

# Monitor-Based Testing of Elastic Cloud Computing Applications

**Michel Albonico**

PhD Student - AtlanMod - EMN (Nantes, France)

[michel.albonico@inria.fr](mailto:michel.albonico@inria.fr)

Jean-Marie Mottu

Gerson Sunyé

# Outline

- Cloud Computing Elasticity
- Motivation
- Test Procedure
- Experiments
- Conclusion and Future Work

# Cloud Computing Elasticity

- Cloud computing elasticity:

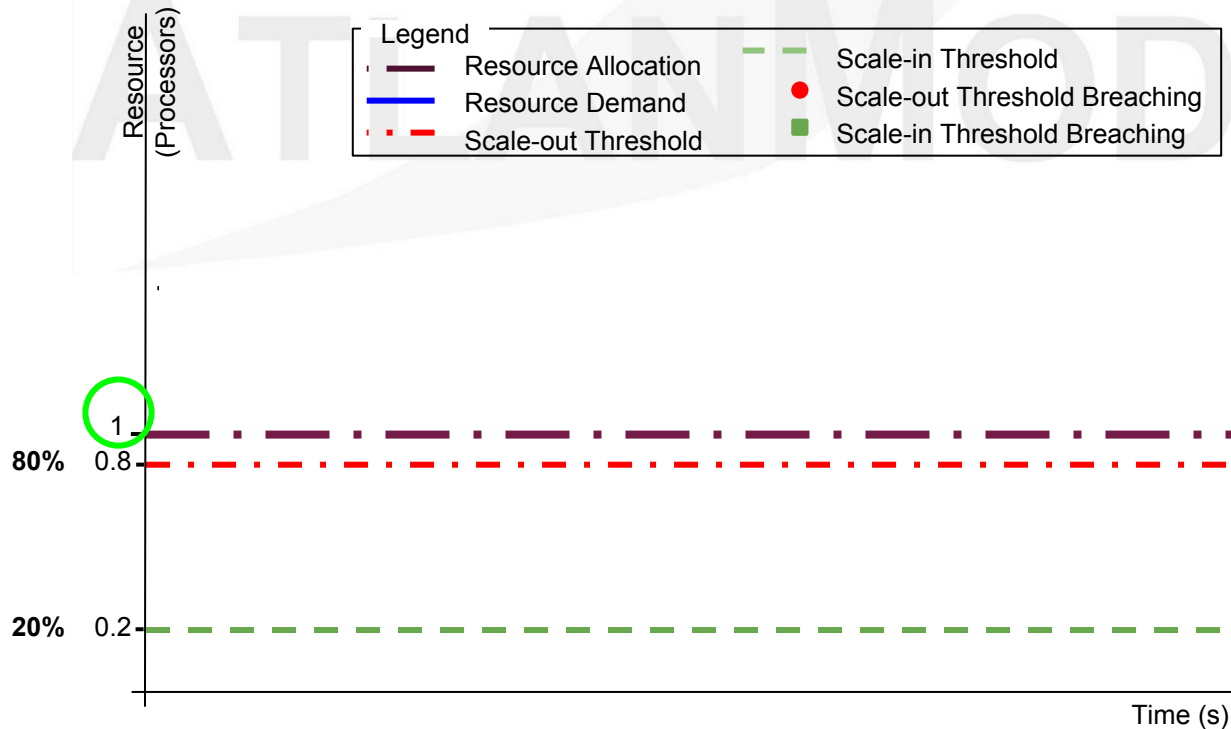
The ability of a cloud infrastructure/system modifying its resource configuration according to demand.

