Performance Evaluation of Live Migration Mechanism in a Private Cloud

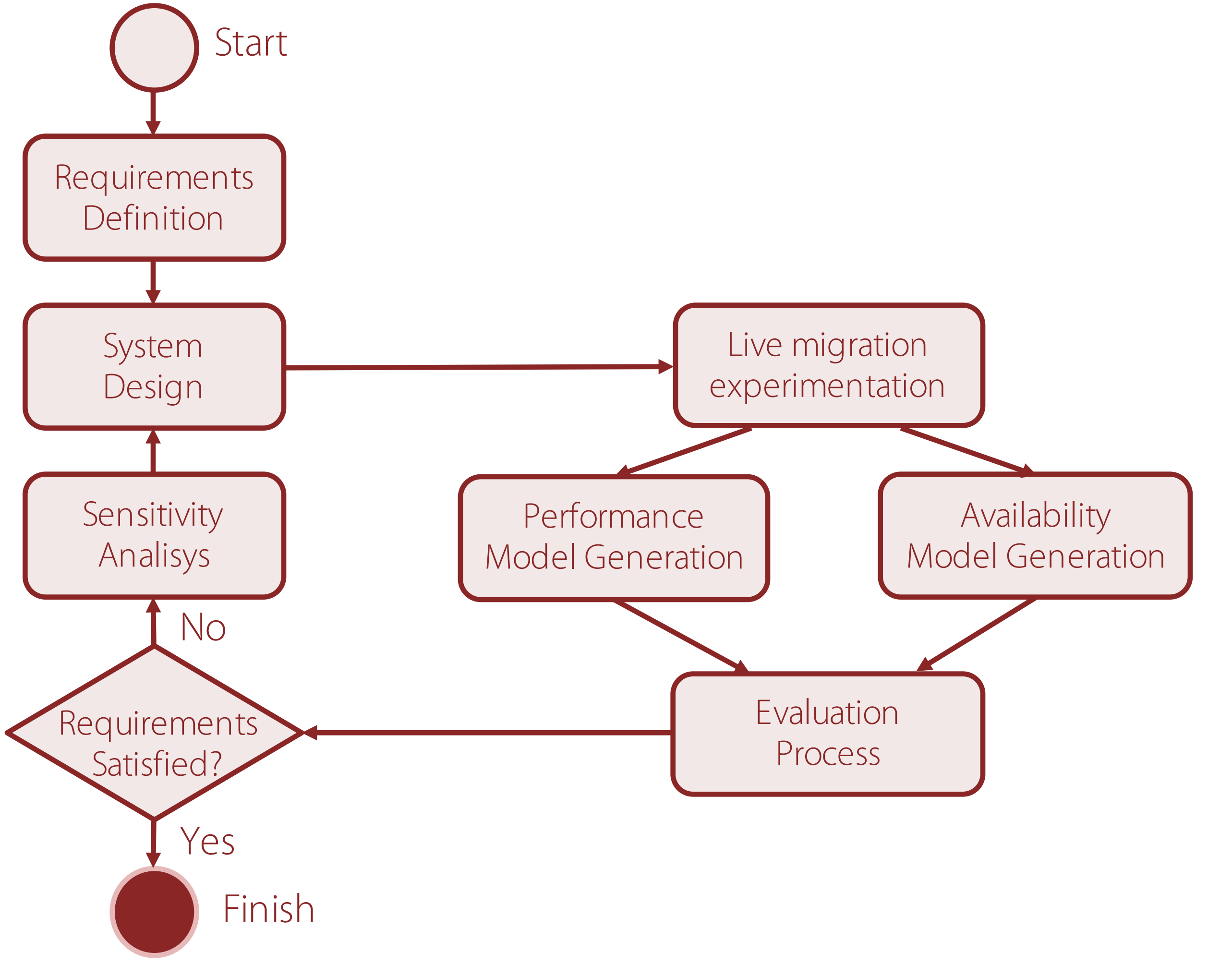
Ph.D. student
André Phillipe Oliveira
apbo@cin.ufpe.br

Advisor
Prof. Paulo R. M. Maciel
prmm@cin.ufpe.br

Outline

- Approach Overview
- Requirements Definition
- System Design
- Live Migration (Case of study)
- Next Steps
- High-Level Model Generation
- Future Works

