



**HAL**  
open science

# Autonomic Approach to the Runtime Management of HPC Cluster Resources

Quentin Guilloteau

► **To cite this version:**

Quentin Guilloteau. Autonomic Approach to the Runtime Management of HPC Cluster Resources. LIG PhD Day, Apr 2022, Grenoble, France. hal-04570283

**HAL Id: hal-04570283**

**<https://hal.science/hal-04570283v1>**

Submitted on 7 May 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Autonomic Approach to the Runtime Management of HPC Clusters Resources

Quentin GUILLOTEAU, Olivier RICHARD, Eric RUTTEN

Univ. Grenoble Alpes, Inria, CNRS, LIG, F-38000 Grenoble France

## Context & Applicative Problem

### Gridad

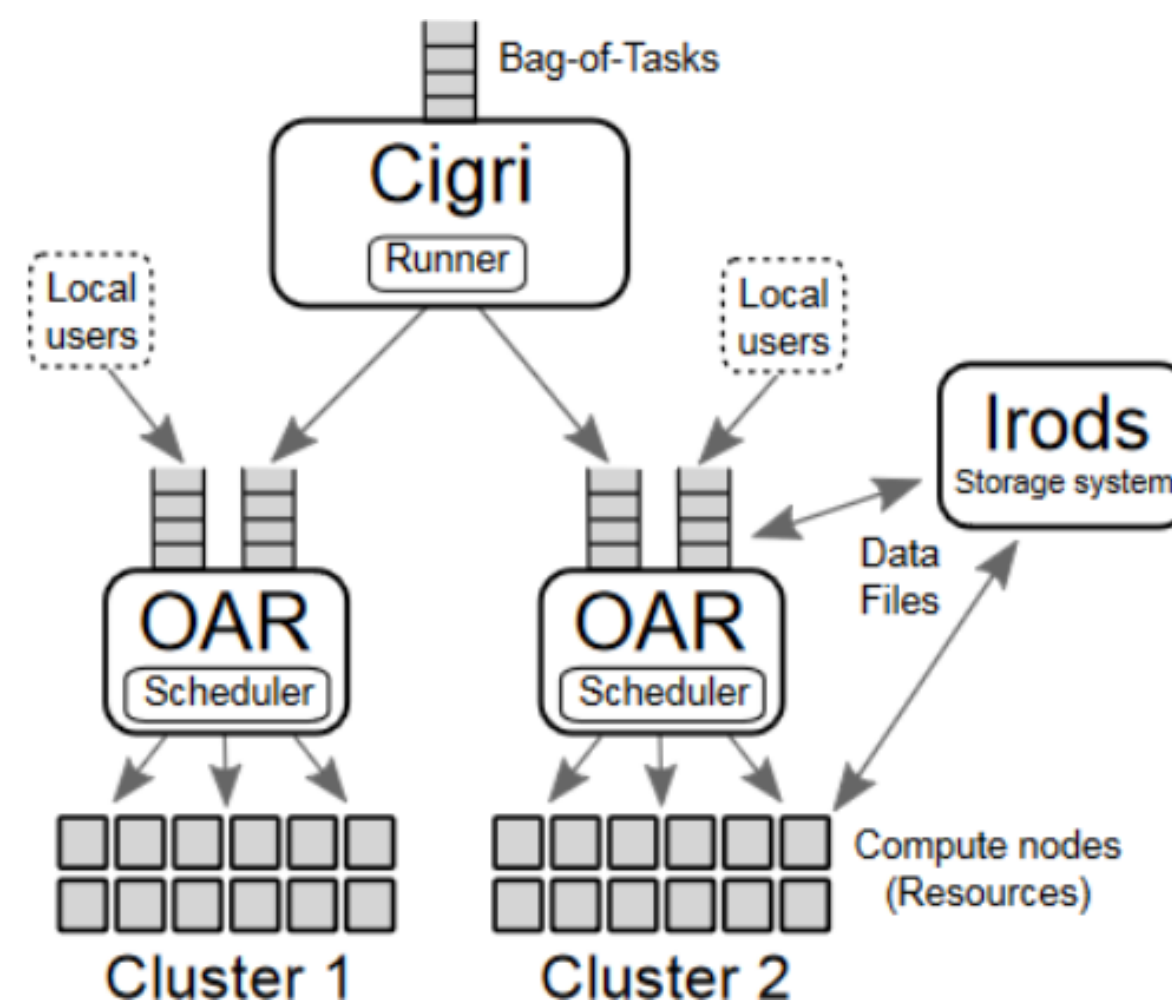


- Grenoble Mesocenter
- 169 TFLOPS
- 5700 CPUs on several clusters
- > 1200 users