ATLANMOD

# Cloud Computing Elasticity

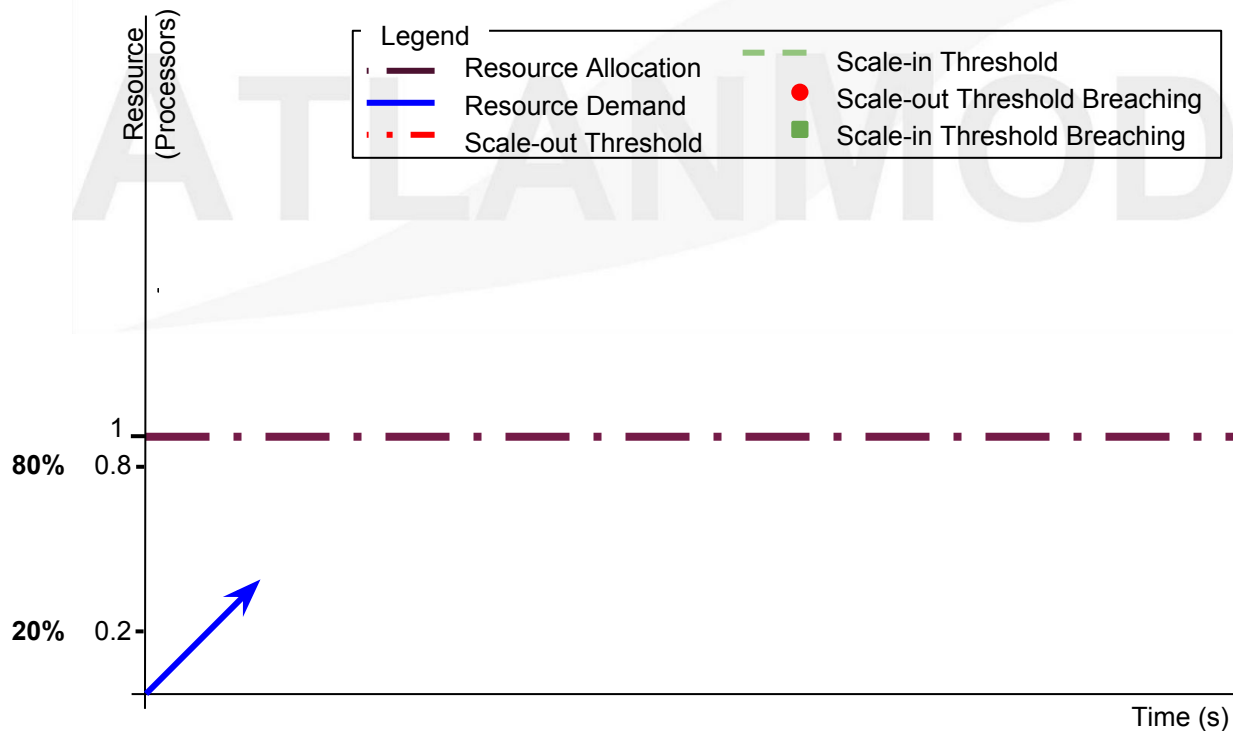
- **Thresholds:**

- Scale-out threshold: maximum resource usage, e.g., 80% of CPU usage;
- Scale-in threshold: minimum resource usage, e.g., 20% of CPU usage;
- Used to decide when varying a resource.