Approach Overview

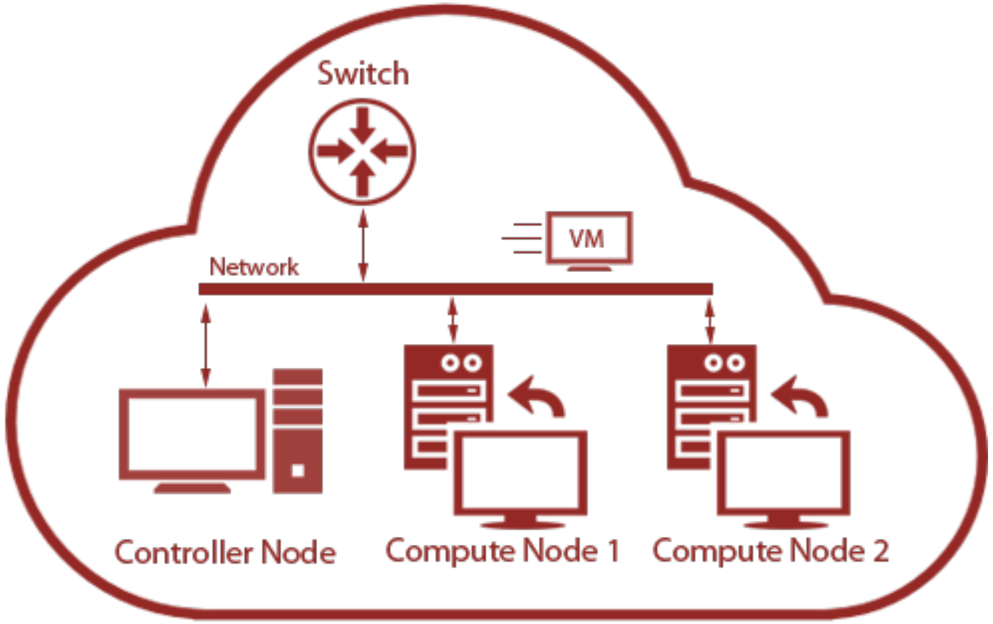


Requirements Definition

- Problem to solve
- Possible contribution
- System boundaries
- Metrics of interest
- Cloud software
- Modeling software
- Sensitivity analysis technique

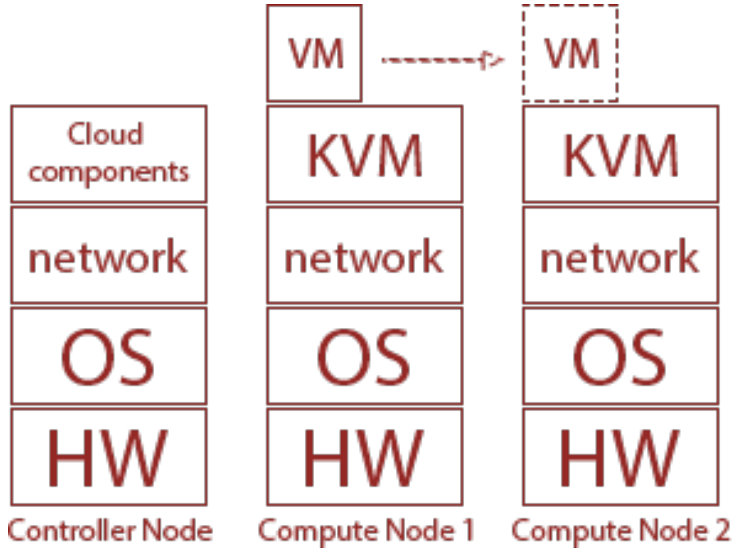
System Design

- Data center
  - Servers configurations
  - Cloud configurations
  - Network devices configurations
  - Select and configure hypervisor
  - Vritual machines configuration



