Flexibility for HPC

Georges Da Costa

Associate Prof.

Toulouse III University

Partners

- Principal Investigator : LIG Grenoble
- IRIT, Toulouse
- LIP6, Paris

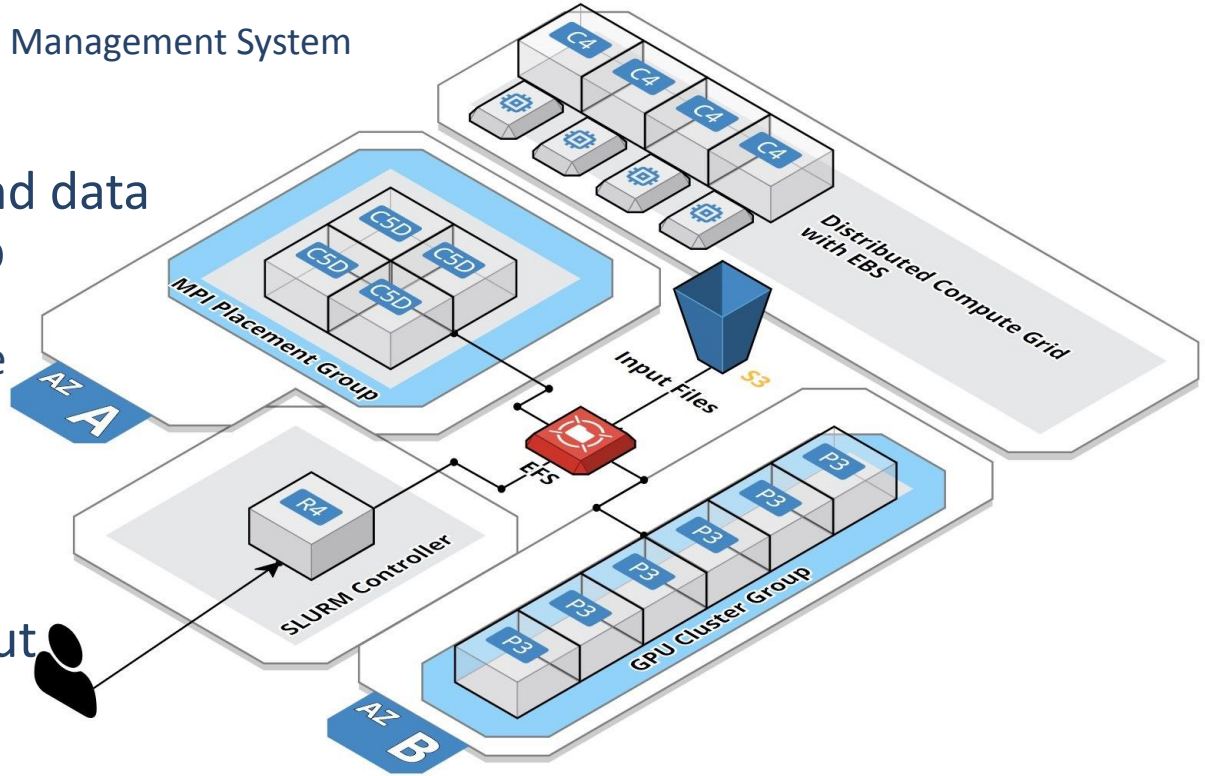
External collaborators

- University of Luxembourg
- TUM, Munich, Germany
- Toulouse Super-computing center CALMIP
- Grenoble HPC center : CIMENT

Classical RJMS

Resources and Jobs Management System

1. User sends jobs and data
2. RJMS schedule job
   a. **Only decisions:** when and where the application starts !!!
3. Job starts
4. Job finishes
5. User obtains output

