Elastic Adaptation of SDN/NFV Systems to Dynamic Service Demands

Pedro Martinez-Julia
Network Science and Convergence Device Technology Laboratory, Network System Research Institute National Institute of Information and Communications Technology
pedro@nict.go.jp

IRTF/NFVRG, IETF 100 Meeting, Singapore
14 November 2017
Outline

- Motivation and Research Topic
- Proposed Approach and Architecture
- Alignment With ETSI-NFV-MANO
- Conclusions & Future Work
Motivation and Research Topic (I)

Trivia:

“High variation in resource demand” vs “Fixed resource allocation”.
Motivation and Research Topic (II)

Trivia:

“High variation in resource demand” vs “Fixed resource allocation”.

Answer:

Dynamic Environments -> Elastic Resource Adaptation
Use Case (I)

HQ Network

Computerized Help Desk
Use Case (II)

NFV Domain 2

NFV Domain 3

NFV Domain 1

NFV Domain 4

HQ Network

Computerized Help Desk
Use Case (III)

- Broken Link
- Overloaded Link
- Overloaded Server

NFV Domain 2

NFV Domain 3

NFV Domain 4

NFV Domain 1
Use Case (IV)

NFV domains built with OpenStack:
- Facilitates the **construction** of virtual computer and network systems.
  - It is widely used to create production-ready **virtualization environments**.
- Enables the **adaptation** of resources:
  - On-demand instantiation or removal of VMs attached to a service.
- Offers application **interfaces**:
  - Monitoring and resource adaptation.
- Provides good support for **NFV**.
- Its operation will be **enhanced** by the results of our research work.
Proposed Approach (I)

System Events
Other Events
Control Actions

ARCA Engine

SDN/NFV Network (D1)
Controller
SDN/NFV Network (D2)
Controller
SDN/NFV Network (D3)
Controller
SDN/NFV Network (D4)
Controller

Computerized Help Desk

© National Institute of Information and Communications Technology
Proposed Approach (II)

Autonomic Resource Control Architecture

- System Events
- Other Events
- Control Actions

Controller

Computerized Help Desk

SDN/NFV Network (D1)

SDN/NFV Network (D2)

SDN/NFV Network (D3)

SDN/NFV Network (D4)

Controller

ARCA Engine

Autonomic Resource Control Architecture

System Events

Other Events

Control Actions
Proposed Approach (III)

Collect observations from multiple sources:
- System elements:
  - Underlying controllers (SDN/NFV), VM monitors, …
- Environment:
  - External event detectors…

© National Institute of Information and Communications Technology
Proposed Approach (IV)

Analyze the observations to find out the specific situation of the system:
- Apply administrative policies and control statements to check resource state.
Adapt assigned resources:
- Set resource boundaries according to found situations.
- Set specific resource amount according to estimated demands.
- Issue actions to the underlying infrastructure (SDN/NFV controllers).
Delay less than 1 second:
- From external event occurrence to action enforcement.
- Support processing thousands of observations per second.
Challenges, Solutions, and Tools:

- **Too much observations** and volatility:
  - *Input filtering*:
    + Ensure information is not lost and underlying system is not overstressed.
- **Reduce delay**:
  - High performance controller.
  - Anticipate situations (*learning*).
- **Reliability**:
  - Continuous check of policies provided by administrators (statements).
Overview of ARCA (I)

Collector

Analysis Statements

Collector

{Observations: CPU load and sensor readings}

Administrator
{Human}

Decision Statements

KB & Reasoner

Analyzer

Closed-Loop

Decider

Actions

Enforcer

Resource Controllers

Resources, Controllers, Things

Overview of ARCA (I)
Overview of ARCA (II)

Exploits automation techniques to minimize human involvement:
- Address complex control and management operations.
- Reduce the time required for resource adaptation.

Resources, Controllers, Things
{Observations: CPU load and sensor readings}

Collector

Administrator {Human}

Decision Statements

KB & Reasoner

Core

Enforcer

Closed-Loop

Resource Controllers

CEP

Analyzer

Decider
Overview of ARCA (III)

Resources, Controllers, Things
{Observations: CPU load and sensor readings}

Administrators set **operational boundaries** for the target system:
- Lower and upper **amount of resources** that can be assigned.
- Lower and upper load **thresholds**.
Overview of ARCA (IV)

Includes the activities defined by Autonomic Computing (AC):
- Separate micro-services: Collector, Analyzer, Decider, Enforcer
- Closed-loop approach: Check effects of decisions afterwards.

Resources, Controllers, Things
{Observations: CPU load and sensor readings}
Overview of ARCA (V)

Exchanges and knowledge follow a common ontology:
- Encoded in RDF/Turtle and exploiting OWL.
- The ontology can be extended to support new concepts.
- Knowledge is stored in the Fuseki KB, supports SPARQL.

Resources, Controllers, Things
{Observations: CPU load and sensor readings}
Alignment With ETSI-NFV-MANO (I)

ARCA-based Engine
{Virtual Infrastructure Manager (VIM)}

Directed by “statements” (policies / rules)

Adapted to “underlying infrastructure providers”, enlarged with external event detectors
Alignment With ETSI-NFV-MANO (II)

- ARCA plays the role of the Virtualized Infrastructure Manager (VIM):
  - Provides autonomic capabilities to:
    - Discharge responsibilities from VNFM and NFVO.
    - Improve the scalability and resiliency of the system in case of disconnection from the orchestrator.
  - Is focused to accomplish requirements of Virtual Network Operators (VNOs).

- The **Nf-Vi** interface (IFA004, IFA019) in ARCA has been:
  - Bound to available underlying and overlying interfaces:
    - Ceilometer/Gnocchi provided by OpenStack.
  - Extended to enable interactions with external elements:
    - Physical / environmental event (incident) detectors.
    - Big Data: analyzers, data sources, etc.

- The **Or-Vi** interface (IFA005) is provided by the basic specification of control/management targets:
  - Its main communication artifact is the specification of control statements:
    - Represent the rules and policies that ARCA must enforce.
    - Will be provided by system administrators and/or external orchestrators.
  - ARCA will enforce such statements in response to changes in the environment and/or user requirements:
    - Requirements are communicated with additional statements.

- The **Vi-Vnfm** interface (IFA006) is currently out of the scope of ARCA.
Conclusions & Future Work

• Designed ARCA:
  − To provide functions of the Virtual Infrastructure Manager (VIM) of NFV-MANO.
  − Extended VIM interfaces to meet real requirements of the real world:
    • Emergency scenarios (!)
  − Achieved good performance within an OpenStack-based deployment:
    • Detailed overlying and underlying infrastructures.
    • Reproduction of production-like environments to ensure transferable research results.

• SDN/NFV and OpenStack stakeholders will benefit from ARCA features:
  − Efficient use of resources:
    • Further reduce CAPEX and OPEX:
  − Benefit to both infrastructure providers and consumers.

• Next steps:
  − Keep reducing ARCA response time.
  − Increase complexity of the validation scenario:
    • Mix clients and servants in the same domains.
  − Align ARCA-based VNC to additional requirements from NFV/SDN.
Thanks for Your Attention
- EOF -