Luigi De Simone

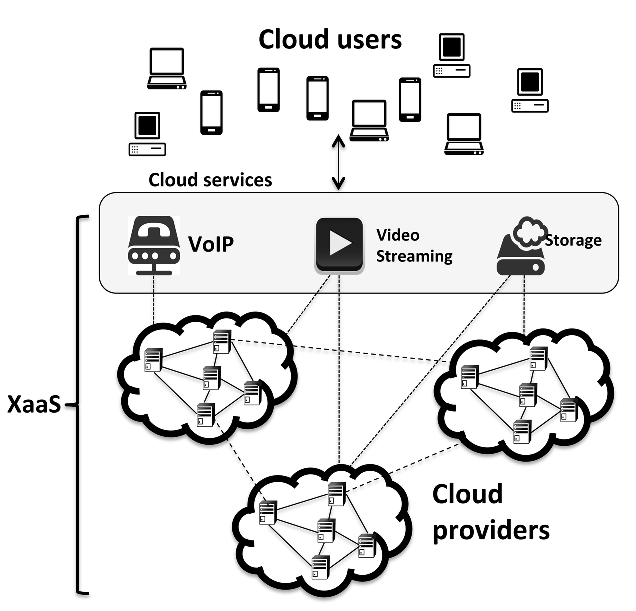
Tutor: Prof. Domenico Cotroneo

XXIX Cycle - II year presentation

Dependability Evaluation in Cloud Computing Ecosystems

Research context

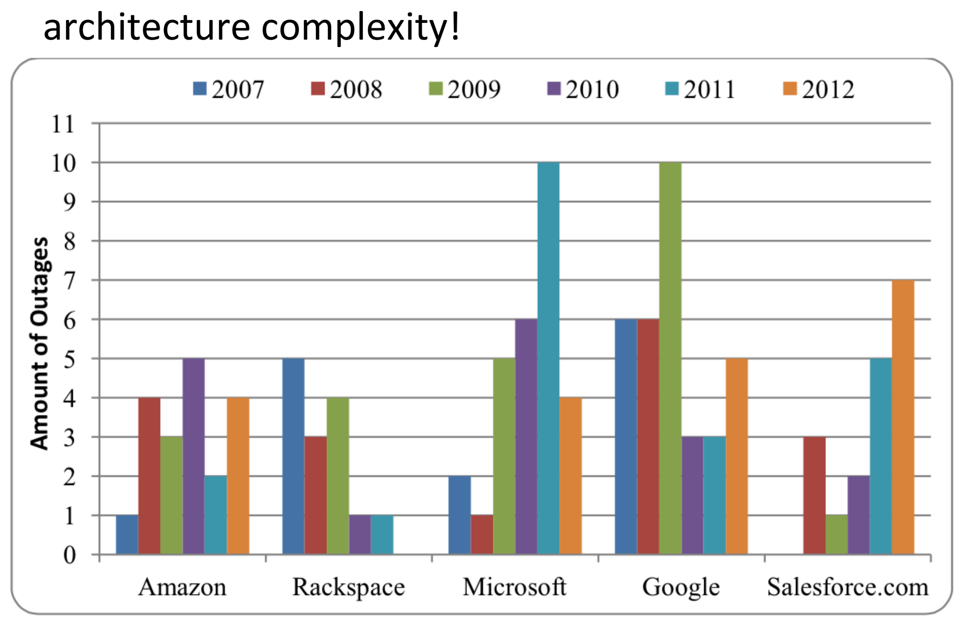
Cloud Computing Ecosystems (CCE)



- **Highly distributed**
- **Heterogeneous hardware** and **software** components
- Expected to provide **highly-available** services requested by **millions of user** in parallel

Motivation

- **Failures** in such ecosystems are inevitable, because too many factors are outside of our control, where almost all is **driven by software**
- **Two degree of complexity**: failures complexity and architecture complexity!