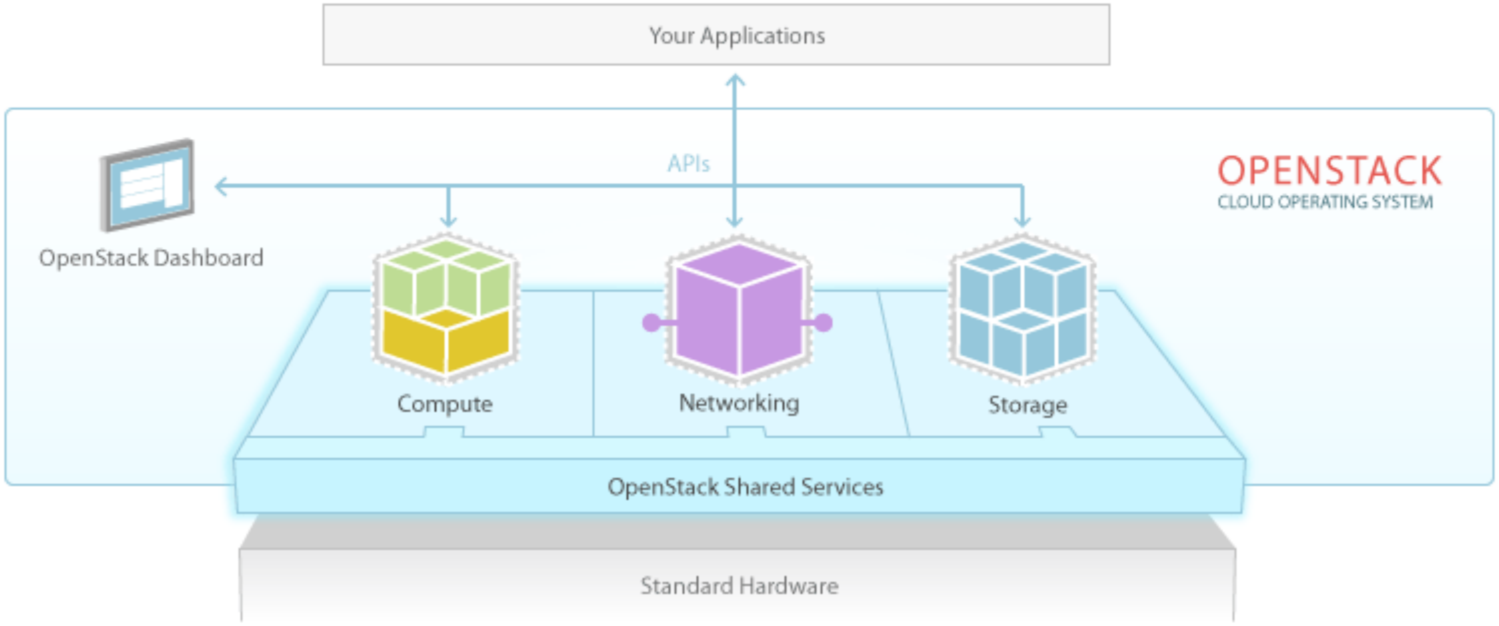
System Design

- Testbed specifications
  - Cluster with 3 servers
    - 1 controller
      - Intel Xeon 3.40GHz (Quad-core processor) 8 CPUs
      - 16Gb RAM
      - Intel VT technology
      - 7200 RPM HD
    - 2 computes nodes
      - Intel Xeon 3.40GHz (Quad-core processor) 8 CPUs
      - 32Gb RAM
      - Intel VT technology
      - 7200 RPM HD
    - 1 switch
      - Gigabit Ethernet



System Design

- OpenStack Newton 3.2.1 (CentOS 7 Linux)



Live Migration (Case of study)

- Live migration is useful for
  - Load balancing
  - Hardware independence
  - Energy saving
  - Geographic migration
  - many other situations
- Sensitivity analysis with DOE
  - Performed to find out which kind of LM has greater effect on customer service availability and performance during the three LM process

Case of Study (experimentation)

- Sensitivity analysis with design of experiments (DOE) mechanism
  - 100 samples of each flavor of virtual machines

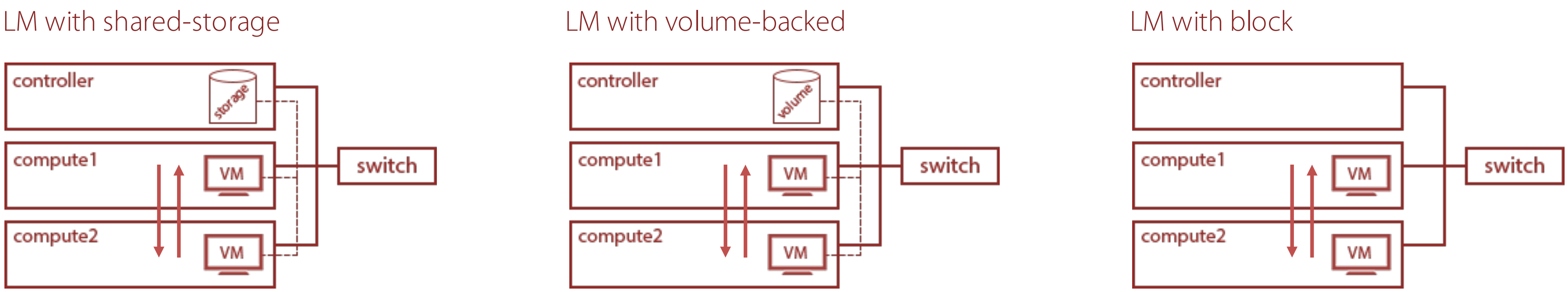
Flavors	VCPUs	Disk (in GB)	RAM (in MB)
m1.tiny	1	5*	512
m1.small	1	20	1024
m1.medium	2	40	2048
m1.large	4	80	4096
m1.xlarge	8	160	16384

Types
Shared storage-based
Block
Volume-backed

* changed to 5Gb because of image minimal requirement

Case of Study (experimentation)

- Sensitivity analysis with design of experiments (DOE) mechanism
  - Live migration types



Case of Study (experimentation)

- Sensitivity analysis with design of experiments (DOE) mechanism
  - Live migration types
    - Shared storage-based
      - The instance has ephemeral (virtual) disks that are located on storage shared between the source and destination hosts.
    - Block
      - The instance has ephemeral (virtual) disks that are **not** shared between the source and destination hosts.
    - Volume-backed
      - Instances use volumes rather than ephemeral (virtual) disks.