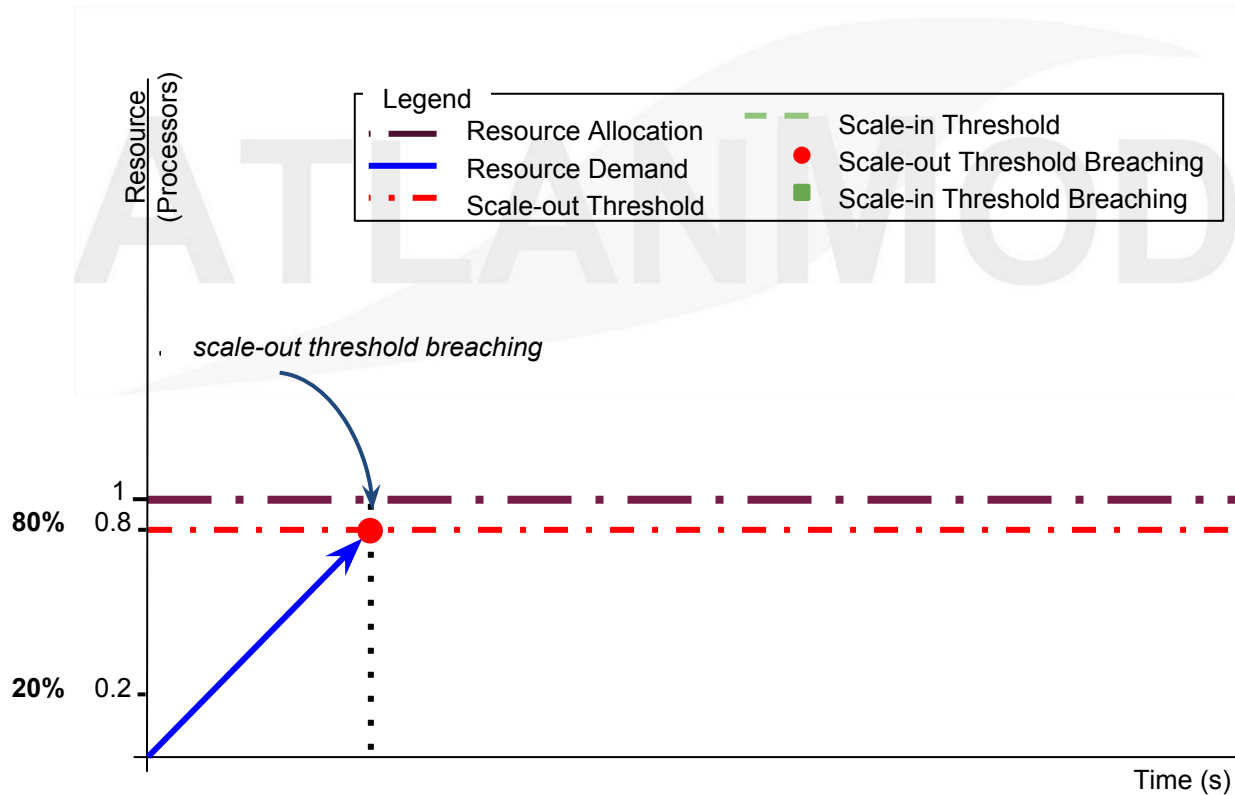
# Cloud Computing Elasticity

- Resource demand varies according to workload variations.
  - Example:
    - number of users increases from 1 to 2, the resource demand doubles.



