# SELF-MANAGEMENT OF CONTAINERS DEPLOYMENT IN GEO-DISTRIBUTED ENVIRONMENTS

**Fabiana Rossi**
University of Rome Tor Vergata

*f.rossi@ing.uniroma2.it*

**InfQ 2019**

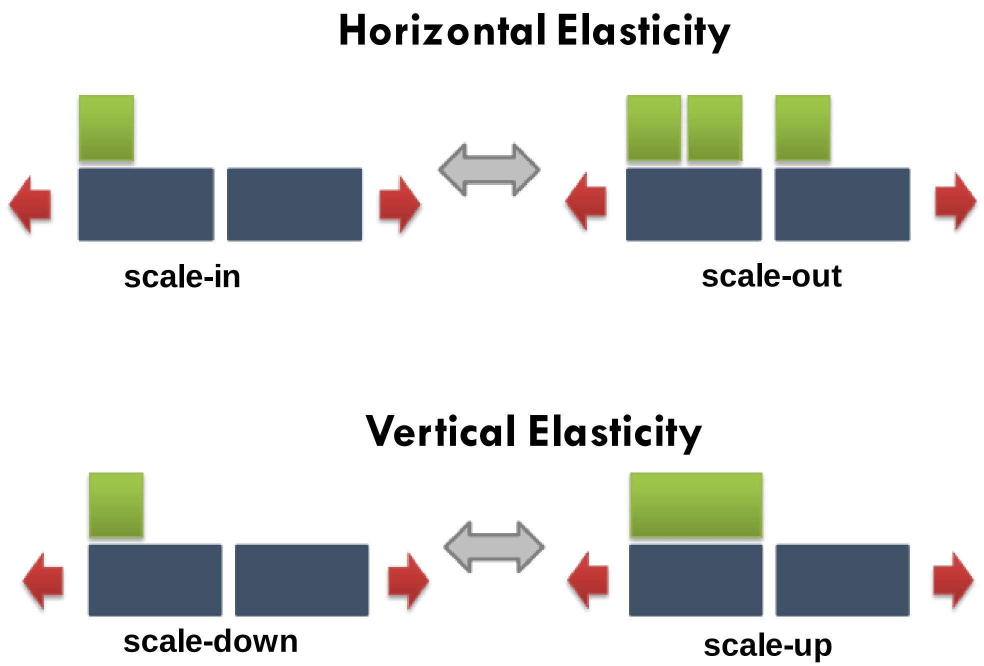
*June 10th, 2019*
*Caserta, Italy*

# ELASTICITY AND PLACEMENT PROBLEM

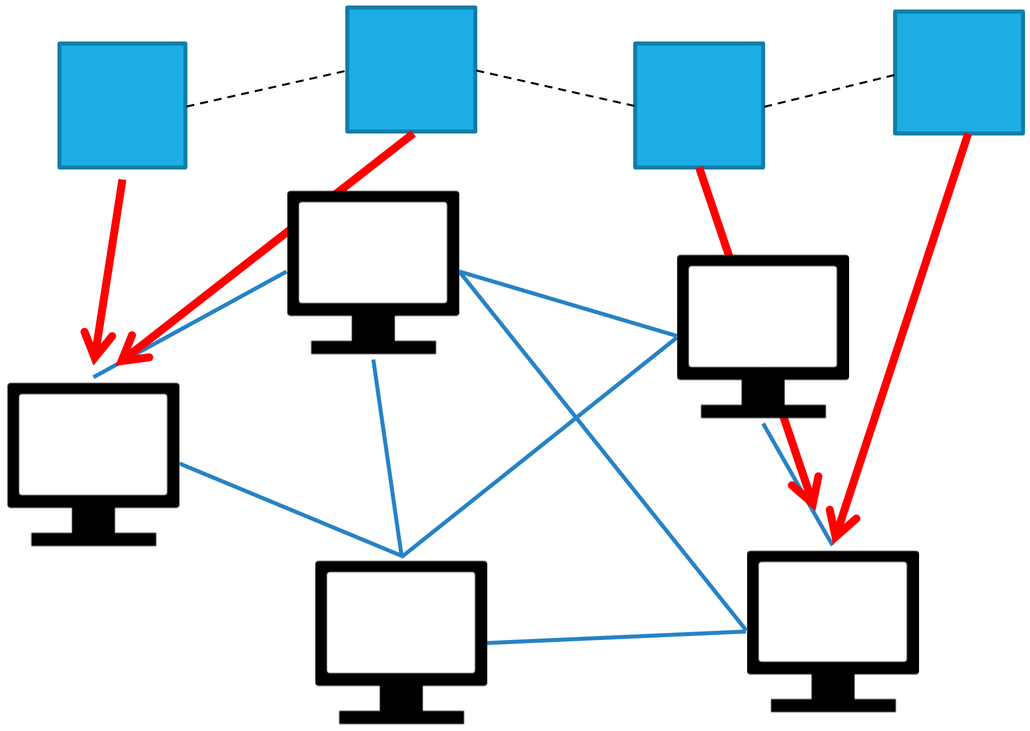
- **New scenario:** IoT edge/fog computing
- **New challenges:** heterogeneity of computing resources and dynamism of working conditions
- **New requirement:** adapt at run-time the application deployment
  - Elasticity problem
  - Placement problem
- **New software architectures:** container-based architectures