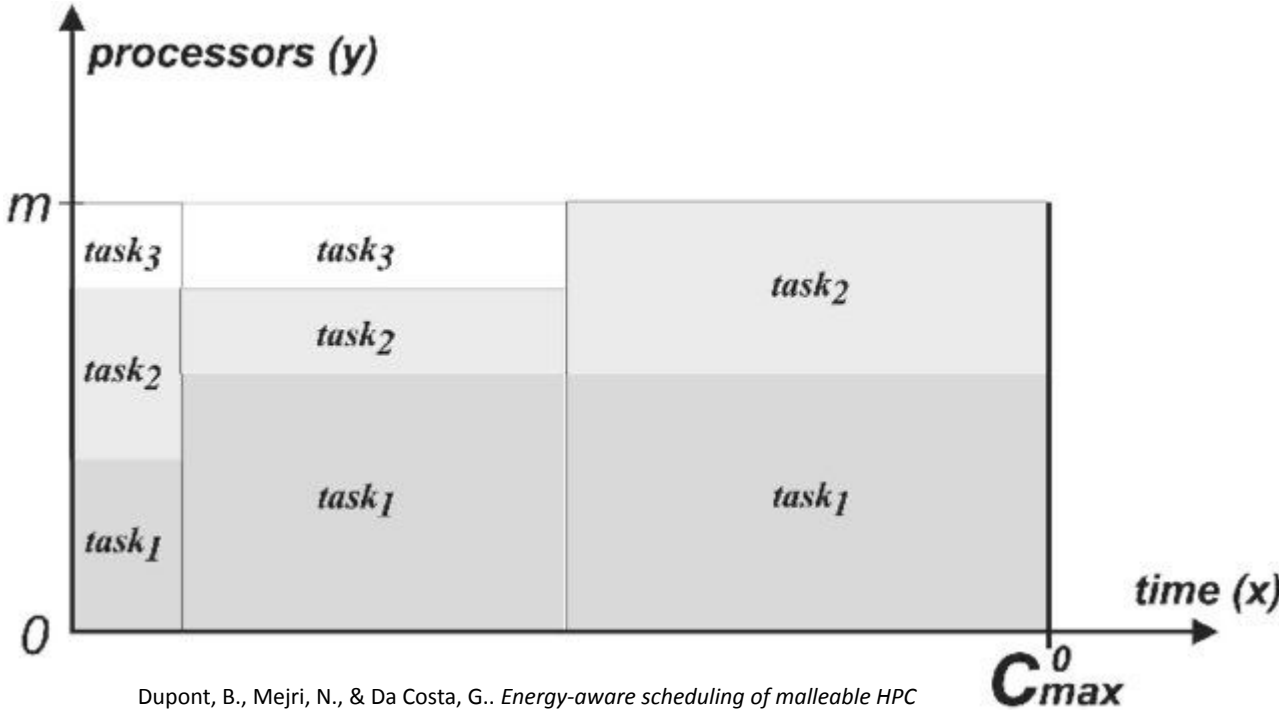


Malleable tasks

Existing simulator including algorithms and cost/performance/energy models

New capability:

- Change resource allocation
  - At starting time
  - During execution
- DVFS
  - Change frequency and voltage



Dupont, B., Mejri, N., & Da Costa, G.. *Energy-aware scheduling of malleable HPC applications using a Particle Swarm optimised greedy algorithm*. **Sustainable Computing: Informatics and Systems**, 2020.