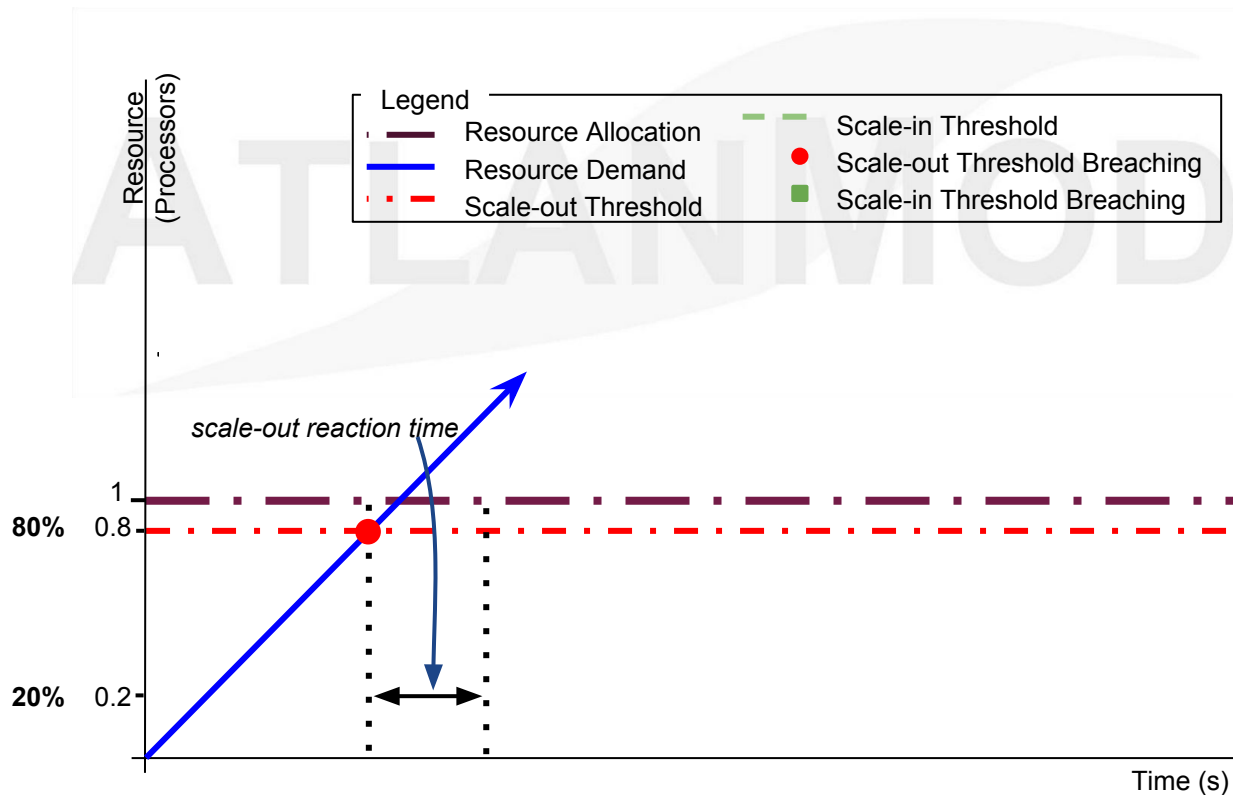
# Cloud Computing Elasticity

- Resource demand varies over time;
- **Scale-out threshold breaching.**



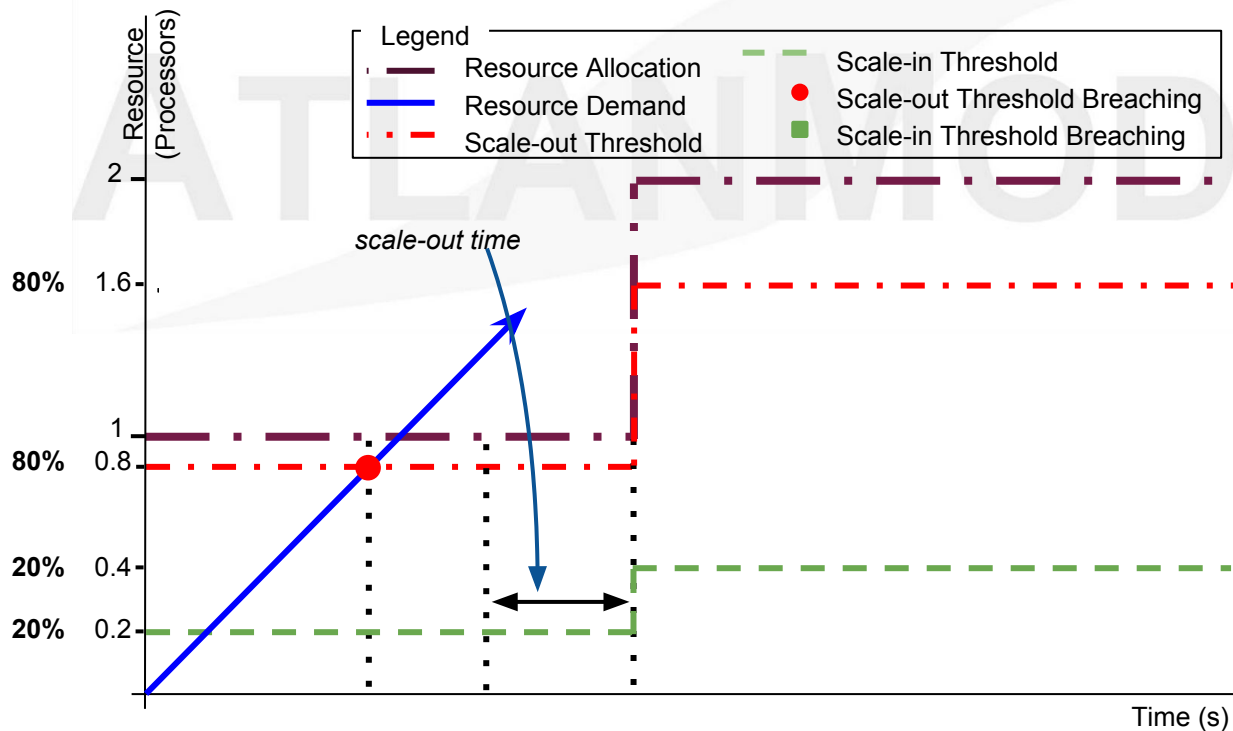
# Cloud Computing Elasticity

- Resource demand varies over time;
- Scale-out threshold breaching;
- **Scale-out reaction time;**