# ELASTICITY AND PLACEMENT PROBLEM

## Elasticity Problem




## Placement Problem



# RESEARCH QUESTIONS

How can we model the placement and the elasticity problem?

How do network latencies affect performances?

What is the role of QoS attributes in determining the container-based application deployment?

What are the challenges of deploying containers in a geo-distributed enviroment?

How can containers be efficiently deployed to work in presence of mobile devices?

How can the containerized deployment model be customized to represent features and requirements of a different context?

# RESEARCH NEEDS

Need of orchestration framework which can

execute containerized applications in a heterogeneous and geo-distributed environment

be equipped with centralized and decentralized deployment policies

provide adaptation capacities

# HORIZONTAL AND VERTICAL SCALING OF CONTAINER-BASED APPLICATIONS USING REINFORCEMENT LEARNING *

How can we model the elasticity problem?

Main contribution:

- Autonomic elasticity of container-based applications
  - Horizontal or Vertical scaling
  - Horizontal and Vertical scaling

Reinforcement Learning algorithms:

- Q-learning
- Model-based

* proc. of the IEEE CLOUD 2019 - Milan, Italy, July 2019.

# SYSTEM DEFINITION

Per-application RL agent

RL agent adapts:

- Number of containers
- Amount of resources assigned to each application container

*Application state:* $s = (k, u, c)$

Goal: minimize the deployment cost

Action carried out in the state s: *a*

- Horizontal or vertical scaling (5 action model)
- Horizontal and vertical scaling (9 action model)

# IMMEDIATE COST

What is the role of QoS attributes in determining the container-based application deployment?

**Immediate Cost**

- Cost of carrying out action a when the application state transits from s to s'

**Formal definition**

- Weighted sum of the
  - Adaptation cost
  - Performance penalty
  - Resource cost

# REINFORCEMENT LEARNING ALGORITHMS

$Q(s, a)$: estimate of long-term cost due to the execution of action $a$ in $s$

**Q-learning**

- uses $Q(s, a)$ to choose the action to be performed in state $s$
- action selection policy: ε-greedy
- estimates $Q(s, a)$ from experience:

$$Q(s_i, a_i) \leftarrow (1-\alpha)Q(s_i, a_i) + \alpha\left[c_i + \gamma \min_{a' \in \mathcal{A}(s_{i+1})} Q(s_{i+1}, a')\right]$$