Case of Study (experimentation)

- Sensitivity analysis with design of experiments (DOE) mechanism
  - Virtual machine image used in experiment
    - **Name**: Ubuntu Server 16.04 LTS (Xenial Xerus)
    - **File name**: xenial-server-cloudimg-amd64-disk1.img
    - **File size**: 272mb
    - **Disk format**: QCOW2
    - **Arch**: amd64

Case of Study (experimentation)

- Sensitivity analysis with design of experiments (DOE) mechanism
  - Results (sample)
    - **Hypervisor**: KVM
    - **Virtualization API**: libvirt
    - **Instante name**: instance-0000011a
    - **Command to retrieve the information**:
      - virsh domjobinfo instance-0000011a --completed
    - **Result fields collected**:
      - Time elapsed
      - Total downtime

Example:

Job type:	Completed
Time elapsed:	**4919 ms**
Time elapsed w/o network:	4918 ms
Data processed:	505.981 MiB
Data remaining:	0.000 B
Data total:	8.016 GiB
Memory processed:	505.981 MiB
Memory remaining:	0.000 B
Memory total:	8.016 GiB
Memory bandwidth:	107.625 MiB/s
Dirty rate:	0 pages/s
Iteration:	3
Constant pages:	1977424
Normal pages:	124942
Normal data:	488.055 MiB
Total downtime:	**60 ms**
Downtime w/o network:	59 ms
Setup time:	27 ms

Case of Study (experimentation)

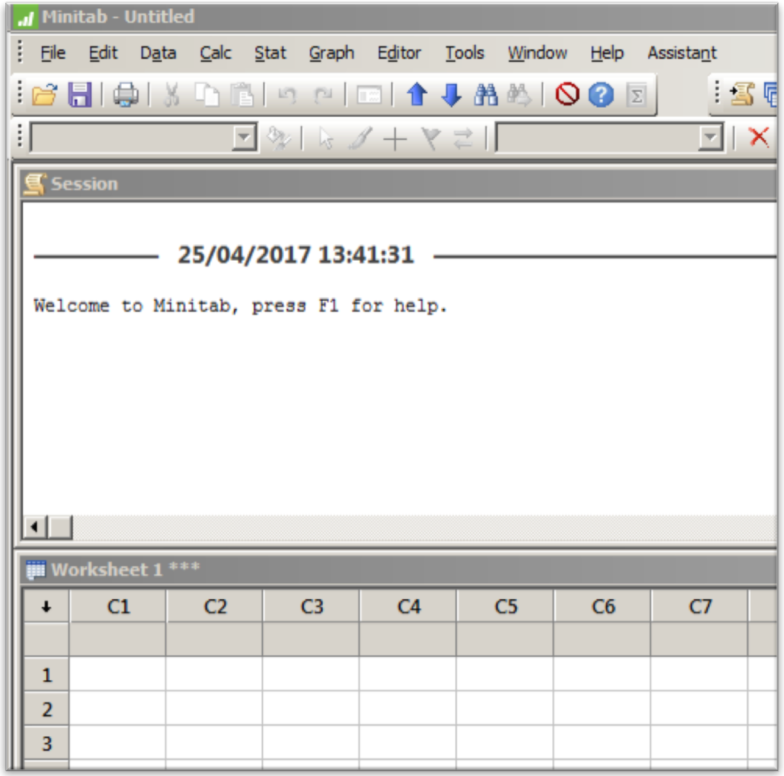
- Sensitivity analysis with design of experiments (DOE) mechanism
  - Results (sample)

flavors	lm_type	mig_time(ms)	downtime(ms)
m1.tiny	shared_storage	3277	52
m1.small	shared_storage	3531	54
m1.medium	shared_storage	4078	60
m1.large	shared_storage	4958	62
m1.xlarge	shared_storage	6738	85

flavor: Flavors are used to define the compute, memory, and storage capacity of nova computing instances.
lm_type: Type of live migration.
mig_time: Migration time in milliseconds.
downtime: Downtime in milliseconds.

