A Holistic Approach to Define Service Chains Using Click-on-OSv on Different NFV Platforms

Alexandre Huff[1,2], Giovanni Venâncio[2], Leonardo da C. Marcuzzo[3], Vinícius Garcia[3], Carlos R. P. dos Santos[3], Elias P. Duarte Jr[2].

[1] Federal Technological University of Paraná – UTFPR, Toledo, Brazil
[2] Federal University of Paraná – UFPR, Curitiba, Brazil
[3] Federal University of Santa Maria – UFSM, Santa Maria, Brazil

GLOBECOM'2018 Abu Dhabi, December 12, 2018

Agenda

- Network Function Virtualization
- Service Function Chaining
- A Holistic Approach do Define Service Chains
- Evaluation
- Conclusion

Network Function Virtualization

- Several network functionalities are implemented by a large variety of middleboxes
  - Specific purposes (e.g. firewall, load balancers, proxies, etc.)
  - Match hardware and software from the same manufacturer

- Network Function Virtualization (NFV) allows to replace services traditionally provided using middleboxes by software
  - Software can be executed by virtualization systems on Commercial-Off-The-Shelf (COTS) hardware
  - Decreases costs, increases flexibility to operate and manage network services

Network Function Virtualization

- Virtual Network Function (VNF)
  - Responsible to process a specific network traffic
  - Operates on different layers of the protocol stack

- The European Telecommunications Standards Institute (ETSI) has proposed an architecture for NFV Management & Orchestration (NFV-MANO)
  - Disseminate the use and interoperability of the NFV
  - It provides the functionality required for the provisioning of VNFs and its related operations

Service Function Chaining

- Complex network services can be formed by composing a set of network functions
- Service Function Chaining (SFC)
  - Consists of composing a set of VNFs and routing the traffic through them according to a predefined order (Service Chain)

- The Internet Engineering Task Force (IETF) has proposed an architecture for the SFC
  - SFC: Employs a flow identifier
    - MPLS (Multiprotocol Label Switching)
    - VXLAN (Virtual eXtensible Local Area Network)
    - NSH (Network Service Header)
  - Conventional Routing
    - Decisions are taken on destination IP address

Motivation

- The deploymet of service chains consisting of multiple network functions can be complex, expensive, and time-consuming
- Network operator performs a large number of tasks and services to accomplish service chain composition
  - Laborious task
  - Requires operational experience of the NFV Platform
  - Knowledge of specific data models, describing languages, and their requirements to compose SFCs
- Network operators can get overwhelmed with details of the NFV platform being used
  - Missing the focus on the service composition logic itself
- The efficient composition of service chains still lacks high-level procedures