**Model-based**

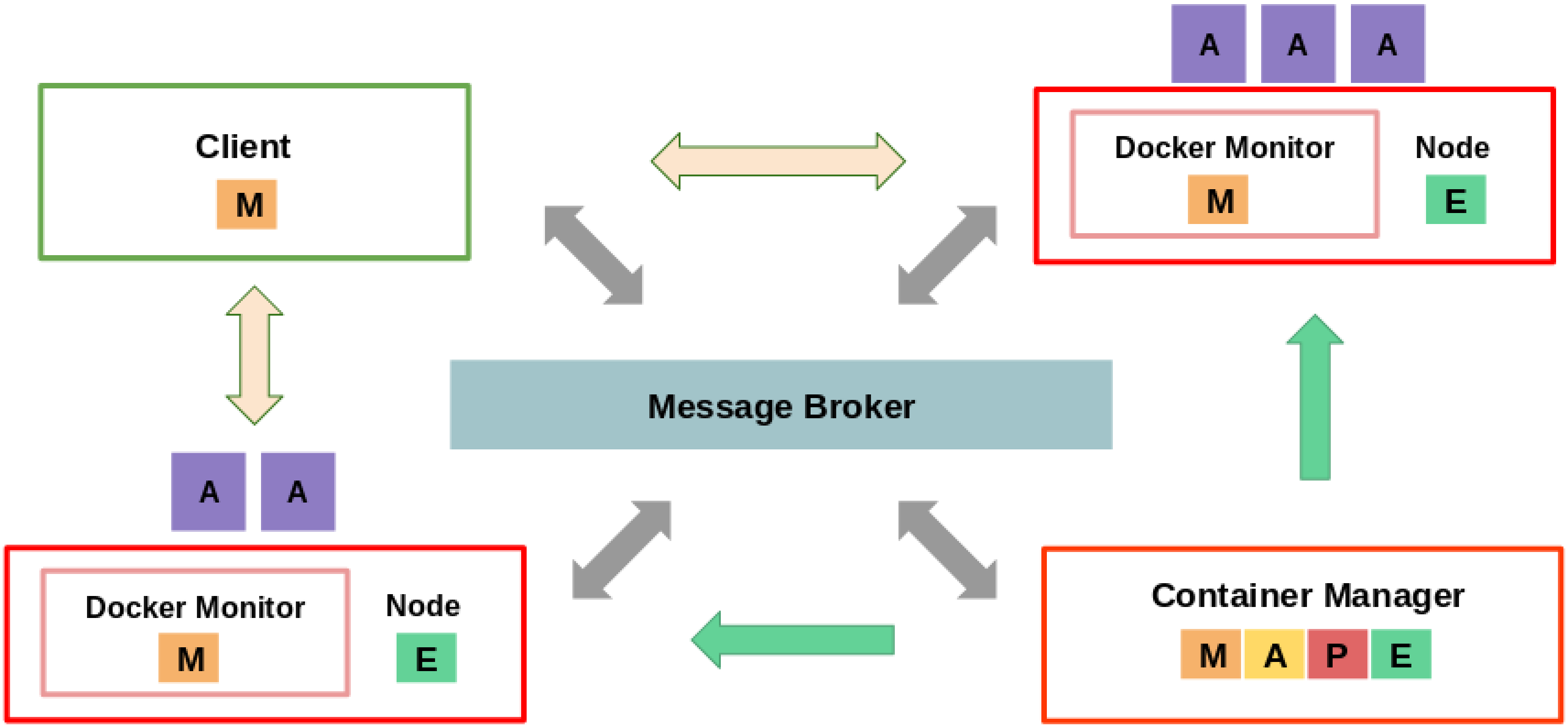
- selects the best action in terms of $Q(s, a)$
- uses Bellman equation to update $Q(s, a)$:

Unknown, but estimated

$$Q(s, a) = \sum_{s' \in \mathcal{S}} \boxed{p(s'|s, a)}\left[\boxed{c(s, a, s')} + \gamma \min_{a' \in \mathcal{A}} Q(s', a')\right] \quad \begin{array}{c}\forall s \in \mathcal{S}, \\ \forall a \in \mathcal{A}(s)\end{array}$$

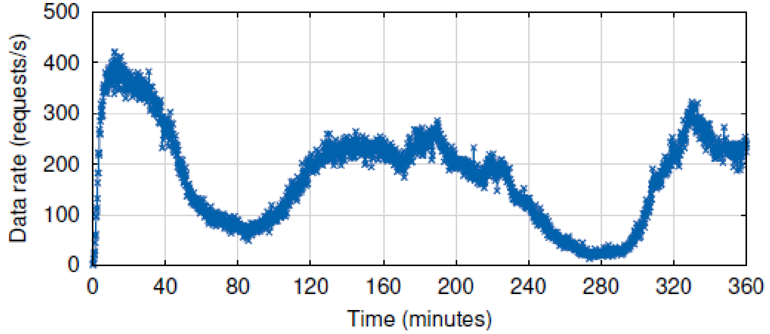
# ELASTIC DOCKER SWARM (EDS)

