

HASLab
HIGH-ASSURANCE
SOFTWARE LABORATORY

Automatic Elasticity in OpenStack

Leander Beernaert, Miguel Matos, **Ricardo Vilaça**,
Rui Oliveira

SDMCMM
4 December 2012



INESCTEC
TECNOLOGIA E CIÊNCIA
LABORATÓRIO ASSOCIADO



Universidade do Minho

Context

- ◎ Cloud Computing has become an increasingly active topic.
- ◎ The illusion of a virtually infinite computing infrastructure
- ◎ Elasticity is a requirement in Cloud computing infrastructures.
 - ◎ Allows pay-per-use billing models.
- ◎ Various open source software solutions for IaaS.
 - ◎ Eucalyptus, OpenNebula and OpenStack.

Elasticity advantages

- Customer

- Ability to adjust resource consumption according to unpredicted demand.
- Ability to free resources when they are not needed allowing to reduce costs.

- Provider

- Allows VM consolidation and reduced server usage.
- Keep SLAs/cluster occupancy ratio with the minimum resources possible.
- Increase profit.

Motivation

- ◎ Some of the systems offer monitoring features.
- ◎ None of them have any form of automated elastic behavior.
- ◎ Elasticity feature is quite immature.
- ◎ Monitoring and timely adapting the active resources of a Cloud computing infrastructure is key to provide elasticity.

Existing work

- © Some projects provide a way to integrate the monitoring component in a custom management software, Zenoss.
- © CloudScale performs a series of previsions and adds resources to or removes resources from the instance.
- © Amazon's Auto-Scale.

Openstack elasticity

- The elasticity feature is still in an embryonary phase.
- Supports elastic operations (e.g.: allocation of more resources to an instance).
 - Need to be performed manually.
- Lack of a monitoring and adaptation mechanisms.