A Holistic Approach to Define Service Chains

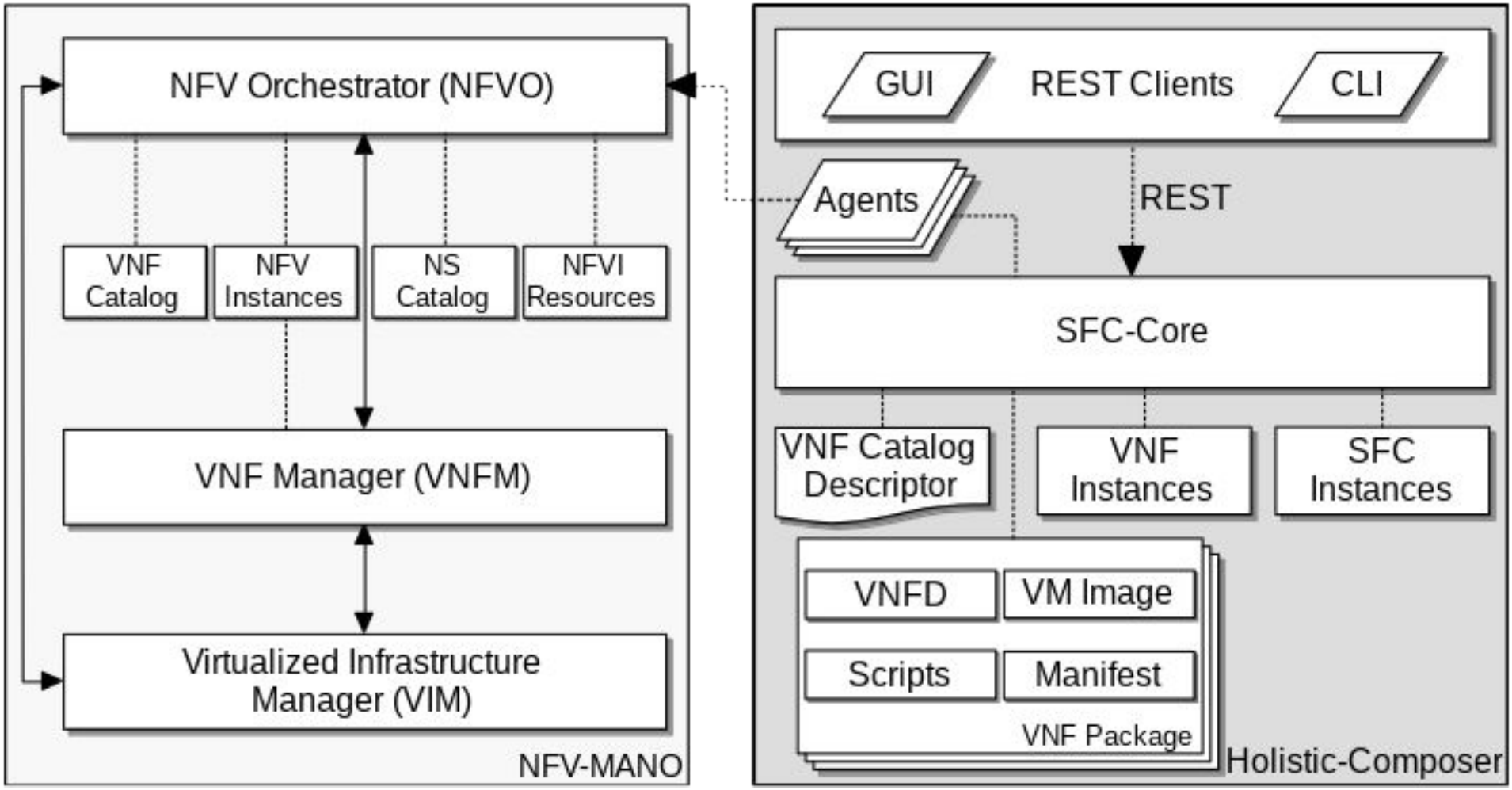
- This work proposes a solution for the definition and lifecycle management of VNF service chains
- A framework architecture is proposed as an extensible solution of the NFV-MANO

- Holistic
  - Defines a generic API for the composition of SFCs leveraging particular details of different NFV orchestrators

A Holistic Approach to Define Service Chains

- Main goals
  - Enable the composition and management of SFCs on different NFV Platforms
  - Allow the use of the framework on several final applications and on different NFV platforms which are MANO-compliant
  - Reduce the complexity of NFV platforms to define and manage service chains
    - Network operators do not need specific knowledge
  - Simplify the SFC composition
    - Network operator just informs the sequence of VNFs