# Cloud Computing Elasticity

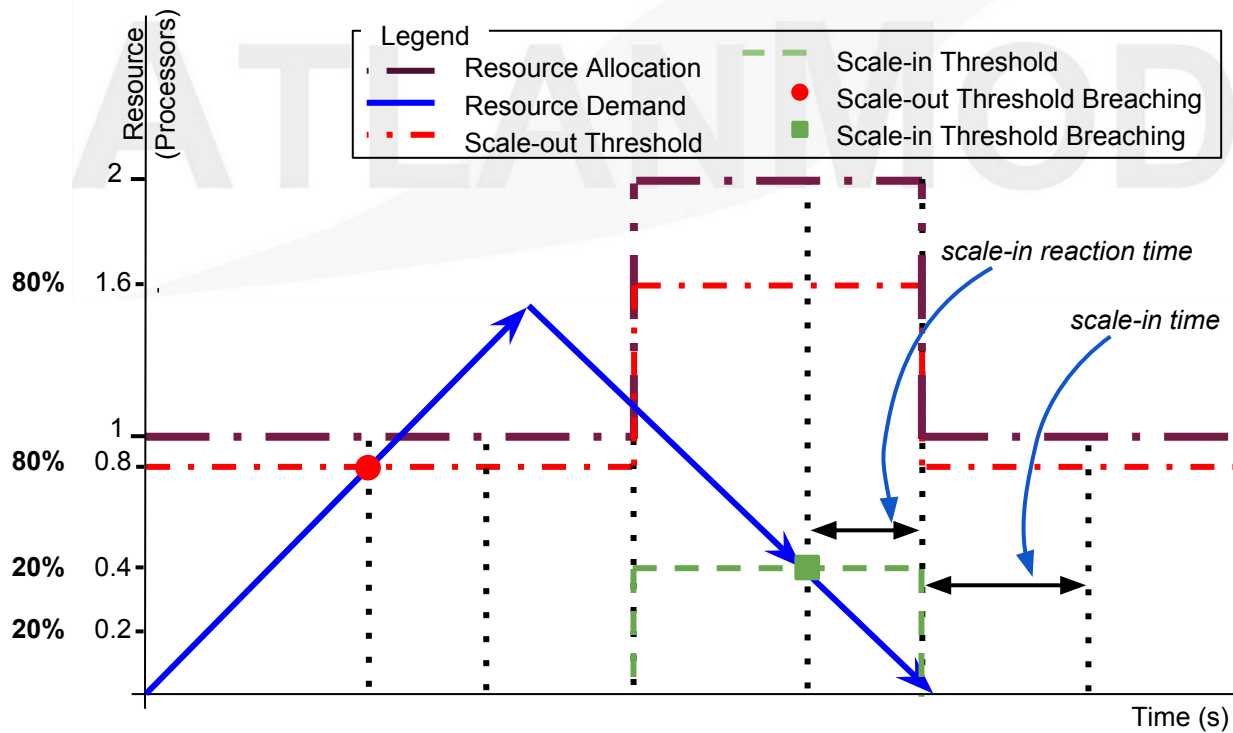
- Resource demand varies over time;
- Scale-out threshold breaching;
- Scale-out reaction time;
- **Scale-out time, then the thresholds are updated.**