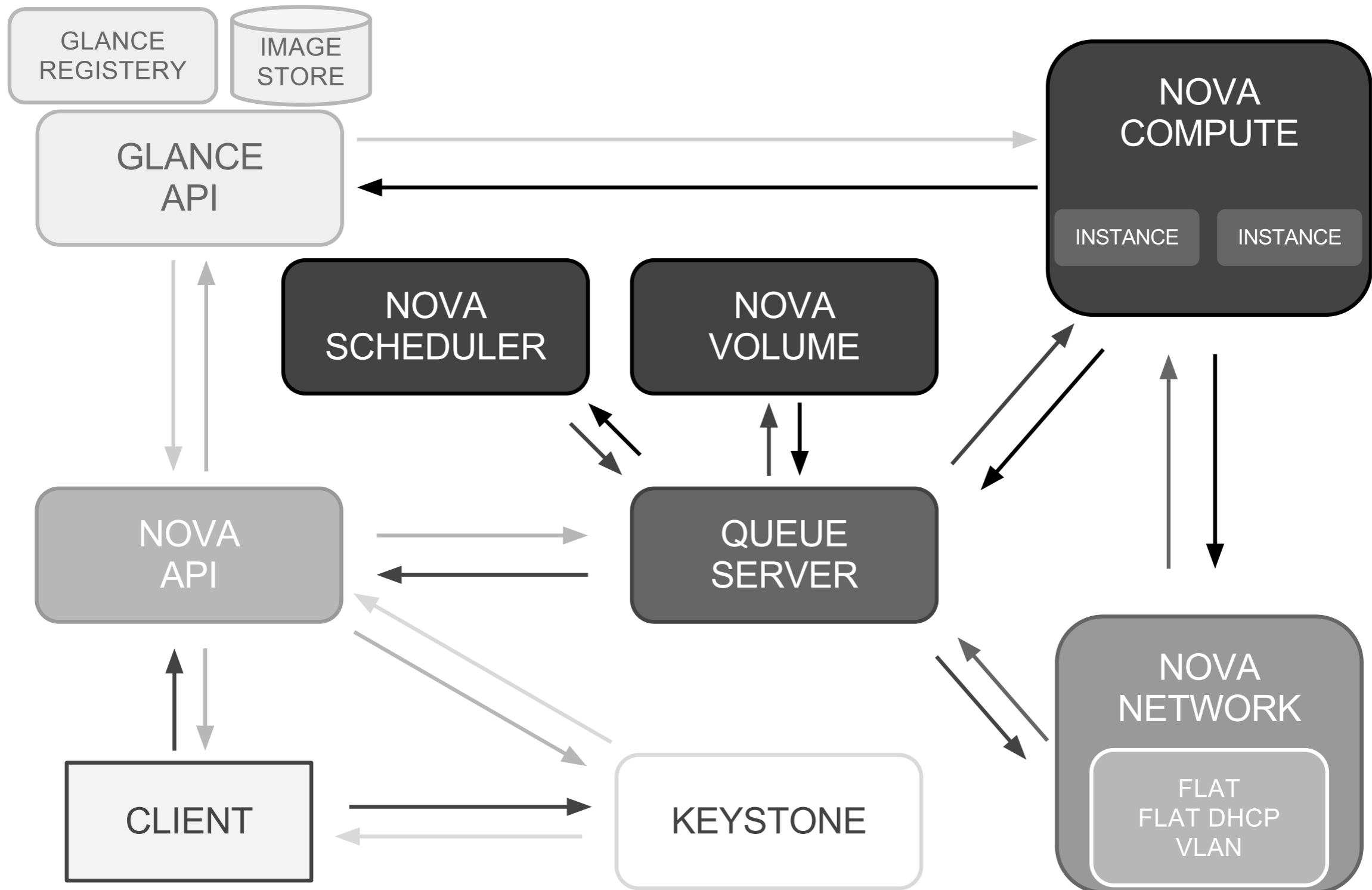
Elastack

- An automated monitoring and adaptive system.
- Monitors the state of the instances and collects data.
- Generic enough to be applied to existing IaaS frameworks.
- Intended to enable the elasticity.
- Bridges the gap in terms of adaptability and monitorization.

OpenStack

- Contributions from NASA, RackSpace, Canonical, Dell, Citrix and VMWare.
- Besides providing its own API, it has an API that is compatible with Amazon's EC2 and S3.
- Composed of various components:
 - **Compute(Nova)**
 - Object Storage (Swift)
 - **Identity (Keystone)**
 - **Image Service (Glance).**