Idle HPC Resources  $\Rightarrow$  Lost Computing Power  
 $\hookrightarrow$  **How to Harvest?**

## A Solution: CiGri

- **bag-of-tasks**: many, multi-parametric jobs
- **Best-effort Jobs**: Lowest priority
- **Objective**: Collect grid idle resources in a non-intrusive way for the users

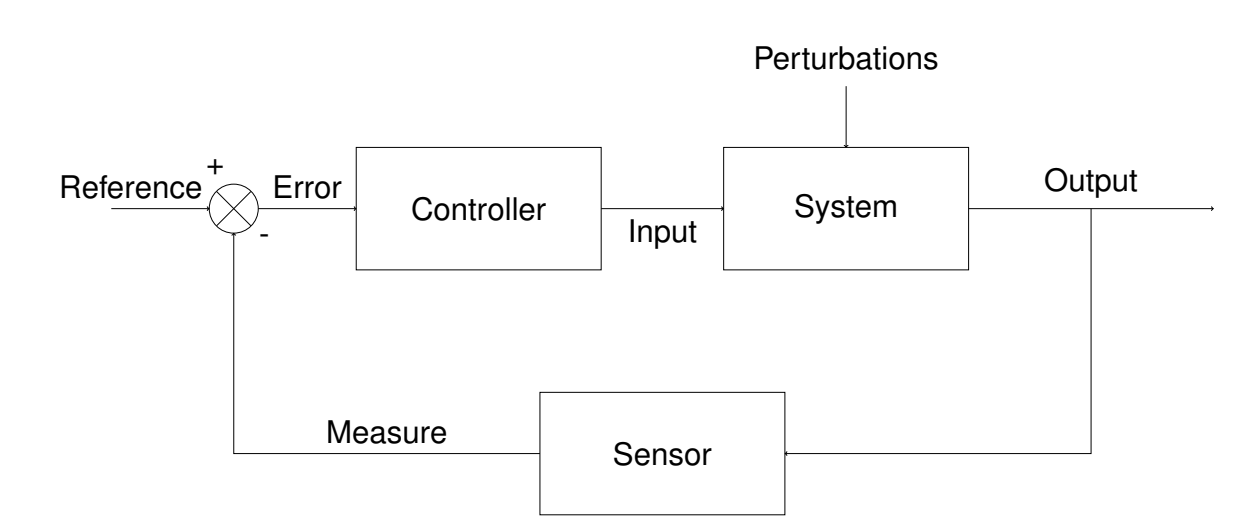
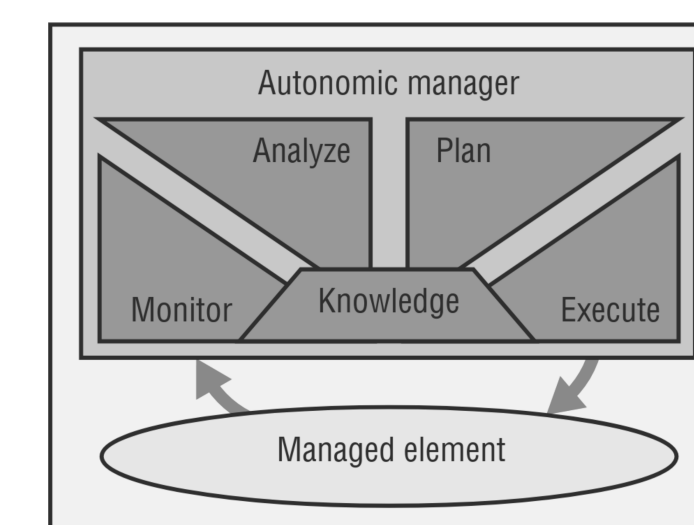


Various Job Behaviours  
 $\hookrightarrow$  Variability  $\Rightarrow$  Requires **Runtime Management**

## MAPE-K Loop & Control Theory

**Auto-regulating** Dynamical Systems given **high-level objectives**

$\hookrightarrow$  Control Theory as an Interpretation of the MAPE-K Loop



**Control Theory** drives system to a desired state by adapting inputs while **minimizing delay, overshoot, or steady-state error** and ensuring control **stability**

## Perturbations from Harvesting

Cannot simply submit huge amount of jobs.  
 $\nearrow$  Harvesting  $\Rightarrow$   $\nearrow$  Perturbations (e.g., I/O)  
 $\hookrightarrow$  **Trade-off to exploit**