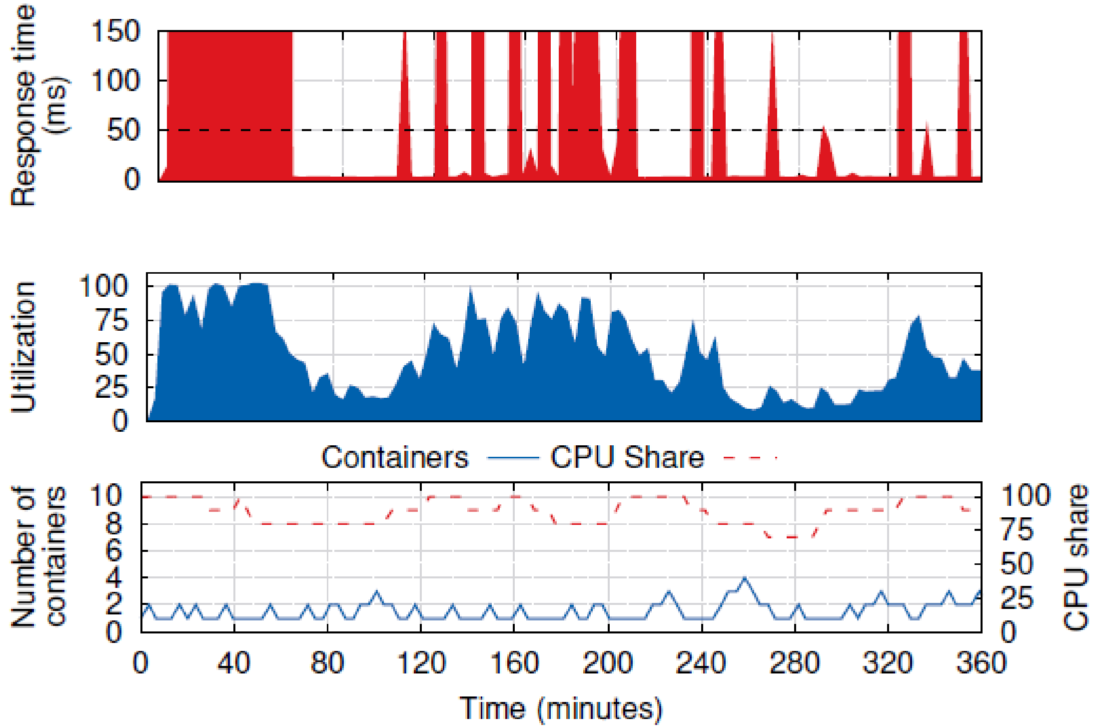
Need of orchestration framework



# EXPERIMENTAL RESULTS

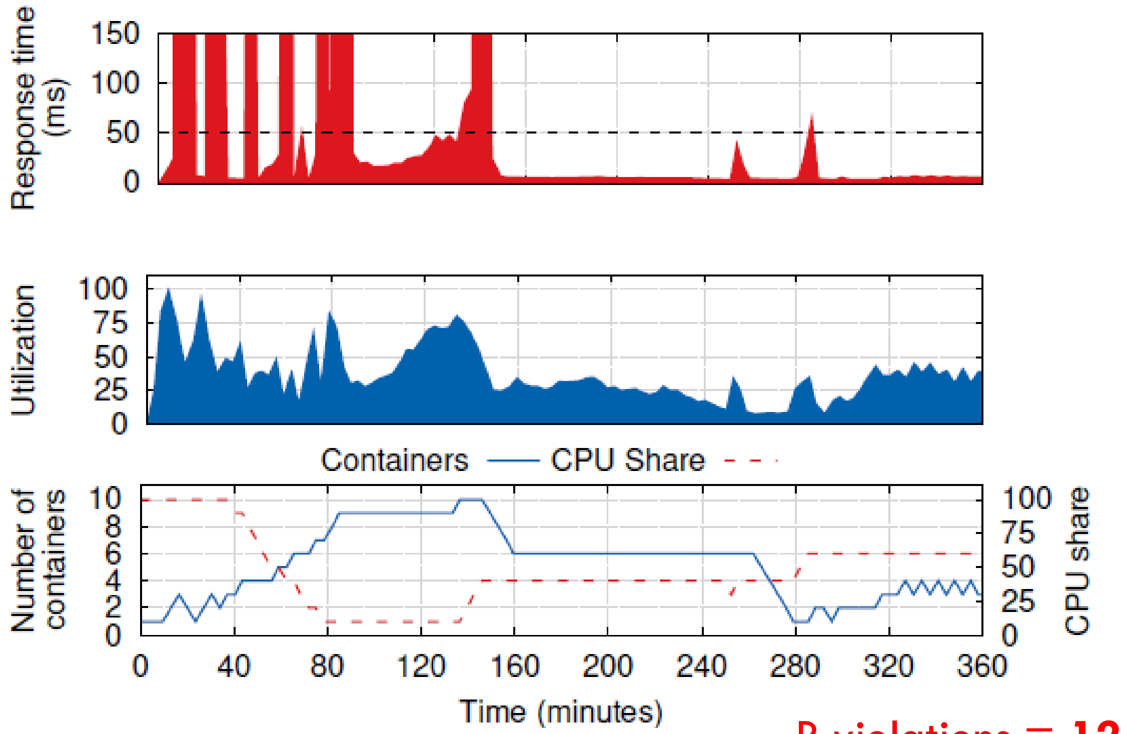
Workload used in the prototype-based experiments.