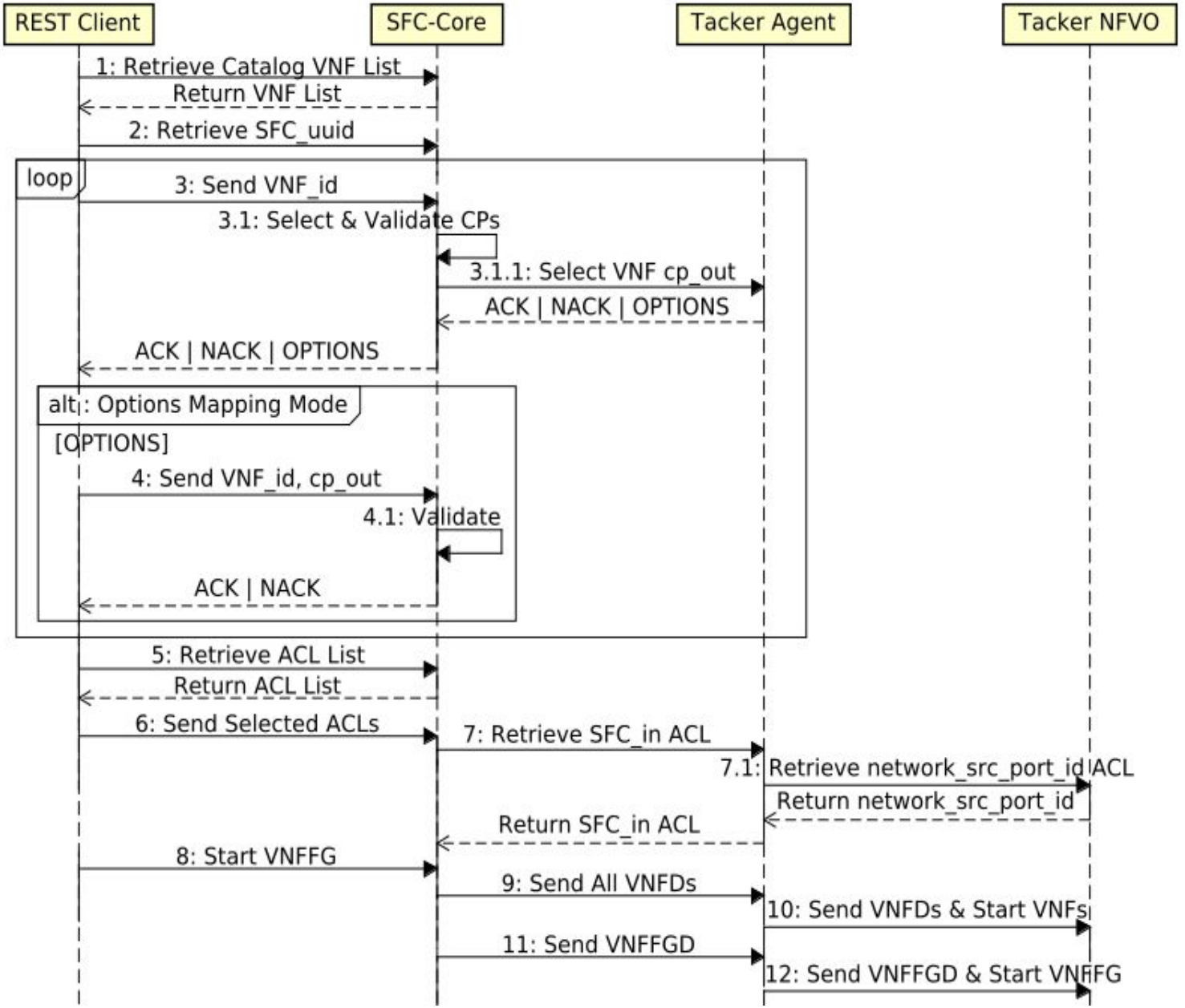
The Proposed Architecture



NFV Enablers

- OpenStack
  - Platform used to control Cloud Computing infrastructure resourses
  - NFV Context: Used in physical resource virtualization for VNFs and network services (VIM)
- Click-on-OSv
  - Enables the execution of network functions built with the Click Modular Router on the OSv unikernel
  - Allows creating high-performance and minimalist VNFs
  - Runs on: Xen, KVM, VMware and VirtualBox
- Tacker
  - Official OpenStack project to develop generic VNFM and NFVO
  - Based on the ETSI NFV-MANO architecture

Service Chain Composition



Prototype Implementation

- Architecture components implemented in Python
- *Flask* library to provide a REST interface
- *Requests* library
  - Communication Agent (Tacker)
  - Client Application (CLI)
- Implementation of an *Element Management* (EM) for the Click-on-OSv
  - Provides FCAPS (*Fault, Configuration, Accounting, Performance and Security*) functionalities through a REST API
- Prototype also allows composing SFCs using traditional VNFs (e.g. Linux)