## Instrumentation

- **Actuator**: #jobs to submit from CiGri to OAR
- **Sensors**: OAR Waiting Queue length, File-System Load (loadavg)

## Controllers

**Controller** = function between objective (or *Reference*) error and system input.

$$Error_k = Reference - Sensor_k$$

Different types of **Controllers** (PID, RST, MFC)

$$Actuator_{k+1} = K_p \times Error_k + K_i \times \sum_k Error_k + K_d \times (Error_k - Error_{k-1})$$

## Response to Perturbations

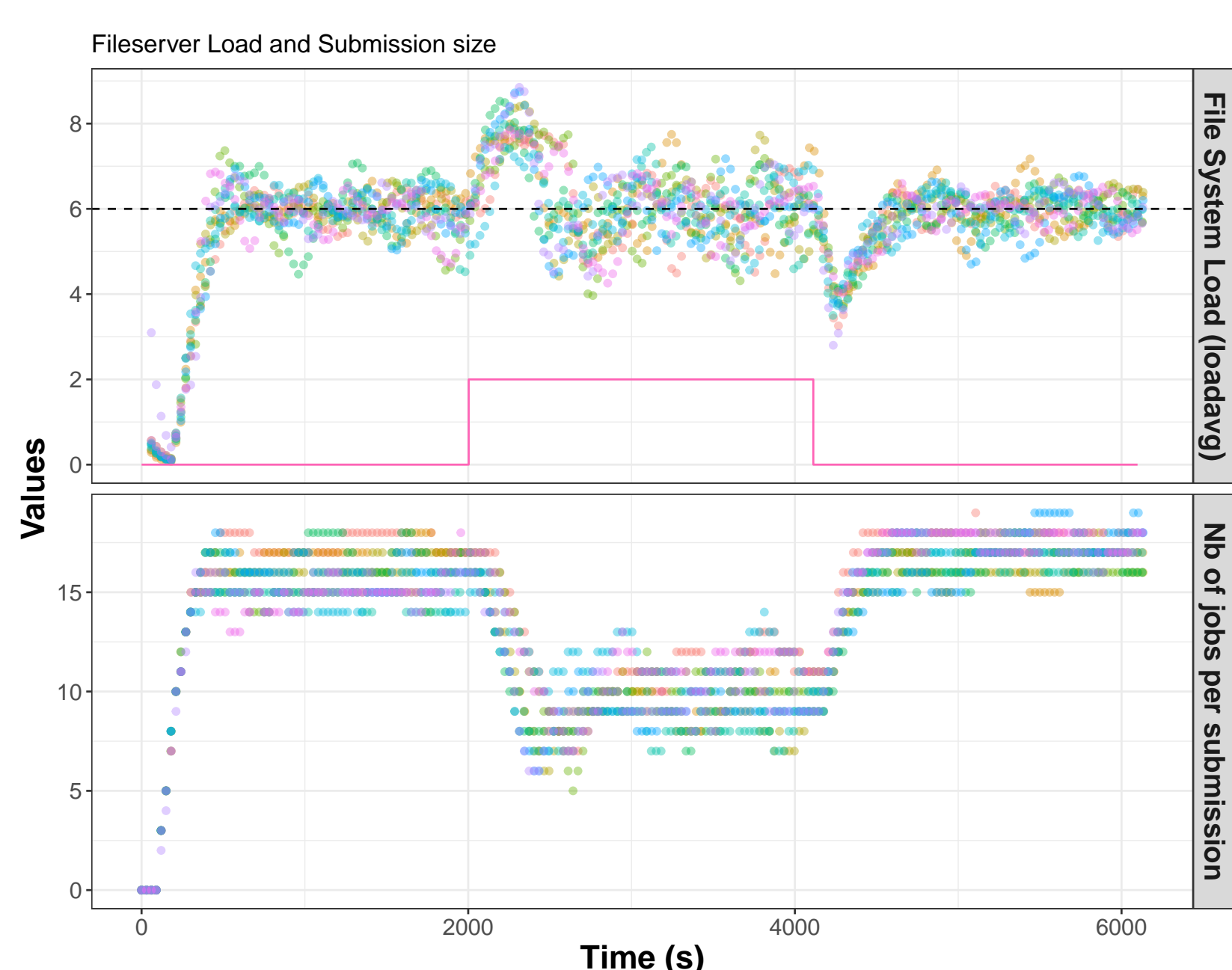


Figure: Response of the closed-loop system (top) to a synthetic step perturbation to keep the load of the File-System around the value 6. Bottom is the number of jobs submitted by CiGri.