(a) Q-learning

R violations = **30%**
Adaptations = **66%**




(b) Model-based

R violations = **12%**
Adaptations = **51%**

Application performance using the 5-action adaptation model and weights $w_{\text{perf}} = 0.90$, $w_{\text{res}} = 0.09$, $w_{\text{adp}} = 0.01$.

# ELASTIC DEPLOYMENT OF SOFTWARE CONTAINERS IN GEO-DISTRIBUTED COMPUTING ENVIRONMENTS *

How can we model the placement and the elasticity problem?

What are the challenges of deploying containers in a geo-distributed enviroment?

How do network latencies affect performances?

* proc. of IEEE ISCC 2019 - Barcelona, Spain, July 2019.

# TWO-STEP DEPLOYMENT ADAPTATION POLICY

**First Step**

- Exploit horizontal and vertical elasticity of containers by means of RL-based policies
  - Q-learning
  - Model-based

**Second Step**

- Determine the container placement on geo-distributed environment:
  - ILP formulation
  - Network-aware heuristic

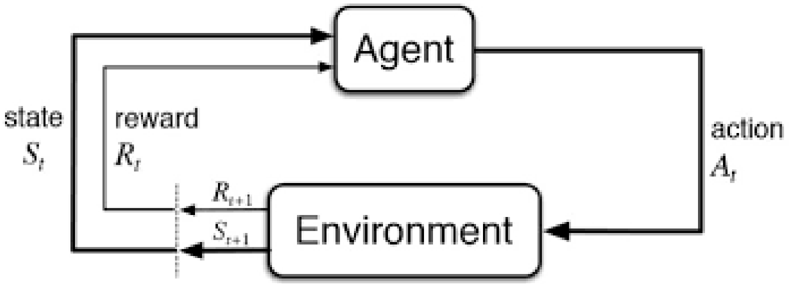
# SOLUTION OVERVIEW

We take into account

First Step

- time needed to deploy every container in VMs
- number of application instances
- assigned CPU share
- application performance requirements
  - expressed in terms of response time percentile

**Elasticity Problem**




$$c(s,a,s') = w_{\text{perf}} \mathbb{1}_{\{R(k+\tilde{k},u',c+\tilde{c})>R_{\max}\}} + w_{\text{res}} \frac{(k+\tilde{k})(c+\tilde{c})}{K_{\max}} + w_{\text{adp}} \frac{D}{D_{\max}}$$