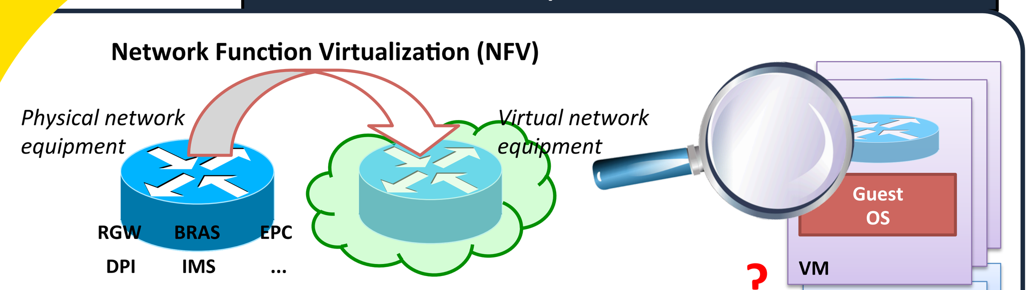


Needs

- What is the **nature** of such **faults** that lead to cloud failures?
- What is the **impact** of **faults** on the CCEs?
- What is the **weakest component/layer** within CCE against faults?

We need to develop **tools** and **methodologies** to evaluate **dependability** issues in CCEs

1. NFV as a complex cloud infrastructure



- **Reduced** costs (CAPEX and OPEX)
- **Improved** manageability
- **Complex** stack of hardware and software **COTS**
- Increasing exposure to **hardware** and **software faults**
- Lack of tools and methodologies for **testing fault-tolerance**

NFV infrastructures have performance and reliability levels compared to legacy systems?

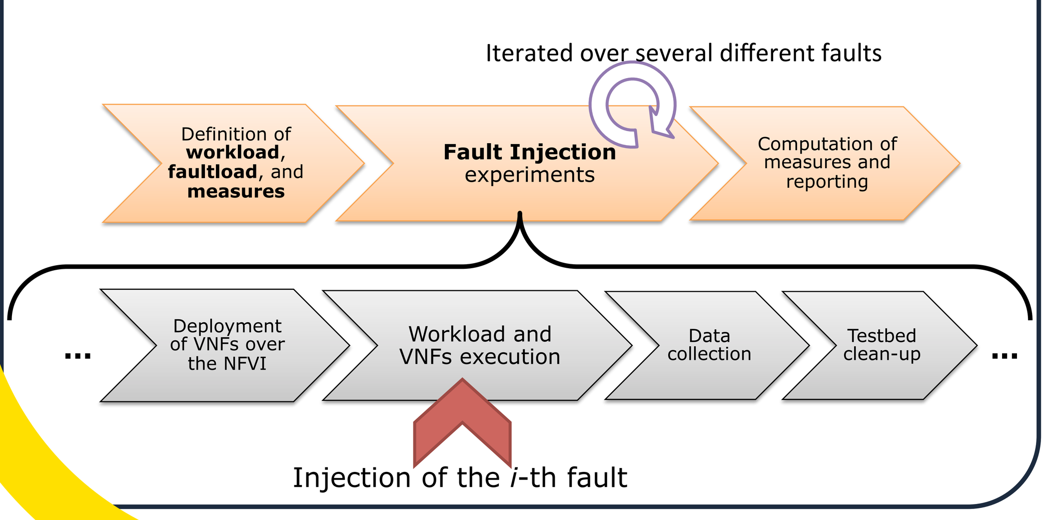
2. Dependability benchmark process

A dependability benchmark evaluates a system **in the presence of (deliberately injected) faults**

Are NFV services still available and high-performing even when a fault is injected?

The dependability benchmark includes:

- **Measures** (*KPIs*) for characterizing performance and availability
- **Procedures**, **Tools**, **Conditions** under which measures are obtained



3. Measures

The dependability benchmark measures the **quality of service** as perceived by NFV users

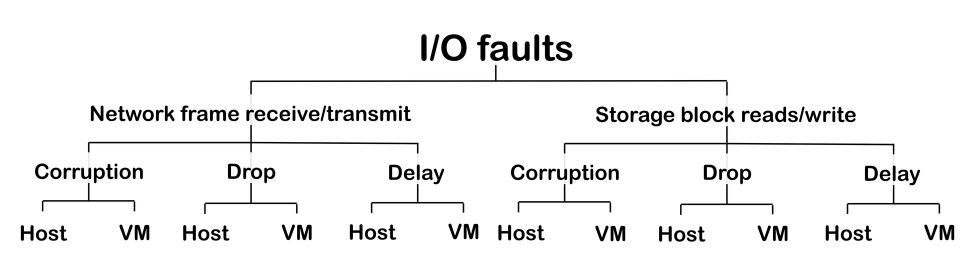
1. ***VNF latency**: the time required to process a unit of traffic (such as a packet or a service request)*
2. ***VNF throughput**: the rate of processed traffic (packets or service requests) per second*
3. ***VNF experimental availability**: the percentage of traffic units that are successfully processed*
4. ***Risk Score**:*