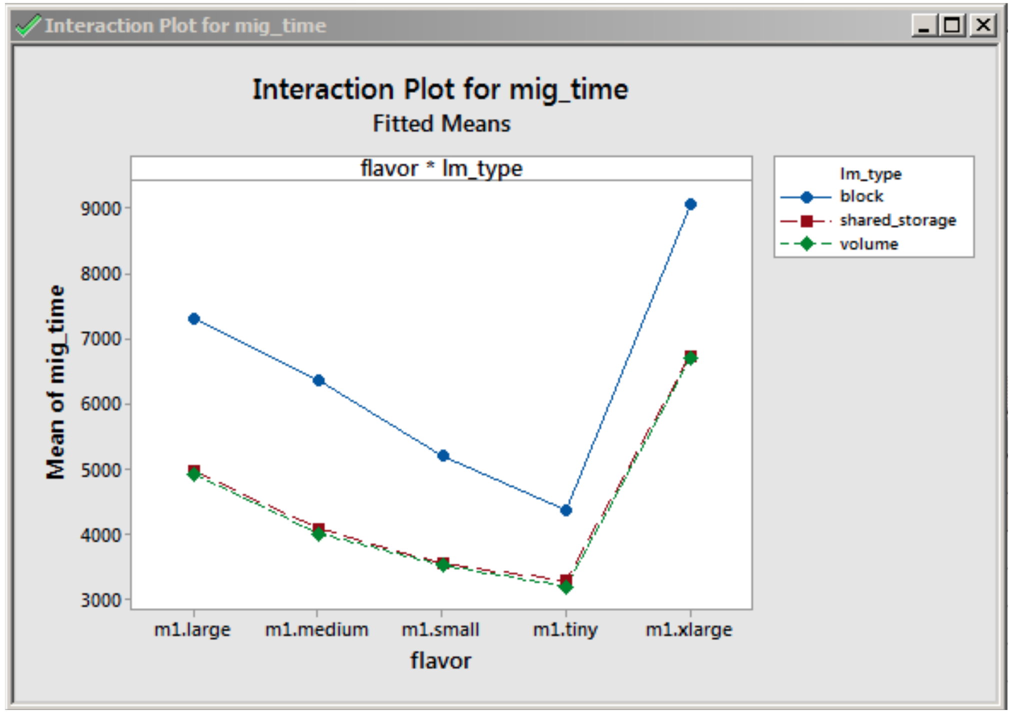
Case of Study (experimentation)

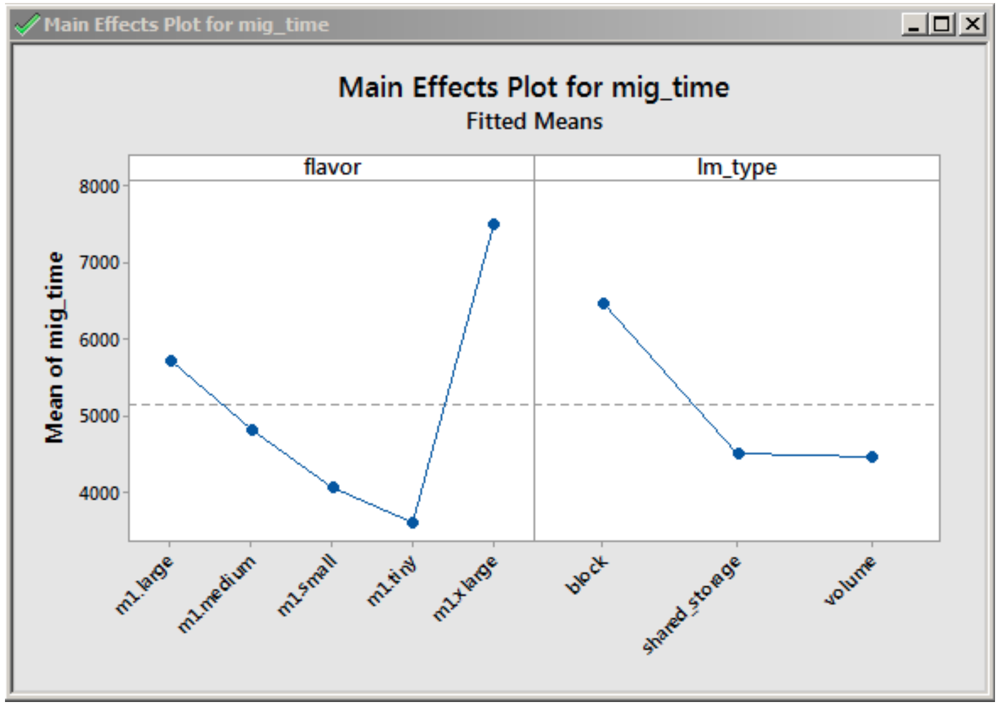
- Sensitivity analysis with design of experiments (DOE) mechanism
  - Minitab 7 tool



