



# Improving Hadoop performance with reliable opportunistic instances in OpenStack

Telles Nóbrega, Henrique Truta, Andrey Brito  
Univesidade Federal de Campina Grande

# Cloud Computing

- Cheap and flexible way to get resources on demand
- Provides resources in a *pay-as-you-go* fashion
- Have three basic types of services:
  - Infrastructure-as-a-Service
  - Software-as-a-Service
  - Platform-as-a-Service

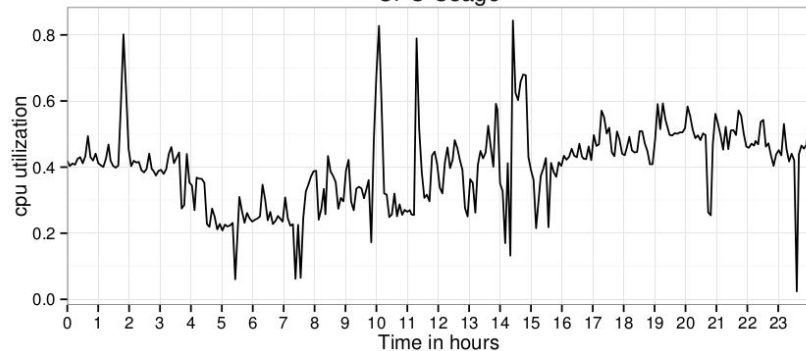
# Underutilization of resources

- Overall cluster utilization is around 60% for CPU and 50% for RAM
- The same behavior can be seen when we look at single machines of the cluster
- Nevertheless, the user's quota (especially on private clouds) cannot be overdimensioned

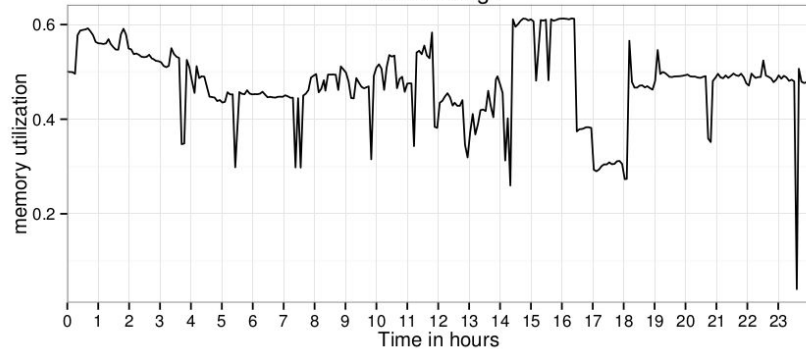
# Underutilization of resources (Google Open Data)