Prototype Evaluation

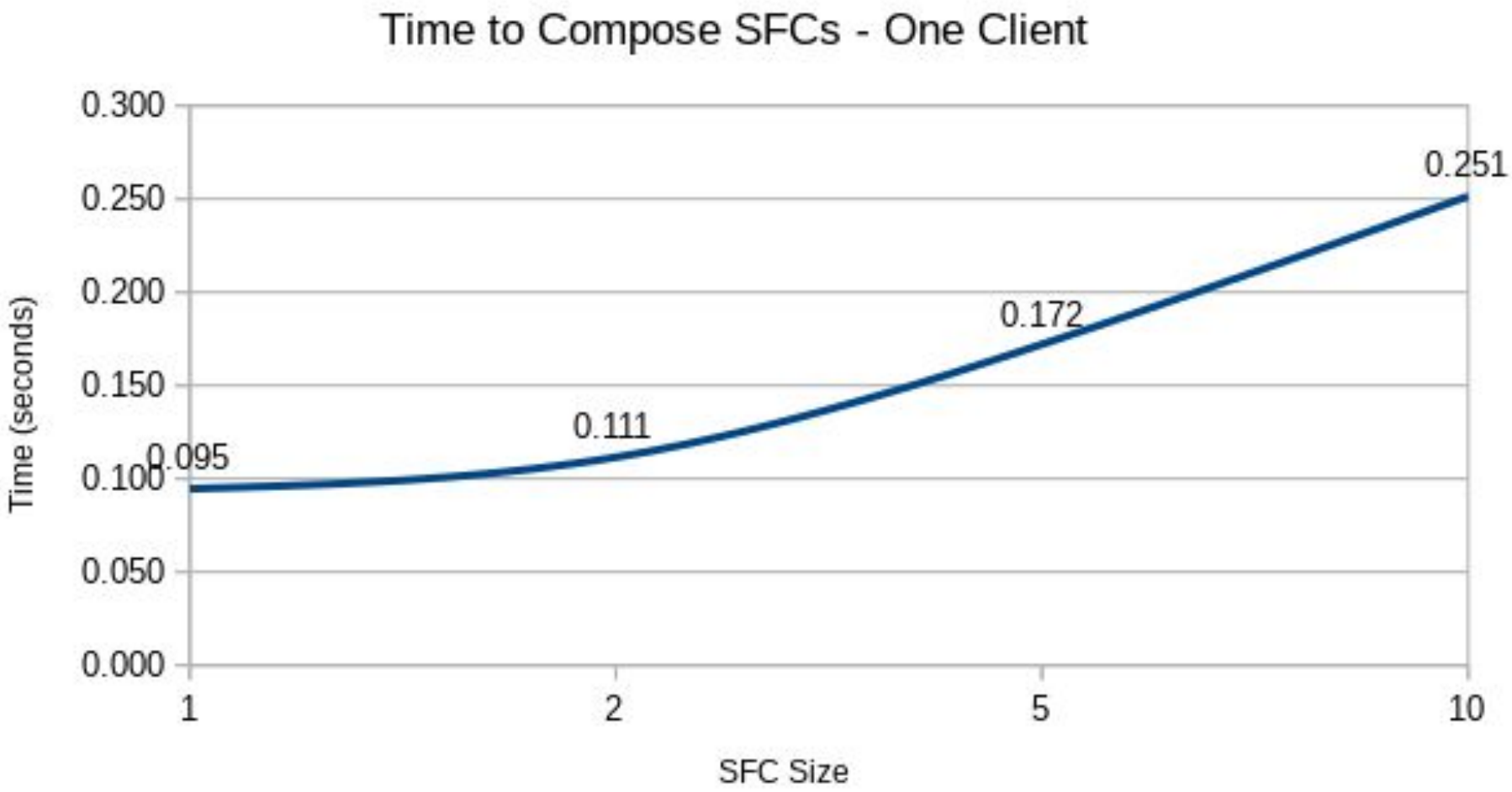
- We are particulary interested on the time it takes to compose SFCs in different execution scenarios
- The experiments were executed on a machine with:
  - Intel(R) Core(TM) i7-6700HQ CPU with 4 cores up to 3.5 GHz
  - 12 GiB of DDR4 RAM at 2133 MHz
  - Linux Ubuntu 16.04.3 running on kernel 4.13.0-37-generic x86_64
  - Apache HTTP Server 2.4.18 with the *mod_wsgi* adapter *libapache2-mod-wsgi-py3*
  - *memcached* daemon 1.4.25
  - MongoDB 2.6.10
- Apache was configured to have 4 processes listening for incoming requests to the SFC-Core