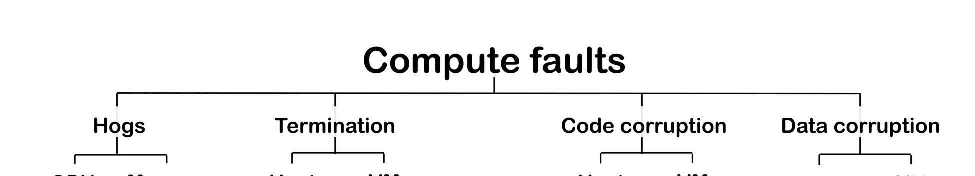
$$RS = \sum_{\substack{\text{Weighted}\\ \text{average over}\\ \text{all faults}}} \left(\%_{\text{Performance failures}} + \%_{\text{Availability failures}} \right)$$

We compare **fault-injected** experiments with the **QoS objectives** and the **fault-free** experiment (*benchmark baseline*)

4. Fault model

- Faults in virtualized environments include disruptions in **network** and **storage I/O traffic**, in **CPUs** and **memory**
- A **fault injector** has been implemented as a set of kernel modules for VMware ESXi and Linux



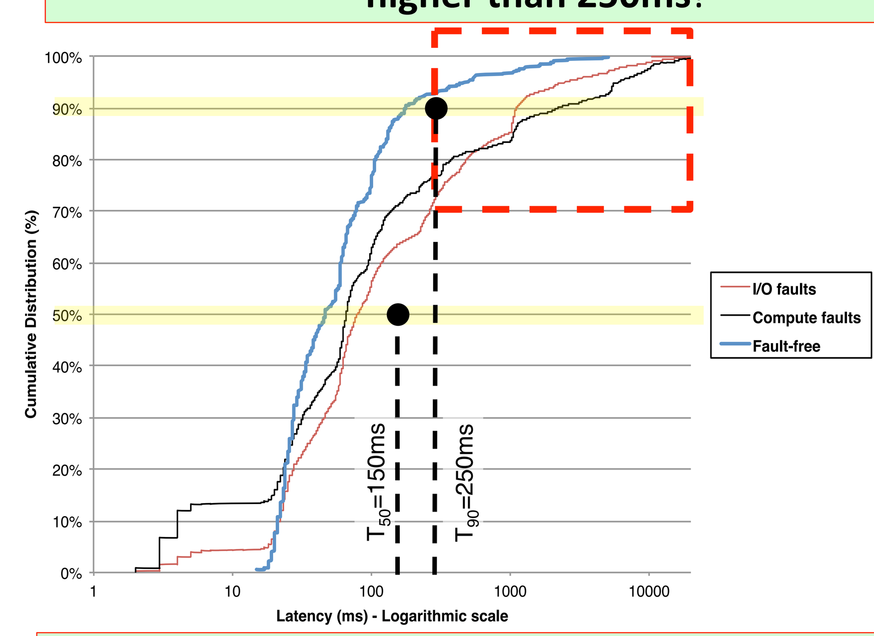


5. Results

Faults have a strong impact on availability! ***Compute faults*** and ***Sprout-VM faults*** have the strongest impact

Fault Type \ Fault Target	Sprout	Homestead	ESXi host	Average
Compute faults	8.01%	39.67%	59.19%	35.62%
I/O faults	48.29%	82.40%	70.67%	67.12%
Average	28.15%	61.03%	64.93%	51.37%

