distance	Mapping		Normalization		Complementation		Weighting		**Index**	
		C. #1	C. #2	C. #1	C. #2	C. #1	C. #2	C. #1	C. #2	C. #1	C. #2
Mtc. 01	[0, 70]	70	0	1	0						
Mtc. 02	[0, 0,4]	0,4	0	1	0						

Appendix - Topologies Evaluation

	Weight
Metric 01	0,3
Metric 02	0,7

	Maximum Absolute Distance	Mapping		Normalization		Complementation		Weighting		**Index**	
		C. #1	C. #2	C. #1	C. #2	C. #1	C. #2	C. #1	C. #2	C. #1	C. #2
Mtc. 01	[0, 70]	70	0	1	0	1	0				
Mtc. 02	[0, 0,4]	0,4	0	1	0	0	1				

Appendix - Topologies Evaluation

	Maximum Absolute Distance	Mapping		Normalization		Complementation		Weighting		Index	
		C. #1	C. #2	C. #1	C. #2	C. #1	C. #2	C. #1	C. #2	C. #1	C. #2
Mtc. 01	[0, 70]	70	0	1	0	1	0	0,3	0		
Mtc. 02	[0, 0,4]	0,4	0	1	0	0	1	0	0,7		

Appendix - Topologies Evaluation

	Objective	Weight	Candidate #1	Candidate #2
Metric 01	Maximization	0,3	100	30
Metric 02	Minimization	0,7	1	0,6

	Maximum Absolute Distance	Mapping		Normalization		Complementation		Weighting		**Index**	
		C. #1	C. #2	C. #1	C. #2	C. #1	C. #2	C. #1	C. #2	C. #1	C. #2
Mtc. 01	[0, 70]	70	0	1	0	1	0	0,3	0	**0,3**	**0,7**
Mtc. 02	[0, 0,4]	0,4	0	1	0	0	1	0	0,7		

Appendix - Case Study [Experimentation]

Composição Candidata	Agregados	Normalização	Complementação	Ponderação	IAT	Função Objetivo
NE EO1 EO6 EO2 { EO3 / EO4 } EO5 EO7 NS	tt: 127,1 dtr: -1853,7 tam: 7	tt: 0 dtr: 1 tam: 0	tt: 1 dtr: 0 tam: 1	tt: 0,333 dtr: 0 tam: 0,333	0,666	1º
NE EO1 EO6 EO2 { EO3 EO5 / EO4 EO5 } EO7 NS	tt: 127,1 dtr: -1932,6 tam: 8	tt: 0 dtr: 0,517 tam: 0,5	tt: 1 dtr: 0,483 tam: 0,5	tt: 0,333 dtr: 0,161 tam: 0,166	0,660	2º
NE EO1 EO2 { EO6 EO3 / EO6 EO4 } EO5 EO7 NS	tt: 138,3 dtr: -1938,2 tam: 8	tt: 0,5 dtr: 0,483 tam: 0,5	tt: 0,5 dtr: 0,417 tam: 0,5	tt: 0,167 dtr: 0,172 tam: 0,166	0,505	3º
NE EO1 EO2 { EO6 EO3 EO5 / EO6 EO4 EO5 } EO7 NS	tt: 138,3 dtr: -2017,1 tam: 9	tt: 0,499 dtr: 0 tam: 1	tt: 0,501 dtr: 1 tam: 0	tt: 0,167 dtr: 0,333 tam: 0	0,500	4º
NE EO1 EO2 { EO3 EO5 EO6 / EO6 EO4 EO5 } EO7 NS	tt: 143,9 dtr: -2017,1 tam: 9	tt: 0,749 dtr: 0 tam: 1	tt: 0,251 dtr: 1 tam: 0	tt: 0,083 dtr: 0,333 tam: 0	0,416	5º
NE EO1 EO2 { EO6 EO3 EO5 / EO4 EO5 EO6 } EO7 NS	tt: 143,9 dtr: -2017,1 tam: 9	tt: 0,749 dtr: 0 tam: 1	tt: 0,251 dtr: 1 tam: 0	tt: 0,083 dtr: 0,333 tam: 0	0,416	5º
NE EO1 EO2 { EO3 EO5 EO6 / EO4 EO5 EO6 } EO7 NS	tt: 149,5 dtr: -2017,1 tam: 9	tt: 1 dtr: 0 tam: 1	tt: 0 dtr: 1 tam: 0	tt: 0 dtr: 0,333 tam: 0	0,333	6º
NE EO1 EO2 { EO3 / EO4 } EO5 EO6 EO7 NS	tt: 149,5 dtr: -1853,7 tam: 7	tt: 1 dtr: 1 tam: 0	tt: 0 dtr: 0 tam: 1	tt: 0 dtr: 0 tam: 0,333	0,333	6º
NE EO1 EO2 { EO3 EO5 / EO4 EO5 } EO6 EO7 NS	tt: 149,5 dtr: -1932,6 tam: 8	tt: 1 dtr: 0,517 tam: 0,5	tt: 0 dtr: 0,483 tam: 0,5	tt: 0 dtr: 0,161 tam: 0,166	0,327	7º

Appendix - Case Study [Experimentation]

- **Physical Machine #1 (Clients)**
  - Ubuntu 14.04
  - KVM hypervisor
  - 8 GB RAM DDR3
  - Core I3 4010U
- **Physical Machine #2 (Services)**
  - Debian 8
  - KVM Hypervisor
  - 8 GB RAM DDR3
  - Core I5 3330

Appendix - Case Study [Experimentation]

- **Tools**
  - HTTP-PERF (legitimate clients)
  - HPING3 (UDP traffic)
  - NPING (malicious clients)
- **DDoS Configuration**
  - Aims to overload the inspector functions
    - Thus making the network infrastructure unavailable
  - Low-rate (10 Mbps)
  - Anomalous packets of 1450 bytes
  - Never stops during the test