Prototype Evaluation

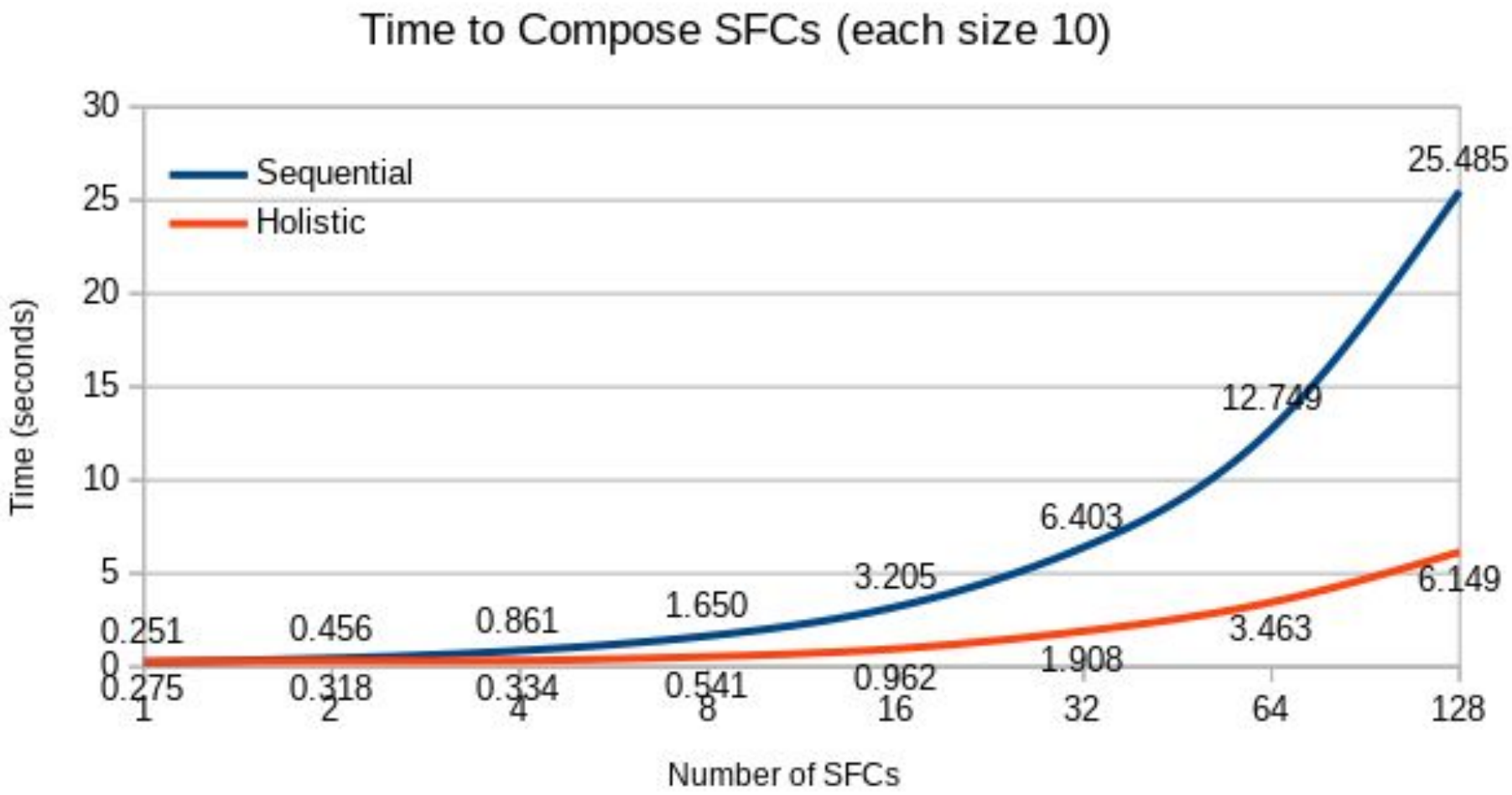
- A REST Client was implemented in order to run SFC composition experiments
- We highlight that:
  - The client application simply sends REST requests to the SFC-Core
  - The SFC-Core has the responsibility of the logic of composing and managing the SFC life cycle
  - The evaluation takes into account only the prototype execution times
- The elapsed time was recorded by the REST Client
  - At the time instants the SFC composition starts and completes