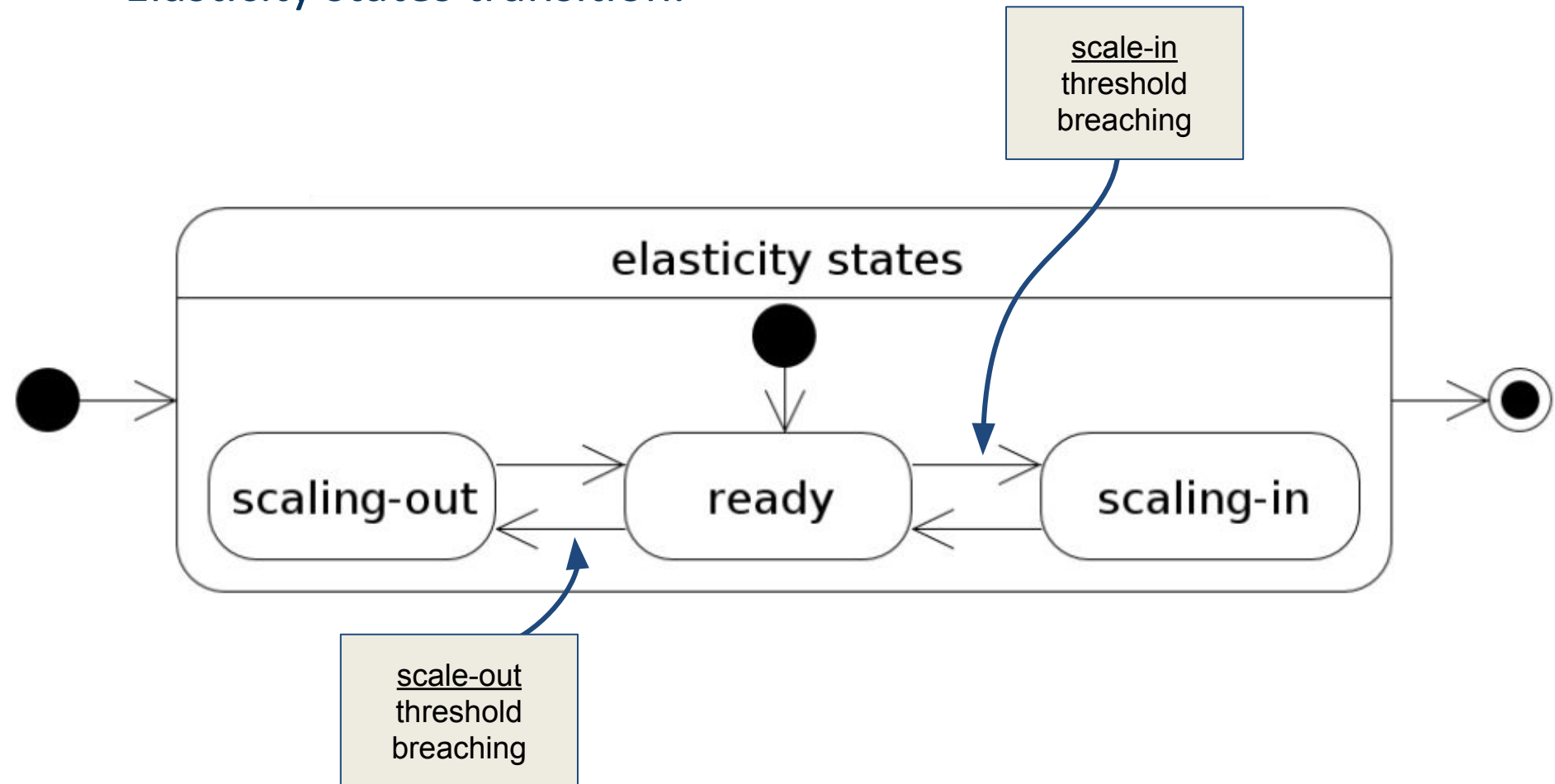
# Cloud Computing Elasticity

- Scale-in:
  - Scale-in threshold breaching;
  - Scale-in reaction time (**resource is no longer available**);
    - Thresholds reconfiguration.
  - Scale-in time.