OpenStack architecture



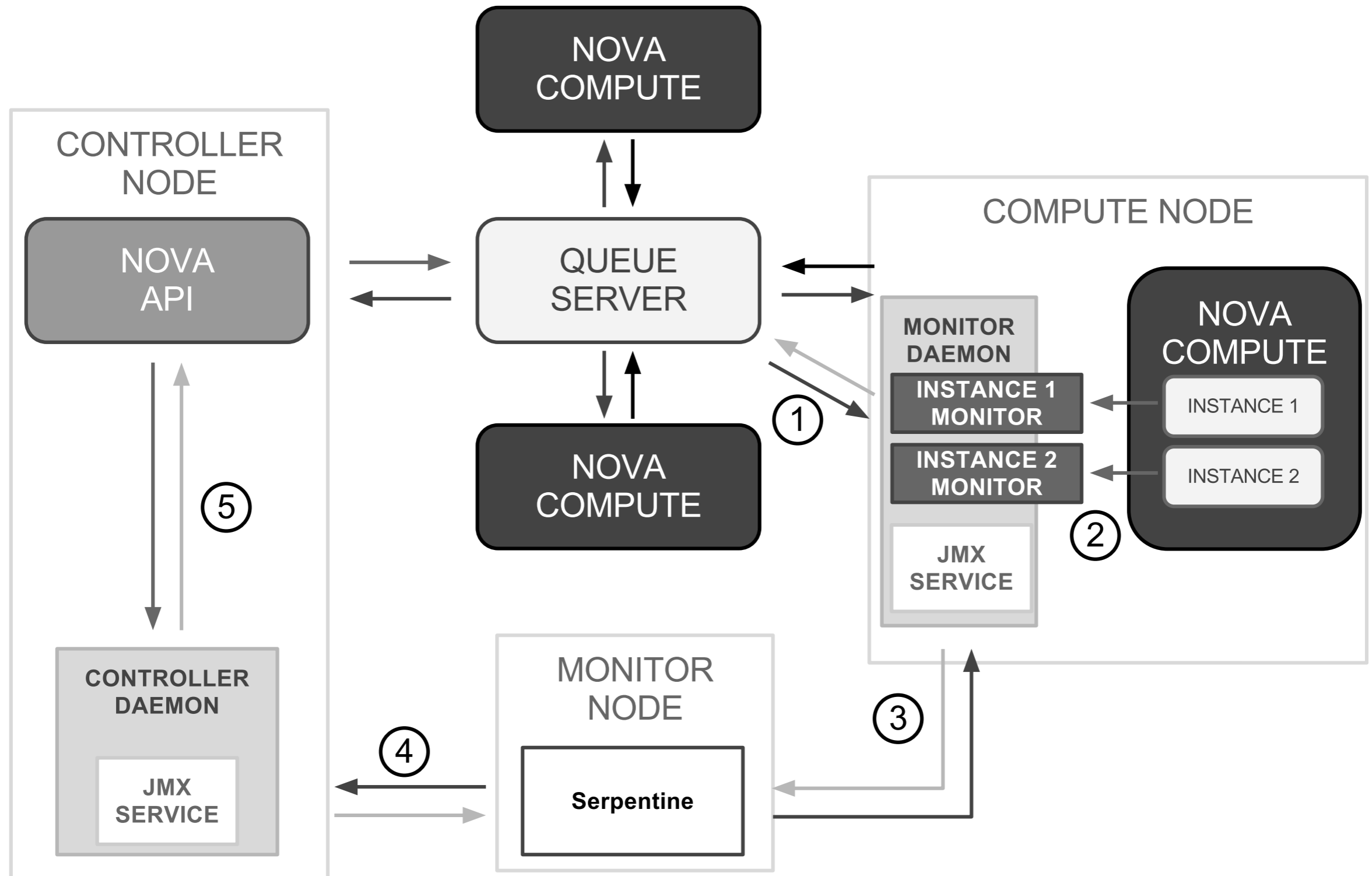
OpenStack details

- Nova
 - Is in charge of all the procedures required to sustain an instance's lifecycle: computational resource management, networking, authorization and scalability.
 - Does not have any virtualization features. It resorts to the libvirt APIs.
- Keystone
 - Central identity and authentication service.
 - Also acts as a discovery point for all the services.
- Glance
 - Is responsible for the storage and availability of the operating system images.

Serpentine

- Is an adaptive middleware which allows a service/system to adapt to changes.
- Was designed to be scalable:
 - its components do not depend on a persistent state.
 - can be organized in a hierarchy, thus permitting micro and macro-management.
- Communication occurs through JMX and the control logic is defined through scripts (JavaScript, Python, Ruby, among others).
- The scripts define the management policies to be applied to the system.

Elastack architecture



Elastack implementation

- In order to monitor each instance, OpenStack provides a command entitled "diagnostics".
 - Is only available for the XEN hypervisor through the XEN API.
- We extend OpenStack to support the method with the libvirt hypervisor.
- To monitor the instances we need to know when they are created and terminated.
 - OpenStack offers a notification system.
- The collected data is made available through a MBean exported by the monitor daemon.
 - This MBean provides methods to access information regarding the host machine, the number of instances running and CPU, memory, network traffic and disk usage of each instance.