Case of Study (experimentation)

- Sensitivity analysis with design of experiments (DOE) mechanism
  - Graphs (samples)





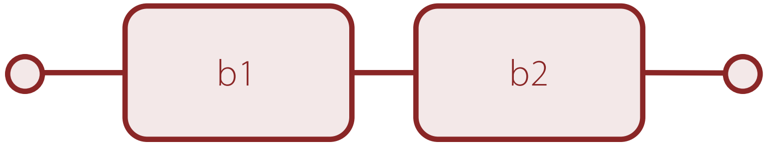
Next Steps

- High-Level Model Generation
- Performance Model Generation
- Availability Model Generation
- Evaluation Process
- Sensitivity Analisys

High-Level Model Generation

- Reliability Block Diagram (RBD) serie

$$P_s(t) = \prod_{i=1}^{n} P_i(t)$$



Metric	Description
MTTF	Mean Time To Fail
MTTR	Mean Time To Repair

In which $P_i(t)$ is the realiability or availability of the blocks.

High-Level Model Generation

- Stochastic Petri Nets (SPN)

$$\mathcal{N} = (P, T, I, O, M_0)$$

$$P = \{p_1, p_2, \dots, p_n\}$$

$$T = \{t_1, t_2, \dots, t_n\}$$

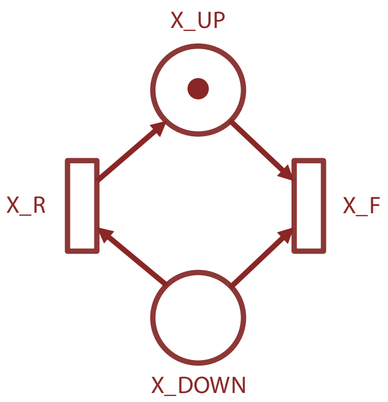
$$I \in (\mathbb{N}^n \rightarrow \mathbb{N})^{n \times m}$$

$$O \in (\mathbb{N}^n \rightarrow \mathbb{N})^{n \times m}$$

$$M_0 \in \mathbb{N}^n$$

$$A_s = P\{\#X\_UP > 0\}$$

Basic component SPN submodel and details



Transition	Delay	Description
X_F	MTTF	Component failure event
X_R	MTTR	Component repair event

Place	Condition
X_UP	Component is working
X_DOWN	Component is not working

High-Level Model Generation

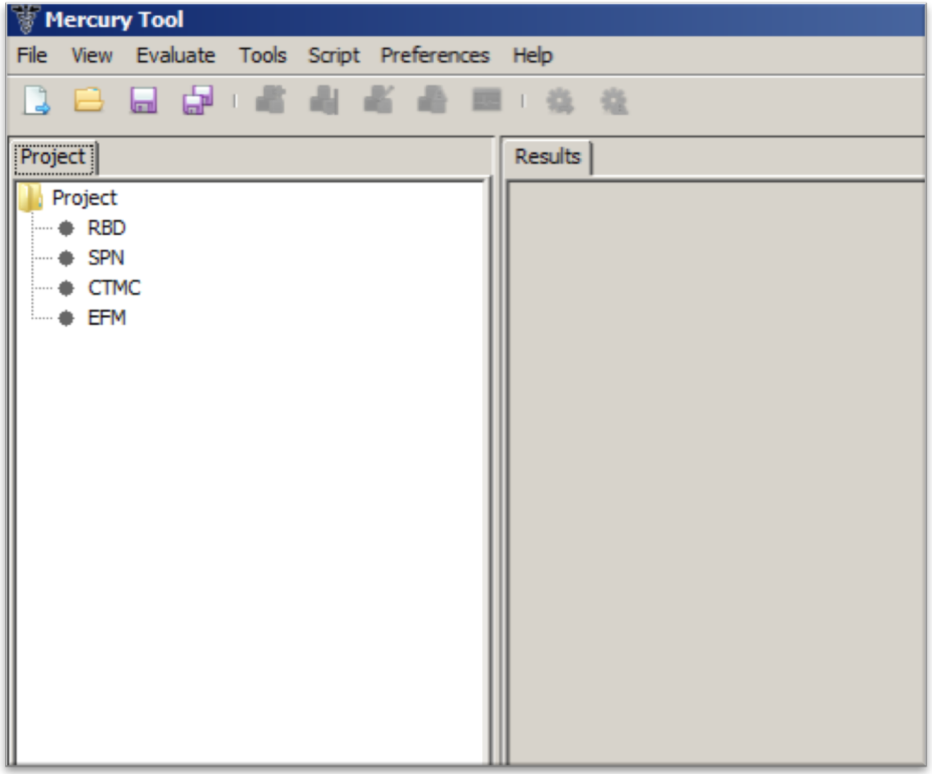
- Live migration process (events) for modeling