Usages for Machine 121306

CPU Usage

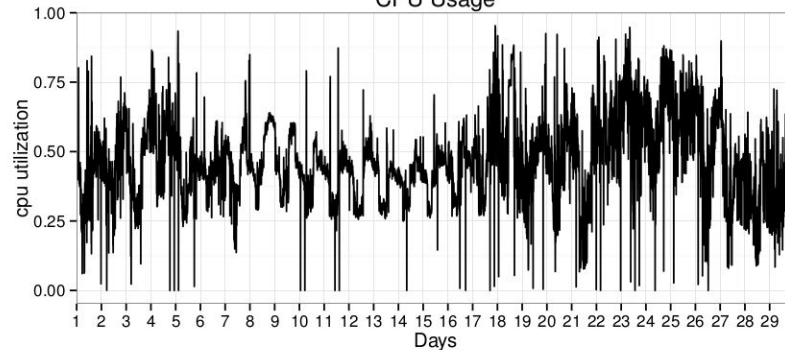


MEM Usage

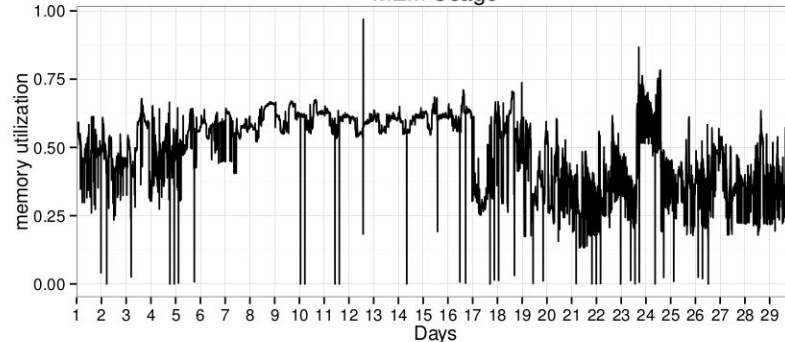


Usages for Machine 121306

CPU Usage



MEM Usage



# Data Processing with Hadoop

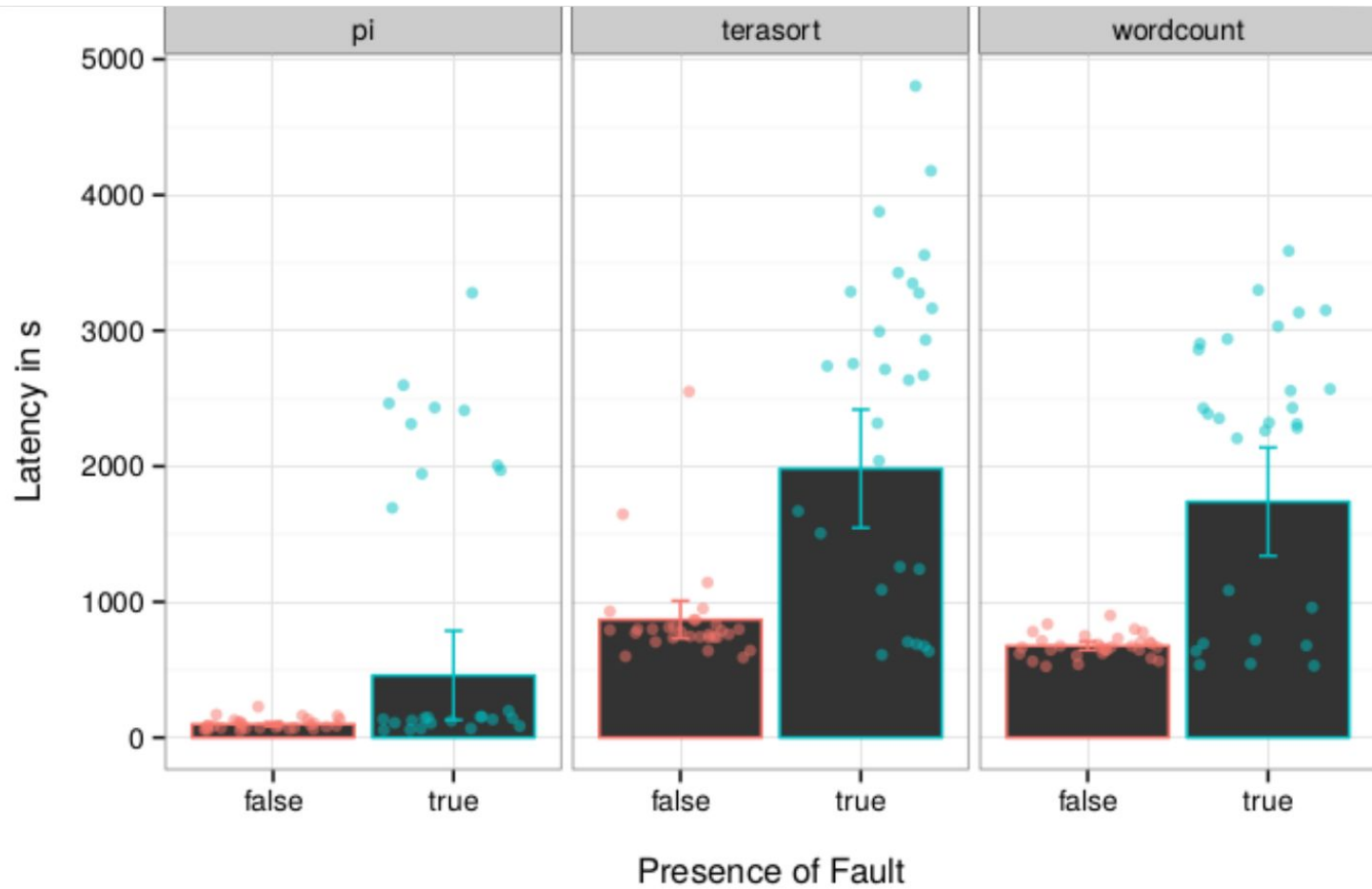
- Hadoop is one of the most used data processing tools on the market
- Implements the MapReduce paradigm
- Fault tolerant
- Often moved to the cloud
  - Easy to scale
  - Lower costs
  - Gain in flexibility often compensate loss in performance