## Exploiting the Trade-Off

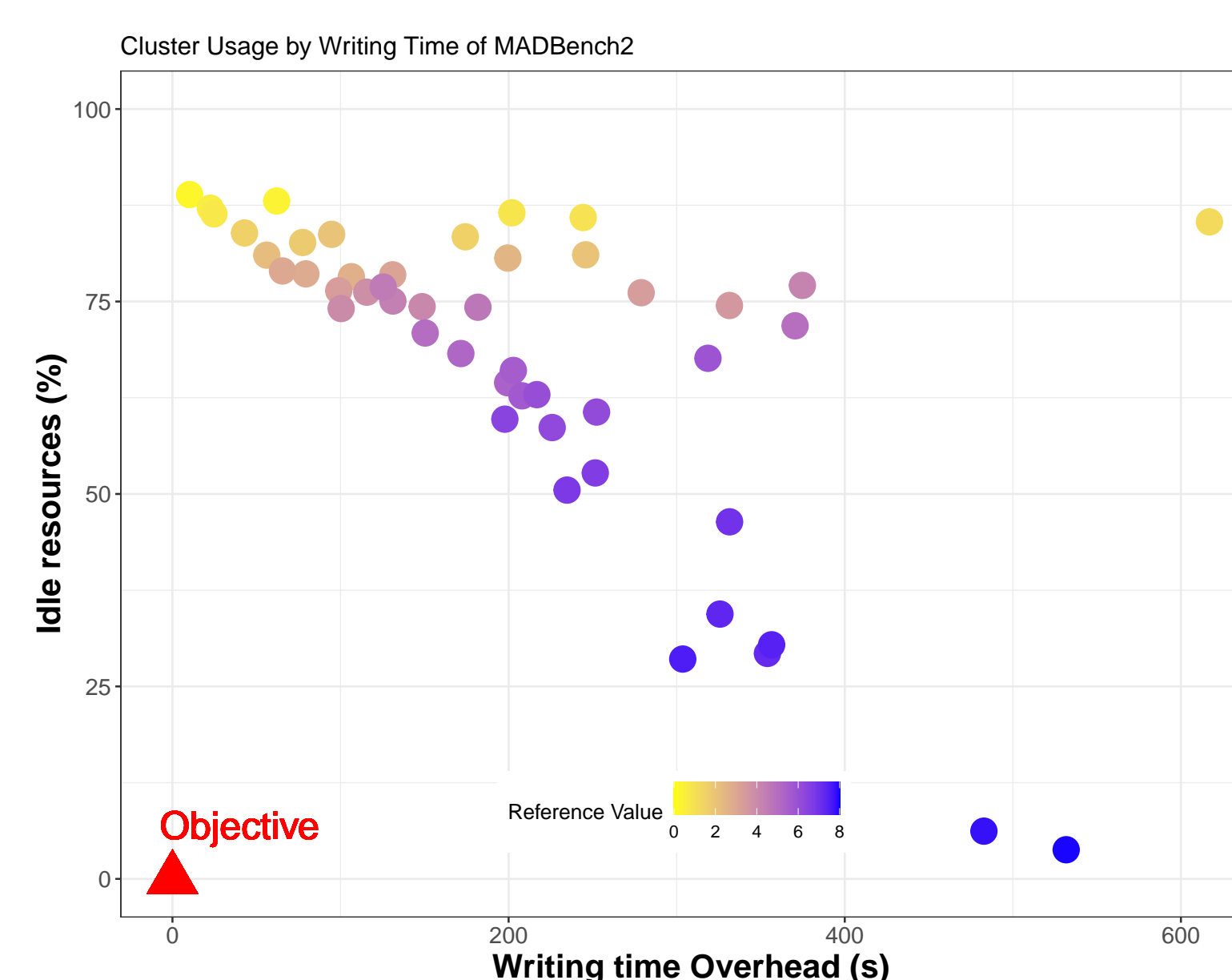
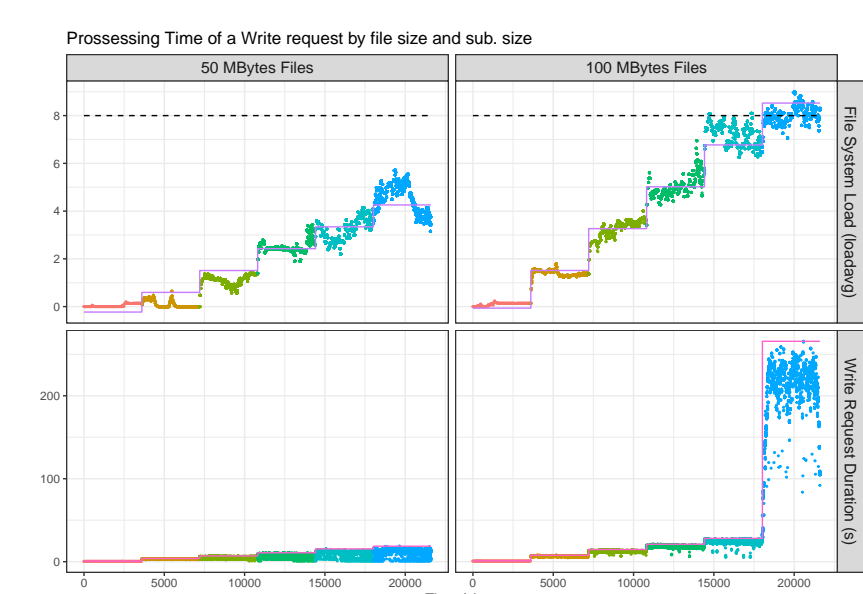