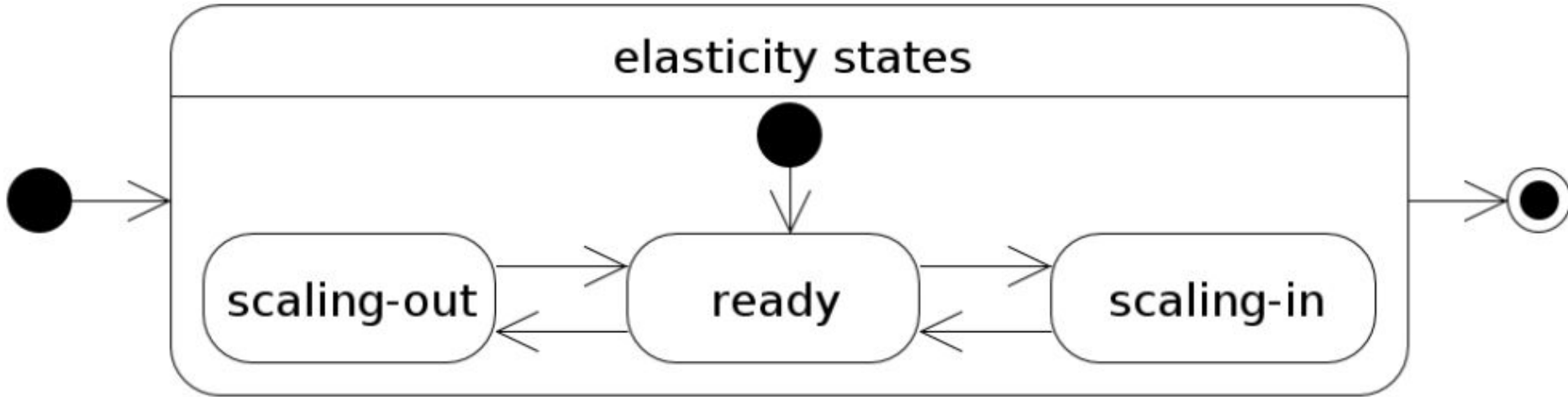
# Cloud Computing Elasticity

- Elasticity states transition.



# Motivation

- Elasticity states transition.



- Related work only test during the ready state;
- Scaling states:
  - Considerable time: in our experiments, scaling-out takes more than 90 seconds (Amazon EC2);
  - Great part of the adaptation tasks: replication data, leader election, etc.

# Test Procedure

- Test cloud systems during all the elasticity states;
- Execute tests dynamically:
  - Associate test cases to a set of elasticity states;
  - Execute the test according to the current elasticity state.
- Test execution:
  - Periodically monitor the resource during the test execution;
    - Current elasticity state.
  - (Re)-execute the associated test cases **during** the current elasticity state.

# Experiments

- **Research questions (answered by the experiments):**
  1. Is it necessary to run the test during different elasticity states?
    - a. Does a cloud system react distinctly depending on the elasticity state?
  2. Is it possible to execute the test during different elasticity states and to assign the test verdicts accordingly?

# Experiments

## Question 1: system behavior during different elasticity state.

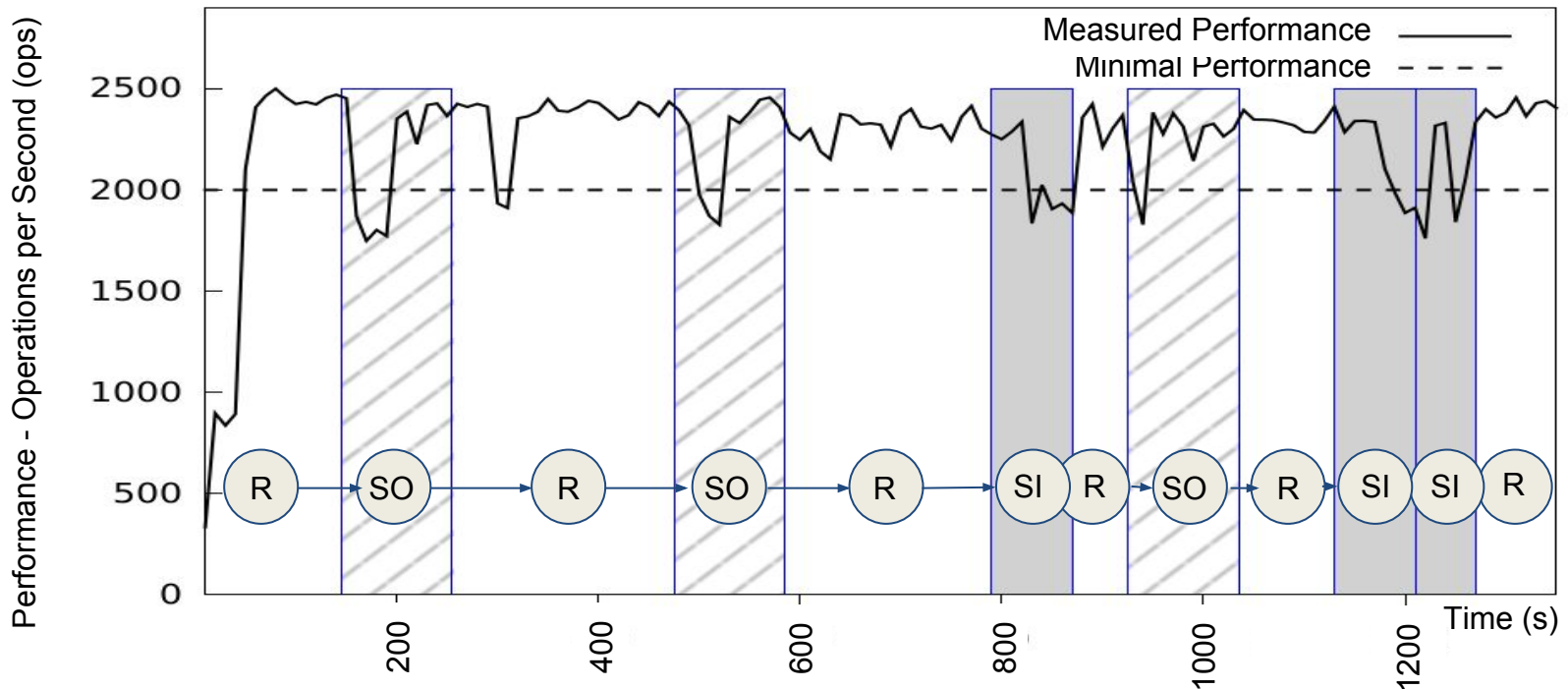
- First experiment:  
**Measure the performance of a cloud system during different elasticity states.**
  - Manually executed;
  - Workload (50% read / 50% write);
  - 2500 operations per second (ops).