# Hadoop as a Service

- The PaaS model makes the Hadoop processing on the cloud easier
- The user can request a configured Hadoop cluster and just submit his/her data and applications
- Amazon Elastic MapReduce is an example, OpenStack Sahara is an equivalent solution

# Opportunistic instances

- Cheaper type of instances (somehow similar to AWS Spot instances)
- Spawned in underused compute resources
- The goal is to use them to speed up Hadoop processing
  - However, those instances can be preempted if the resources are needed for a higher priority (regular) instances
  - Losing the instance in the middle of a job execution is extremely harmful for the application performance

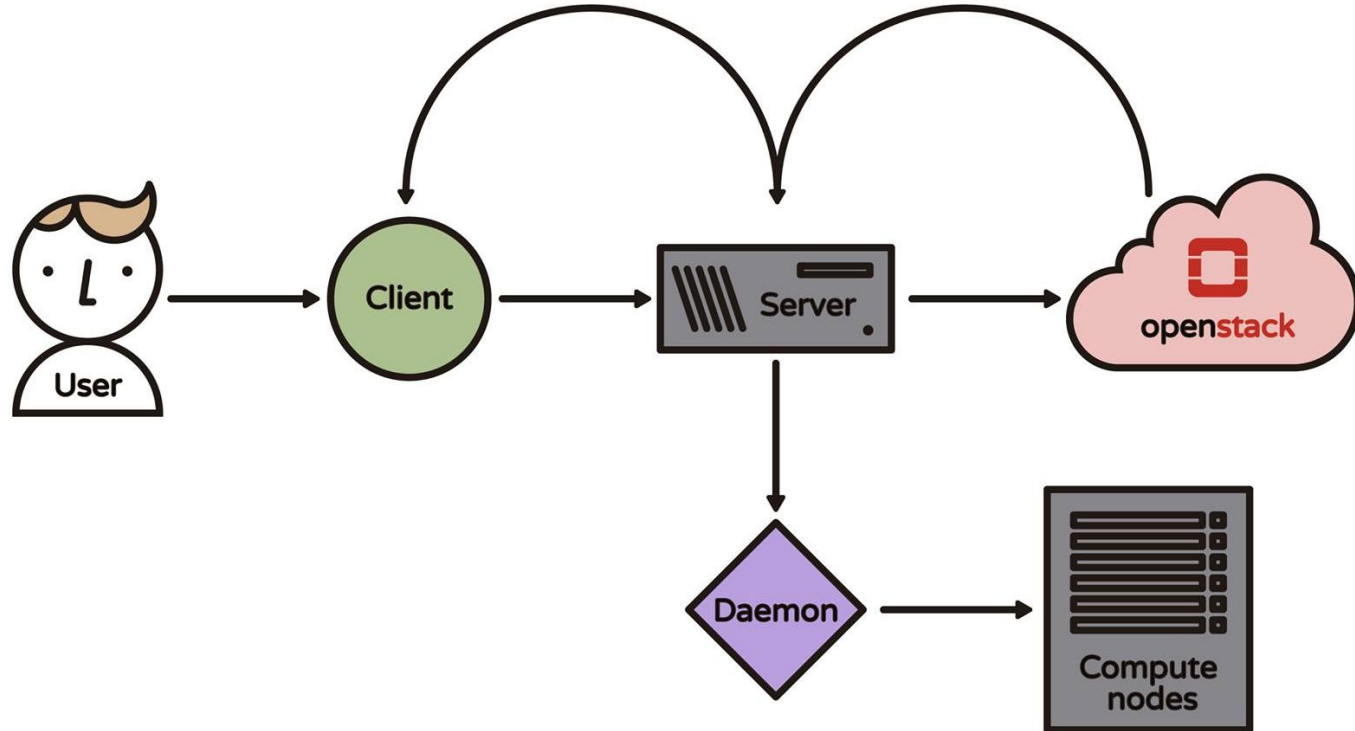




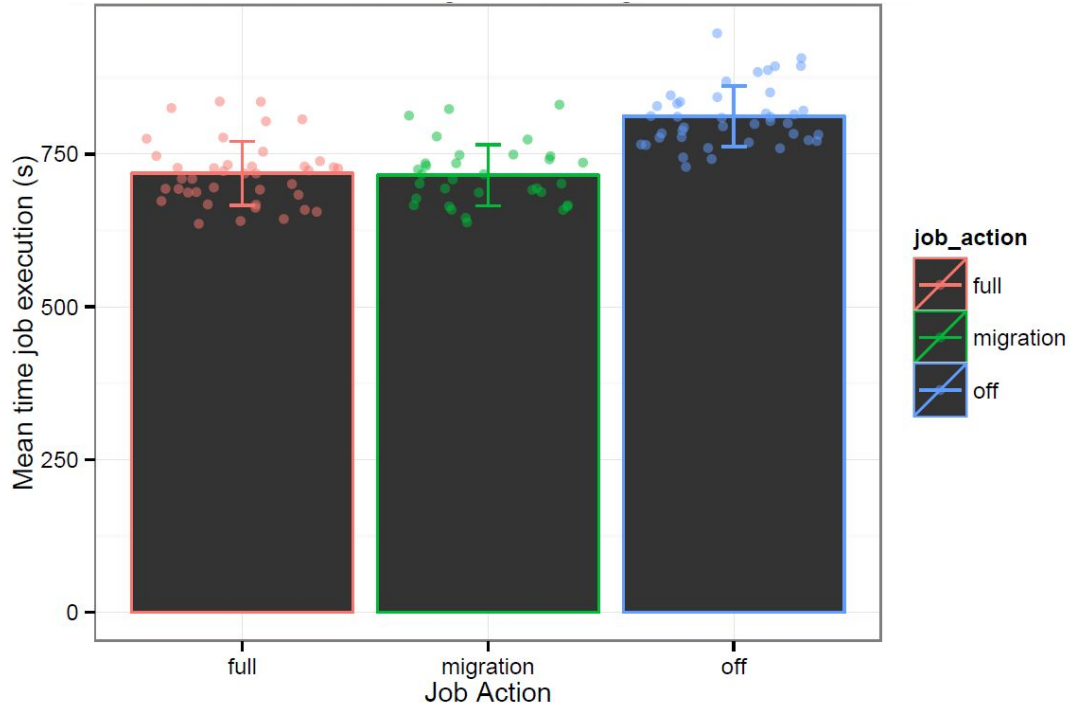
# The approach

- A framework that uses opportunistic instances in Hadoop processing with a higher guarantee that this instance will not be lost
- Uses a workload predictor to forecasts the cloud utilization in a short time window, estimating how many instances will be available
- If the predictor underestimates the workload, live migration is used so the VM is not lost
  - Moves the instance to a different host without turning them off
  - Downtime ranges from a milliseconds to a couple of seconds

# The approach



# Results with an extra instance (from 3 to 4)



# Results

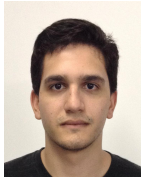
- VMs loss reduced with prediction
- Live migration has almost no interference on other processes
- Better resource usage in the cloud without the risks over overprovisioning or increasing the quota

# Thank you!



**Andrey Brito**

Professor - Universidade Federal de Campina Grande



**Telles Nóbrega**

Master Degree Student - Universidade Federal de Campina Grande  
OpenStack ATC



**Henrique Truta**

Master Degree Student - Universidade Federal de Campina Grande  
OpenStack ATC