Figure: Writing Overhead of an application (MADBench2 Benchmark) with CiGri based on Reference Value

$\hookrightarrow$  We have **Control over the Degradation and Harvesting** with the choice of Reference Value

## Open-Loop & Identification



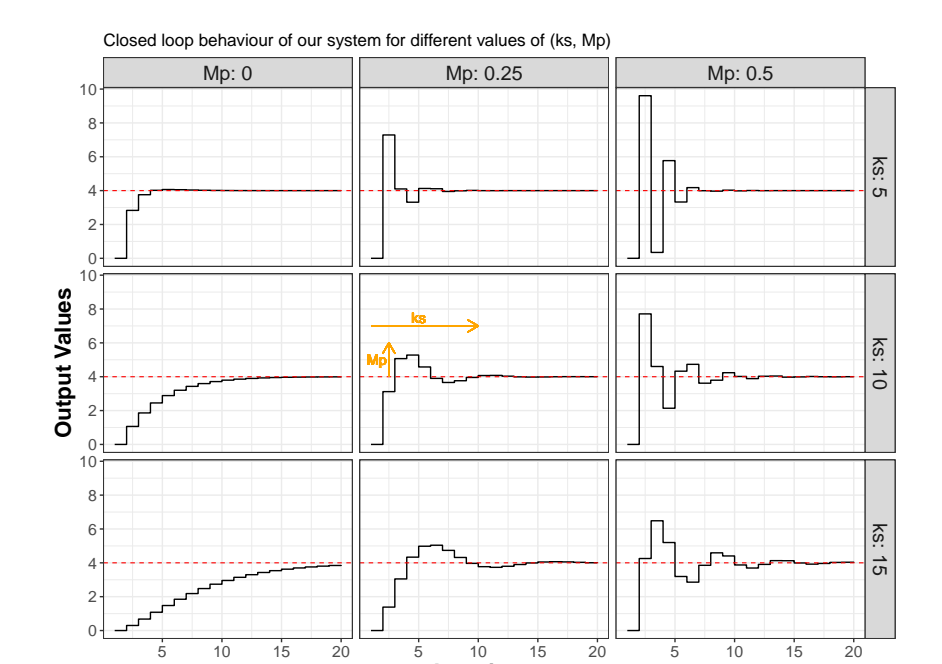
- 1 Vary nb jobs submitted
- 2 Look at FS load
- 3 Model

$\hookrightarrow$  Saturation when File-System Load > 8

## Closed-Loop Behaviour

Can pick the **desired system behaviour**:

- $k_s$ : max **time** to steady state
- $M_p$ : max **overshoot** allowed



**Non-Intrusive Harvesting** means:

- no overshoot
- "fast" response

$K_p, K_i, K_d$  are functions of the model,  $k_s$  and  $M_p$

## Perspectives

- Adaptive Controllers to deal with different I/O Profiles
- Interaction with Scheduler to anticipate the arrival of new Priority jobs
- Minimize the amount of Best-effort jobs killed (also represents lost computing power)
- Reproducible Experiments with Nix and NixOS

## References

Q. Guilloateau, O. Richard, B. Robu and E. Rutten. *Controlling the Injection of Best-Effort Tasks to Harvest Idle Computing Grid Resources*, ICSTCC 2021, hal-03363709



# CTRL-A

Control for safe Autonomic computing systems

Laboratoire d'Informatique de Grenoble



2022

Acknowledgements: Experiments presented in this poster were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see <https://www.grid5000.fr>).