# Experiments

- First experiment results:
  - 2000 ops: covers all the performance drops;
  - Elasticity states extracted from the log files;

RQ1: ✓

It is necessary to run the test during different elasticity states.



# Experiments

## Question 2: test execution during different elasticity states + test verdicts assignment.

- Second experiment: (same workload)
  - We use our test procedure;
  - We monitor the elasticity states throughout the test execution;
  - Test Case:
    - answered operation  $\geq 2000$  ops -> pass
    - otherwise -> fail
  - Same test case associated to every elasticity state.
    - Test case re-executed throughout the cloud system execution.

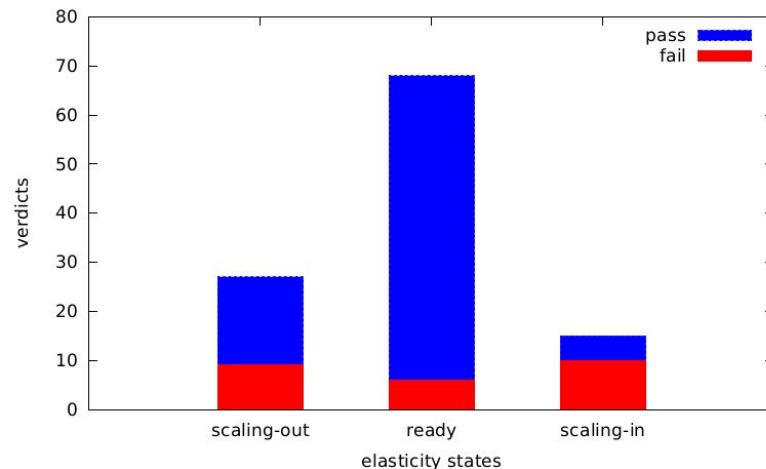


# Experiments

- Result of the second experiment:
  - **Test through different elasticity states;**
  - **Assign test verdicts to different elasticity states;**
    - **Proportional to the previous experiment (correct elasticity states).**

RQ2: ✓

It is possible to execute the test according to the elasticity state, and we are able to assign the test verdicts correctly.



fr)

# Conclusion and Future Work

- Identify all the performance problems;
- Assign the test verdicts to the correct elasticity states (at runtime);
- Address the scaling states, which are not addressed by related work;
- Future work:
  - Write functional test cases;
  - Apply to other study cases;
  - Generate test cases based on elasticity states.

# Monitor-Based Testing of Elastic Cloud Computing Applications

**Michel Albonico**

PhD Student - AtlanMod - EMN (Nantes, France)

([michel.albonico@inria.fr](mailto:michel.albonico@inria.fr))

Jean-Marie Mottu

Gerson Sunyé