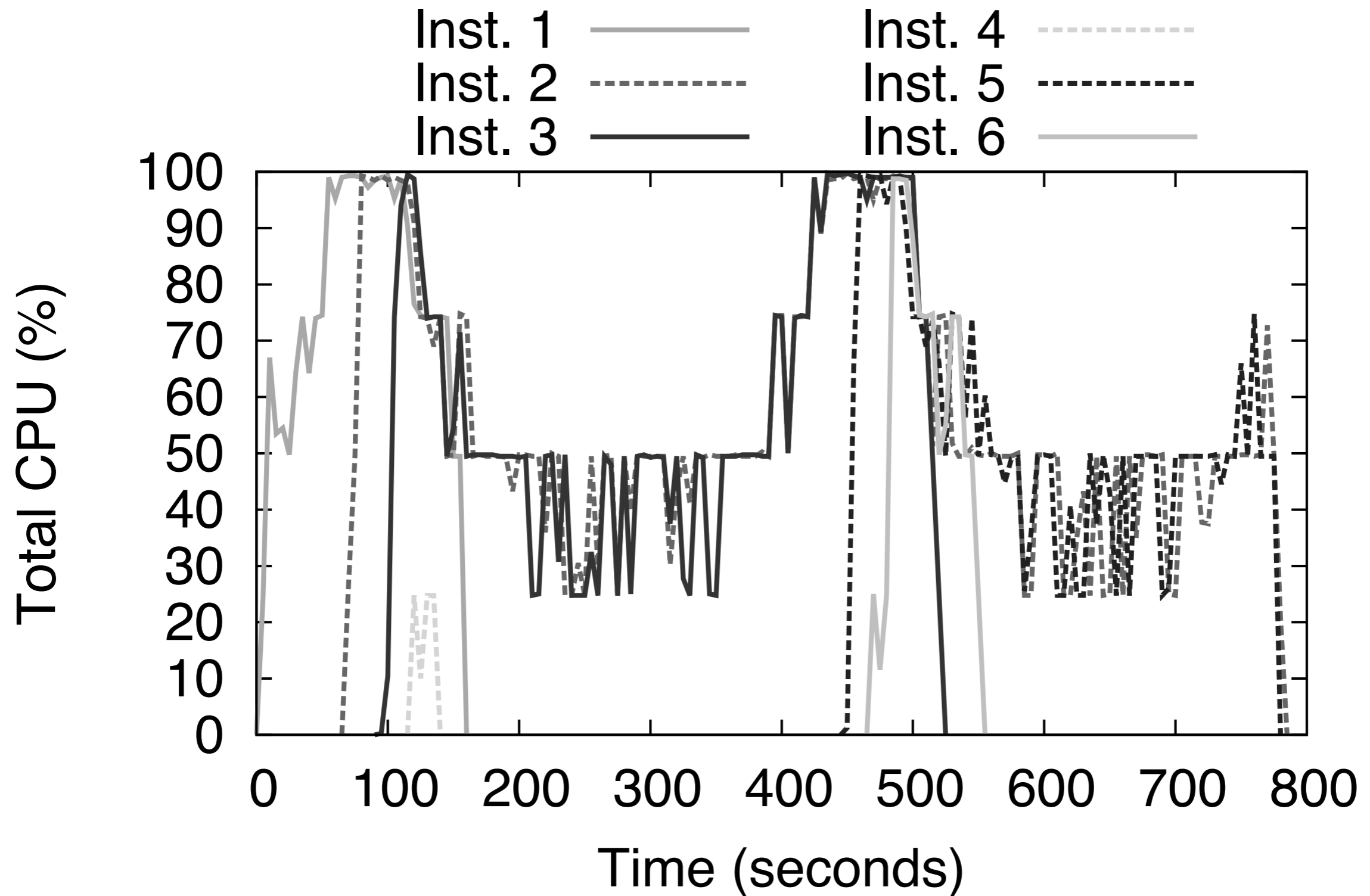
Evaluation setting

- Installation of OpenStack Essex with four nova-compute nodes and one controller node.
- Each of the machines is equipped with a i3-2100 processor clocked at 3.10GHz, 4GB RAM and a 250 GB SATA II hard drive.
- New instances are created when $x \geq (90 * n)$, where x is the combined CPU usage of all instances and n the number of instances.
- Instances are removed when the total CPU usage can be distributed by the remaining instances.