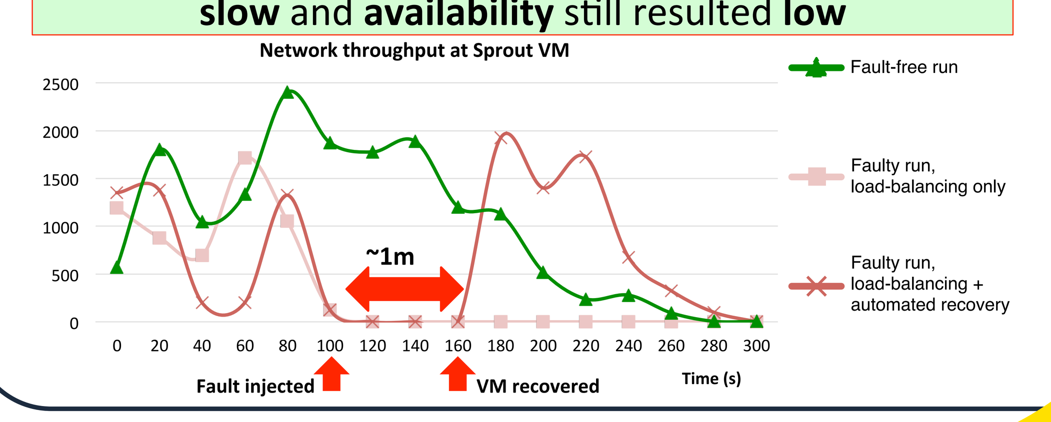
Over than 10% of requests exhibit a latency **much higher than 250ms**!



The **overall risk score (55%)** is quite high and reflects the strong impact of faults on the infrastructure

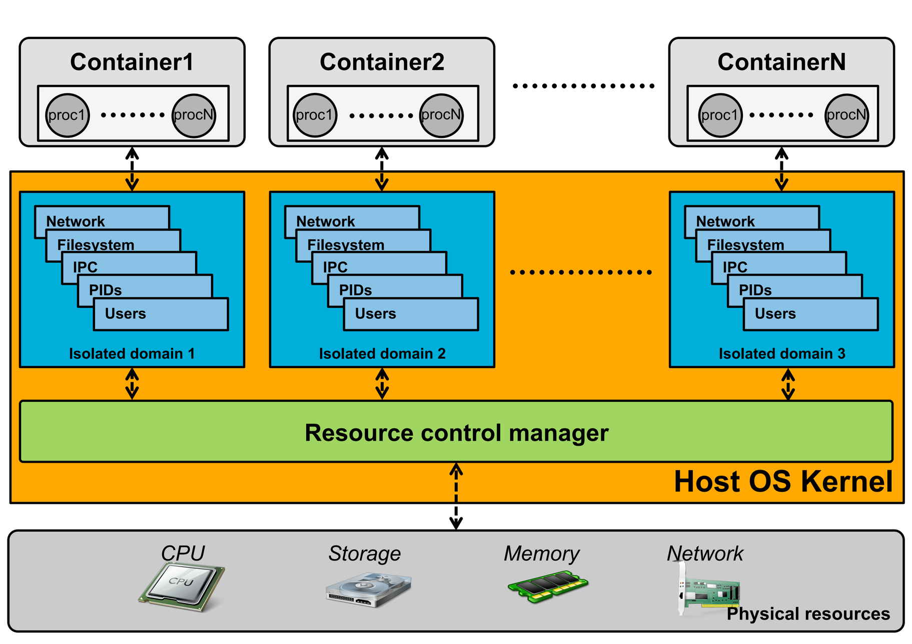
Fault Type \ Fault Target	Sprout	Homestead	ESXi host	All targets
Compute faults	100%	100%	47%	67%
I/O faults	68%	58%	37%	48%
All faults	79%	69%	38%	55%

In our experiments, automated VM recovery was too **slow** and **availability** still resulted **low**



I'm a member of **Mobilab** research group at DIETI – UNINA. I collaborated within a **global leader company of TLC solutions** in a research project that aimed to evaluate reliability in Network Function Virtualization (NFV) infrastructures.

Container-based virtualization technology



Container-based characteristcs

- Container-based Virtualization, also known *OS-level virtualization*, leverages host OS kernel capabilities to execute appliances in isolated domains
- Containers are expected to be **fast**, thus providing **high performance**, since there is no extra overhead due to emulation of devices
- Virtualization is **more manageable**, since creating and moving containers is easier and faster

Container-based NFVI

- NFV infrastructures require
  - extremely low packet processing overheads
  - controlled latency
  - automatic recovery from faults
  - extremely high availability (99.99% or higher).
- Container-based virtualization can be a more suitable solution for NFV
- I'm collaborating within **DEEDS group - TU Darmstadt** - to evaluate and quantify dependability in container-based infrastructures

TECHNISCHE UNIVERSITÄT DARMSTADT