# SOLUTION OVERVIEW

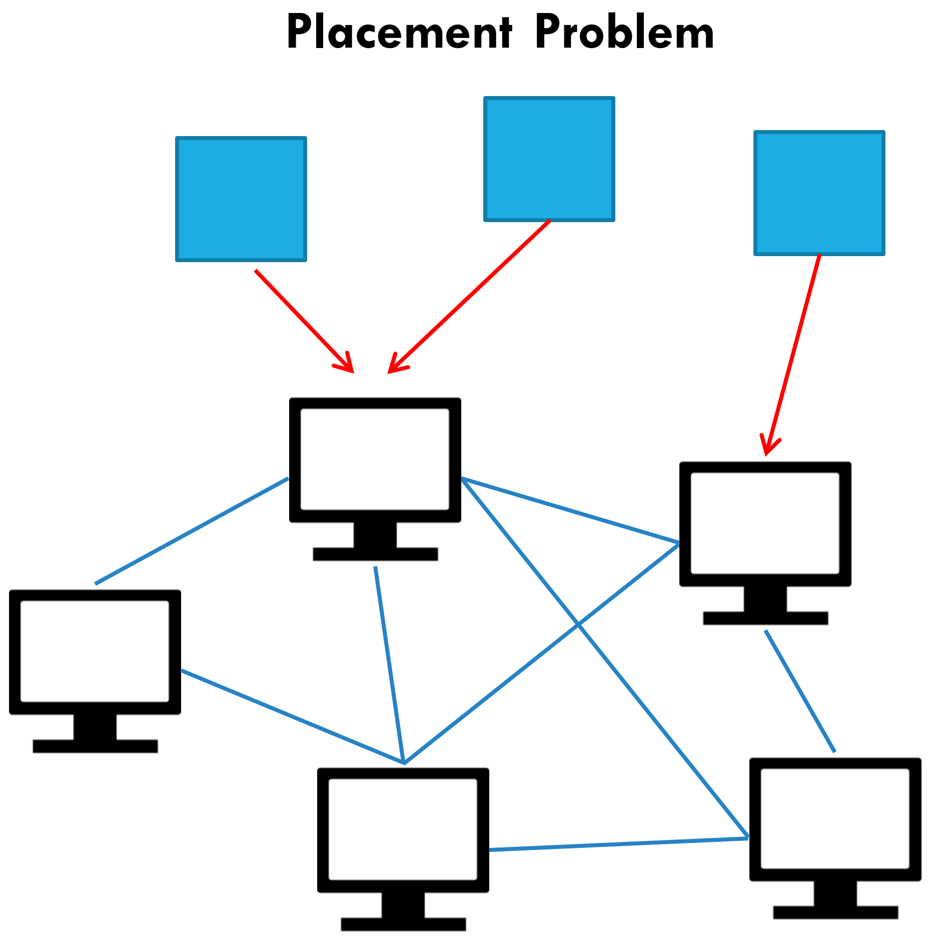
Second Step

We take into account

- network delay between VMs
- number of active VMs
- time needed to deploy every container in VMs