All experiments were performed 100 times, averages are

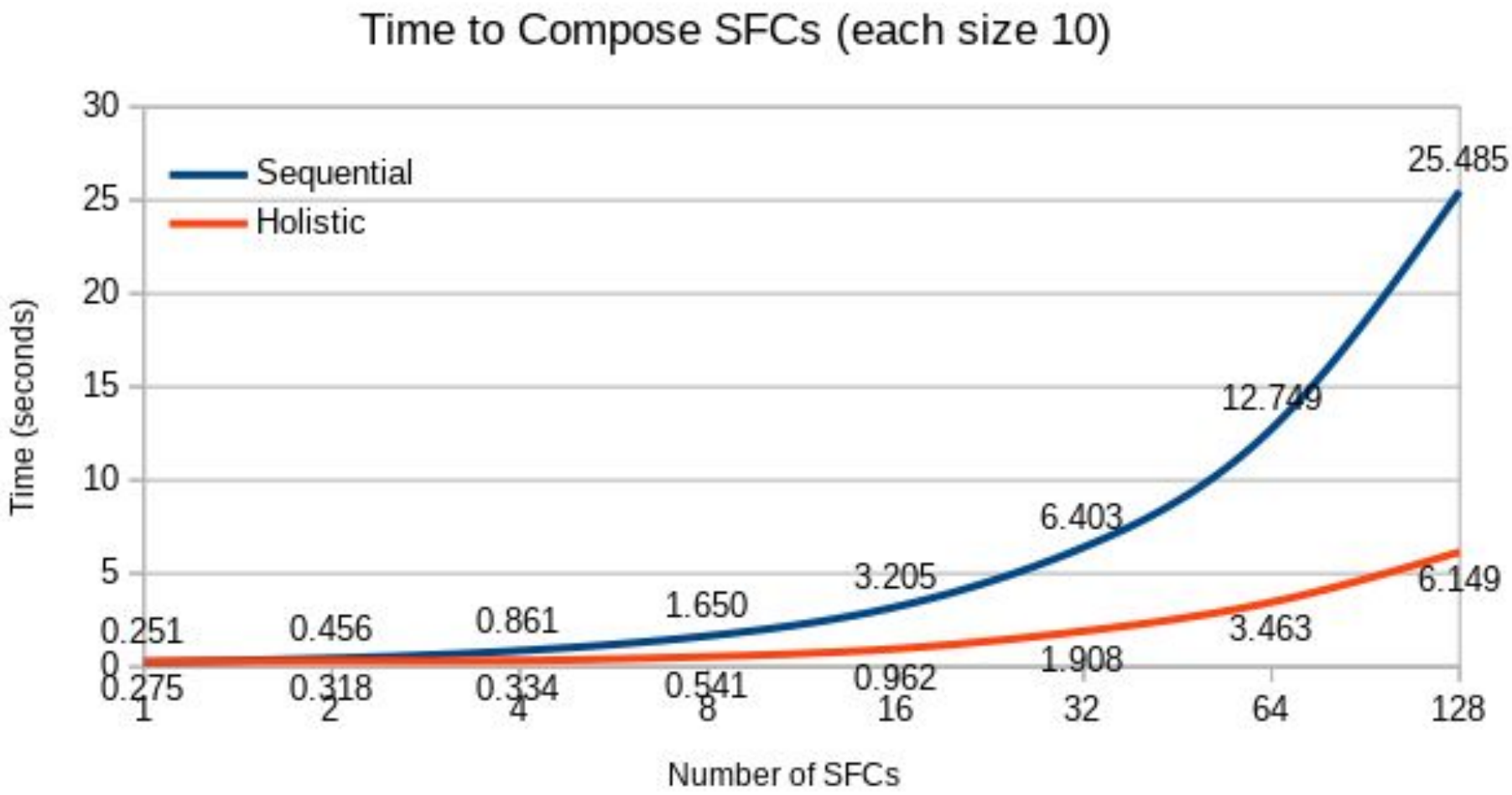
Time a single REST client takes to compose SFCs with sizes 1, 2, 5, and 10 VNFs



Time to compose different numbers of SFCs, each consisting of 10 VNFs



Time to compose different numbers of SFCs, each consisting of 10 VNFs



The Holistic approach is around 4.145 times faster than the Sequential approach

Conclusion

- The Holistic-Composer framework simplifies the composition and lifecycle management of SFCs on different NFV platforms
  - Defines generic operations for the composition and lifecycle management of SFCs
- Allows the integration with NFV enablers that do not provide interactive SFC composition functionality
- A prototype was implemented for the composition and lifecycle management of SFCs built with Click-on-OSv running on the OpenStack Tacker
  - Results were presented for multiple SFC compositions done concurrently

Future Work

- We are currently implementing a communication agent for the OSM Orchestrator
- Other orchestrators are also planned, specially those on the edge
- Extending the architecture to allow resource management, in particular SFC elasticity
- Automatic sharing of VNF instances on multiple SFCs

Thank you!

https://github.com/alexandre-huff/holistic-composer

e-mail: alexandrehuff@utfpr.edu.br
ahuff@inf.ufpr.br