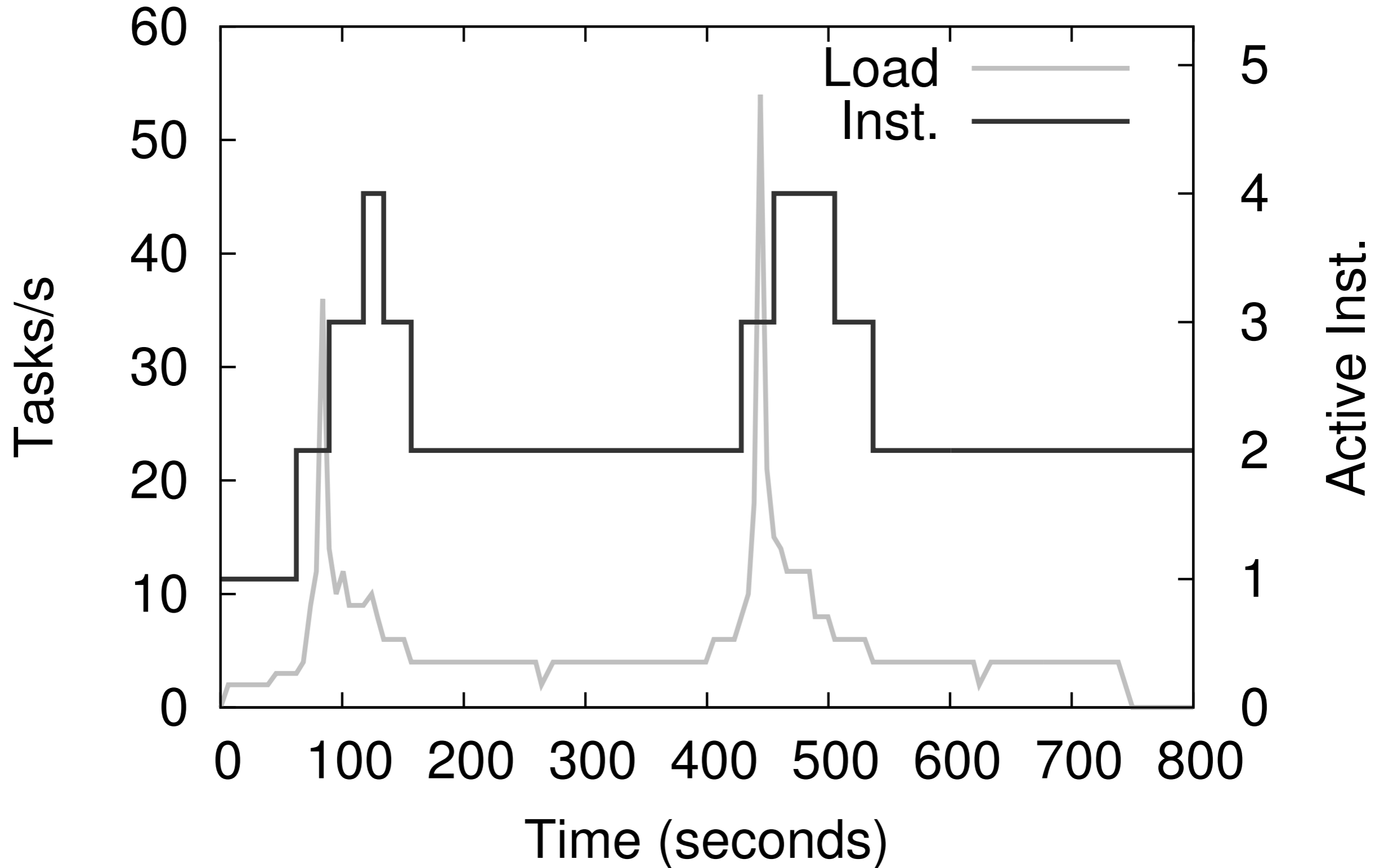
Workload

- ⦿ A small load balancer written in Java which distributes a collection of tasks.
- ⦿ Each instance is preconfigured to run the client process which will receive the tasks from the load balancer.
- ⦿ Each task executes the stress command for one CPU core during 22 seconds.
- ⦿ Simulates a variable workload.
 - ⦿ The load balancer sleeps for a certain amount of seconds before distributing a new set of tasks to the instances.
 - ⦿ A greater sleep time will result in a reduced work load.

Evolution of the instance's CPU load



Evolution of instances and incoming throughput



Conclusion

- Elastack, an adaptation and monitoring component for Cloud infrastructures (IaaS).
- Evaluation of Elastack demonstrated a good adaptability to the submitted load.
 - Increase or decrease the number of virtual machines.
- Elastack can be used for a wide variety of tasks beyond providing elasticity and monitoring capabilities.
- Implementation on OpenStack Folsom release.