1. conductor_migrate_server
2. compute_check_can_live_migrate_destination
3. compute_check_can_live_migrate_source
4. compute_live_migration
5. **compute_pre_live_migration**
6. **compute_post_live_migration_at_destination**

1. compute.instance.update
2. compute.instance.live_migration.pre.start
3. compute.instance.live_migration.pre.end
4. **compute.instance.update** (downtime starts)
5. compute.instance.live_migration._post.start
6. compute.instance.live_migration._post.end
7. compute.instance.live_migration.post.dest.start
8. **compute.instance.update** (downtime ends)

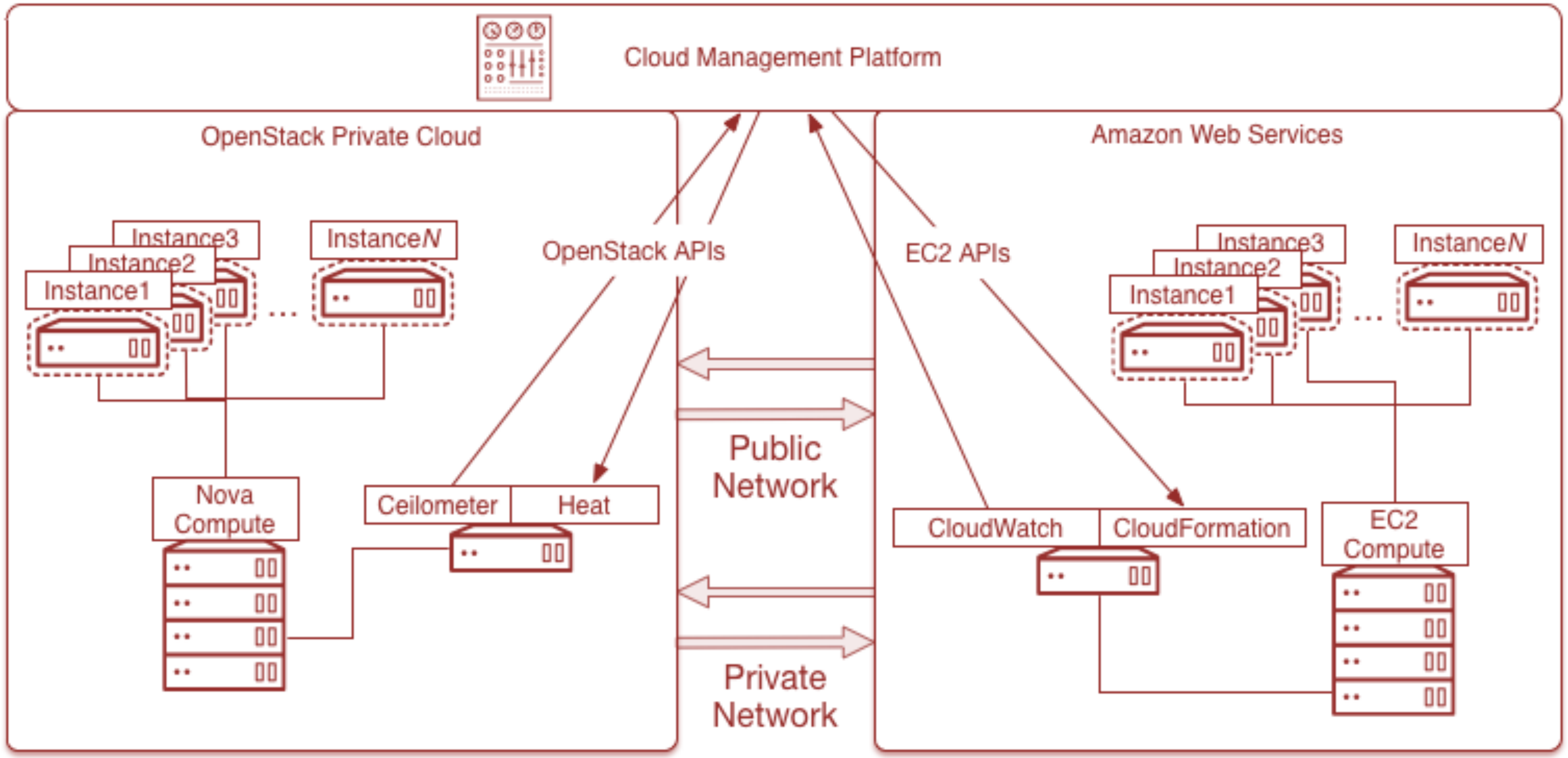
High-Level Model Generation

- Mercury 4.6.3



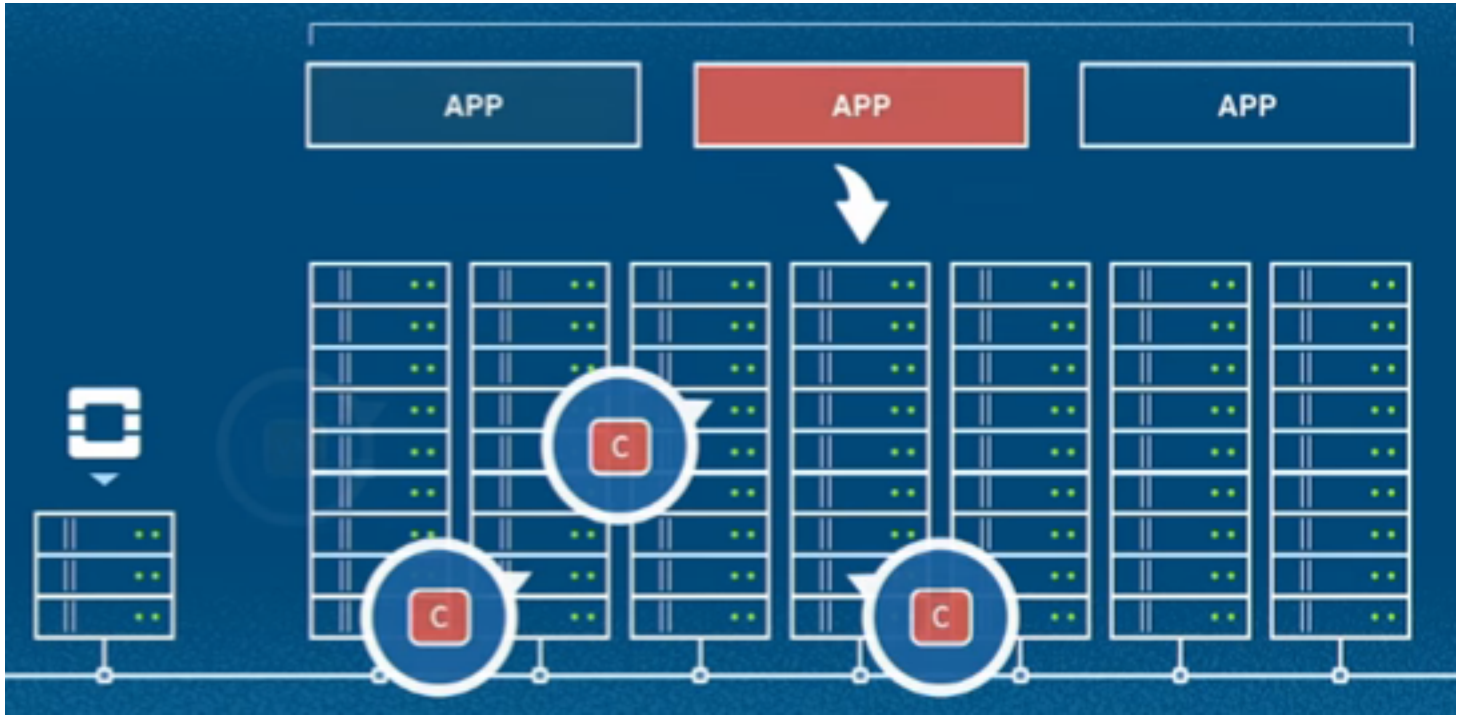
Future Works

- Hybrid architecture



Future Works

- Service distributed in containers



References

- Herzog, U. (2001). Formal methods for performance evaluation. In Lectures on Formal Methods and Performance Analysis
- Maciel, P. R., Trivedi, K. S., Matias, R., & Kim, D. S. (2011). Dependability modeling. Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions, 1, 53-97.
- Avizienis, A., Laprie, J. C., & Randell, B. (2001). Fundamental concepts of dependability. Newcastle upon Tyne, UK: University of Newcastle upon Tyne, Computing Science.
- Cassandras, C. G. (1993). Discrete event systems: modeling and performance analysis. CRC.
- Jiang, J., Sekar, V., & Zhang, H. (2012, December). Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. In Proceedings of the 8th international conference on Emerging networking experiments and technologies (pp. 97-108). ACM.
- JAIN, R. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. [S.l.]: Wiley, 1991. (Wiley Professional Computing). ISBN 9780471503361.

Thank you

Workshop MoDCS Apr'17