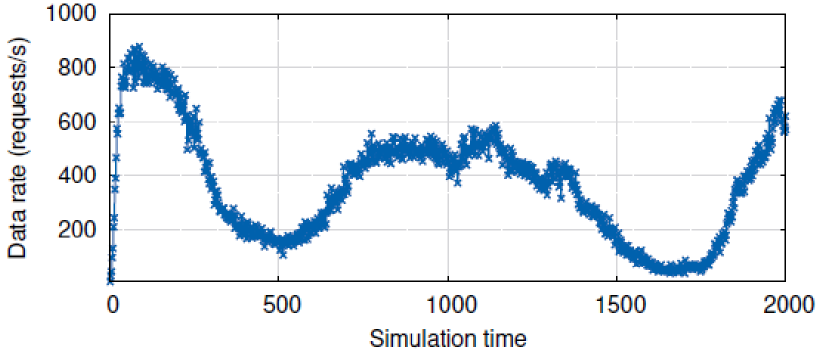
Solution proposed

- ILP formulation
- Network-aware heuristic

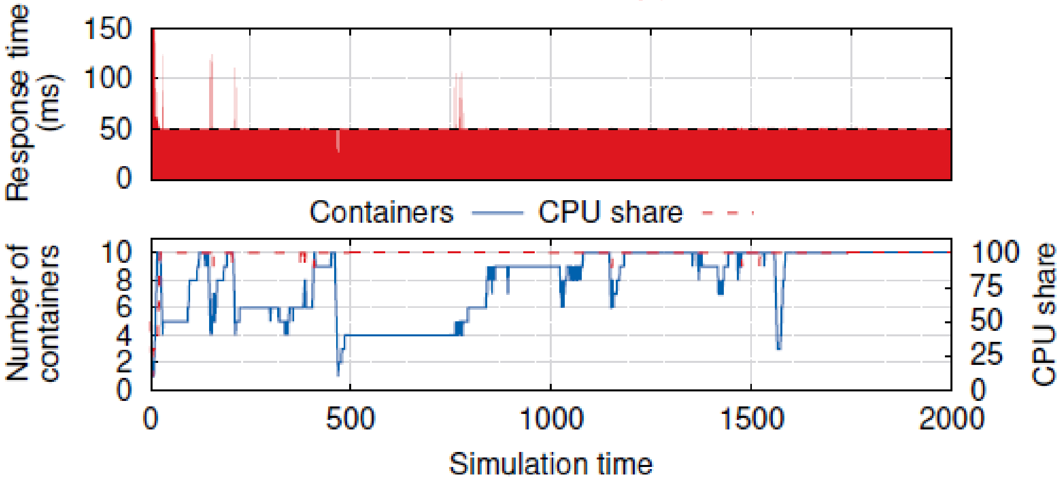


# EXPERIMENTAL RESULTS

Application workload used in simulation.

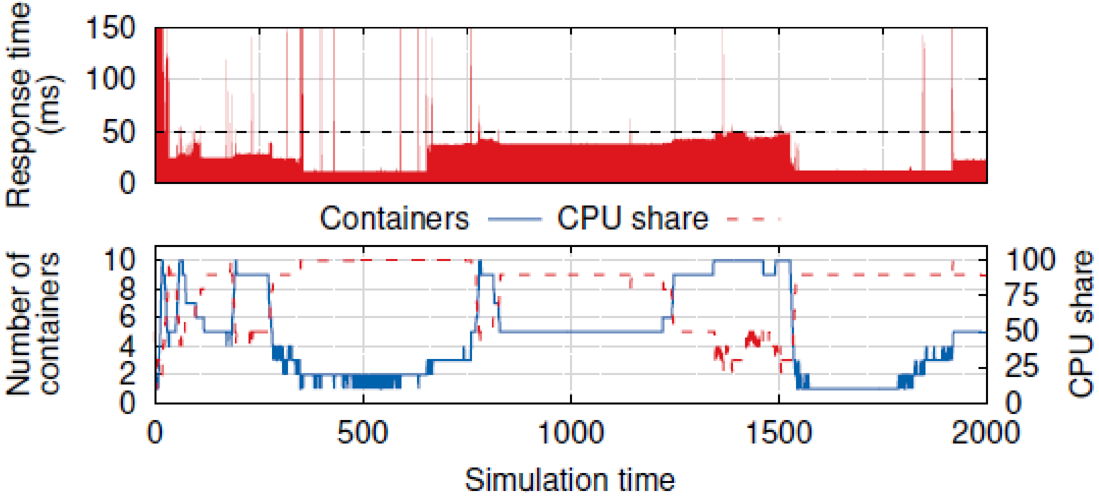



R violations = **7%**




Round Robin heuristic

R violations = **2.80%**

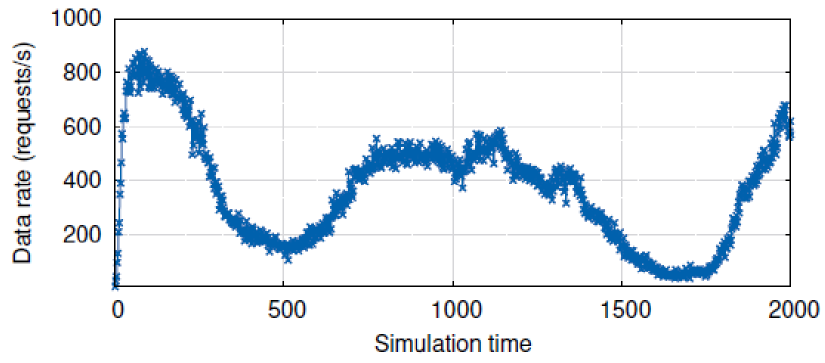



Optimal placement policy (OPT)

Application performance and run-time deployment adaptation ($w_{\text{perf}} = 0.90$, $w_{\text{res}} = 0.09$, $w_{\text{adp}} = 0.01$).

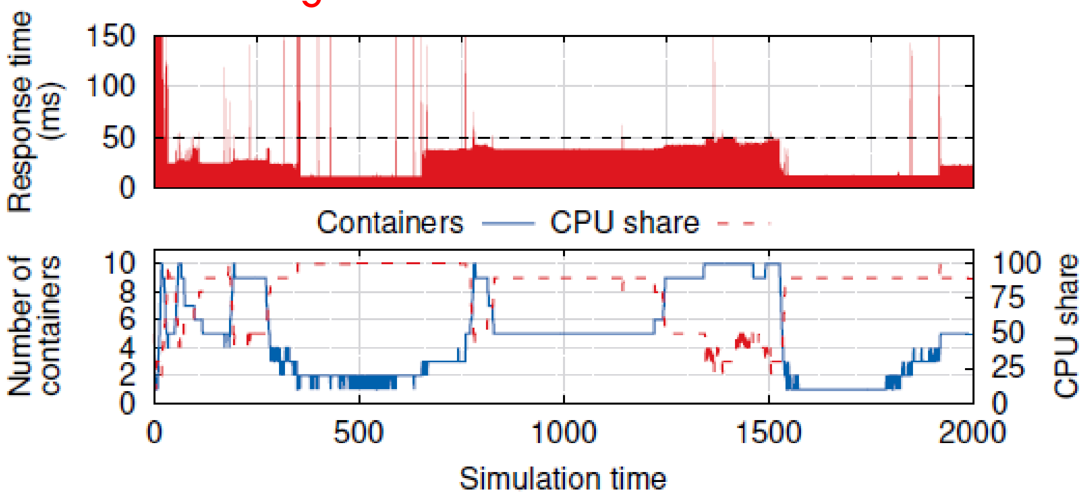
# EXPERIMENTAL RESULTS

Application workload used in simulation.




R violations = **2.80%**

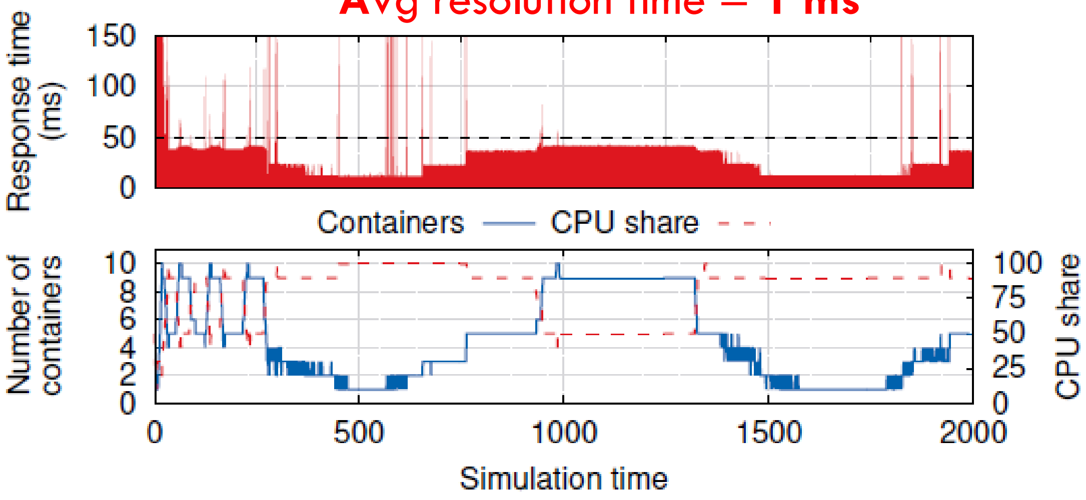
Avg resolution time = **51 ms**




Optimal placement policy (OPT)

**R violations = 3%**

**Avg resolution time = 1 ms**




Network-aware greedy heuristic (NetAware)

Application performance and run-time deployment adaptation ($w_{\text{perf}} = 0.90$, $w_{\text{res}} = 0.09$, $w_{\text{adp}} = 0.01$).

# RESEARCH QUESTIONS

How can we model the placement and the elasticity problem?

How do network latencies affect performances?

What is the role of QoS attributes in determining the container-based application deployment?

What are the challenges of deploying containers in a geo-distributed enviroment?

How can containers be efficiently deployed to work in presence of mobile devices?

How can the containerized deployment model be customized to represent features and requirements of a different context?

# FUTURE WORKS

- Placement Heuristics
- Multi-level adaptation
- Multi-component adaptation
- IoT and Mobility
- Elastic Docker Swarm in Fog environment

# THANK YOU!

**Fabiana Rossi**
University of Rome Tor Vergata

f.rossi@ing.uniroma2.it
http://www.ce.uniroma2.it/~fabiana/