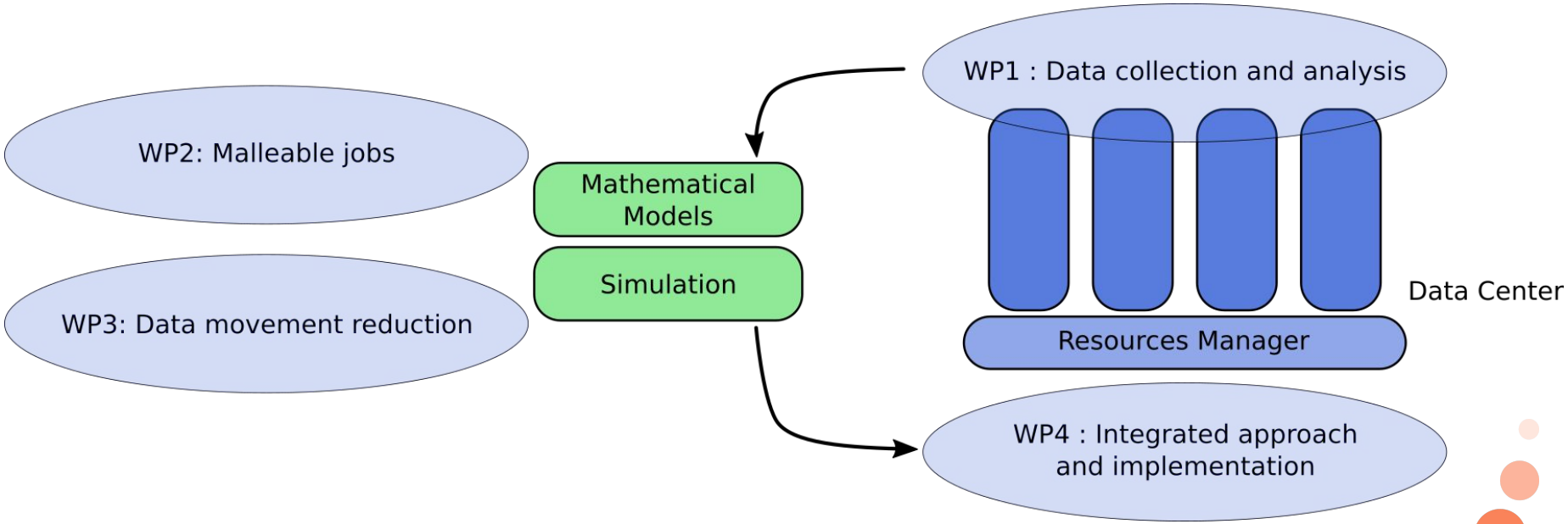
Energumen

Disruptive RJMS for large scale systems

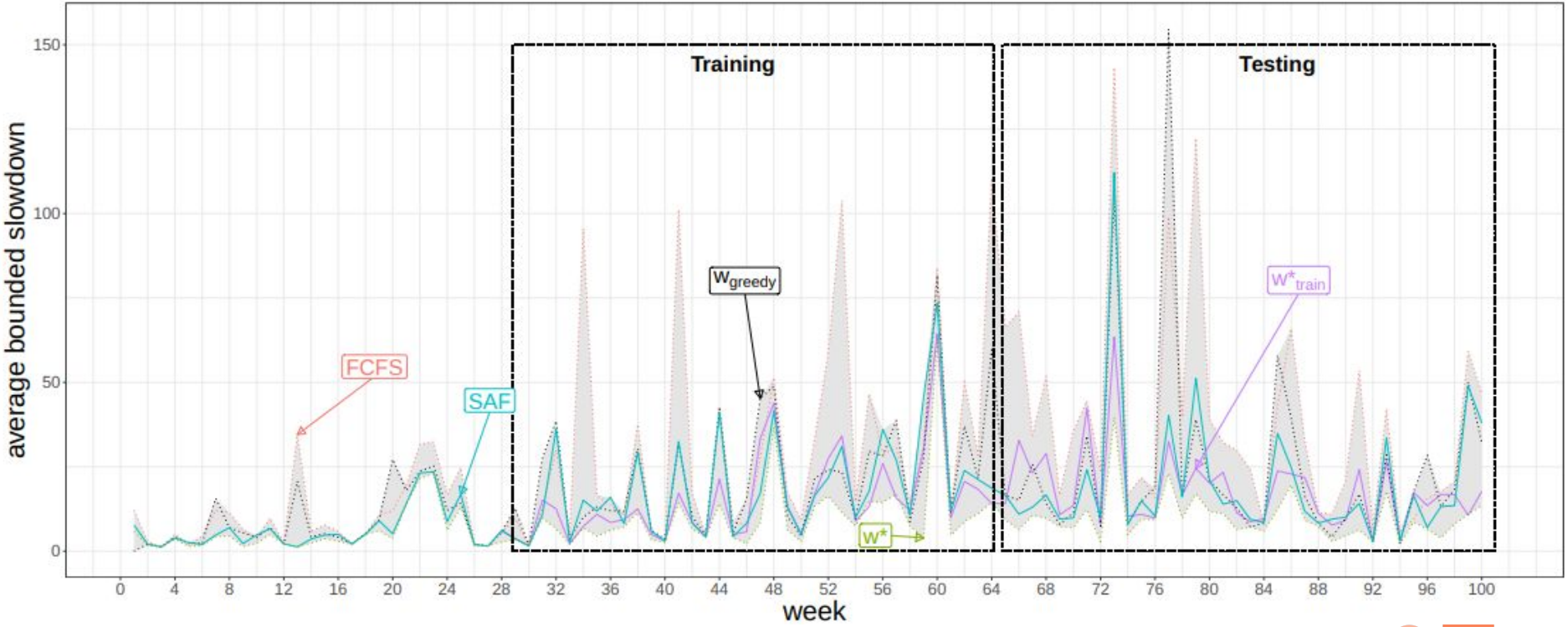
Challenges

1. Collect the most relevant data for energy.
2. Dynamic redimensioning of parallel jobs.
3. Reduce data movements to save energy.
4. Transferability.

Structure of Energumen



Current work Monitoring, data acquisition, learning for scheduling



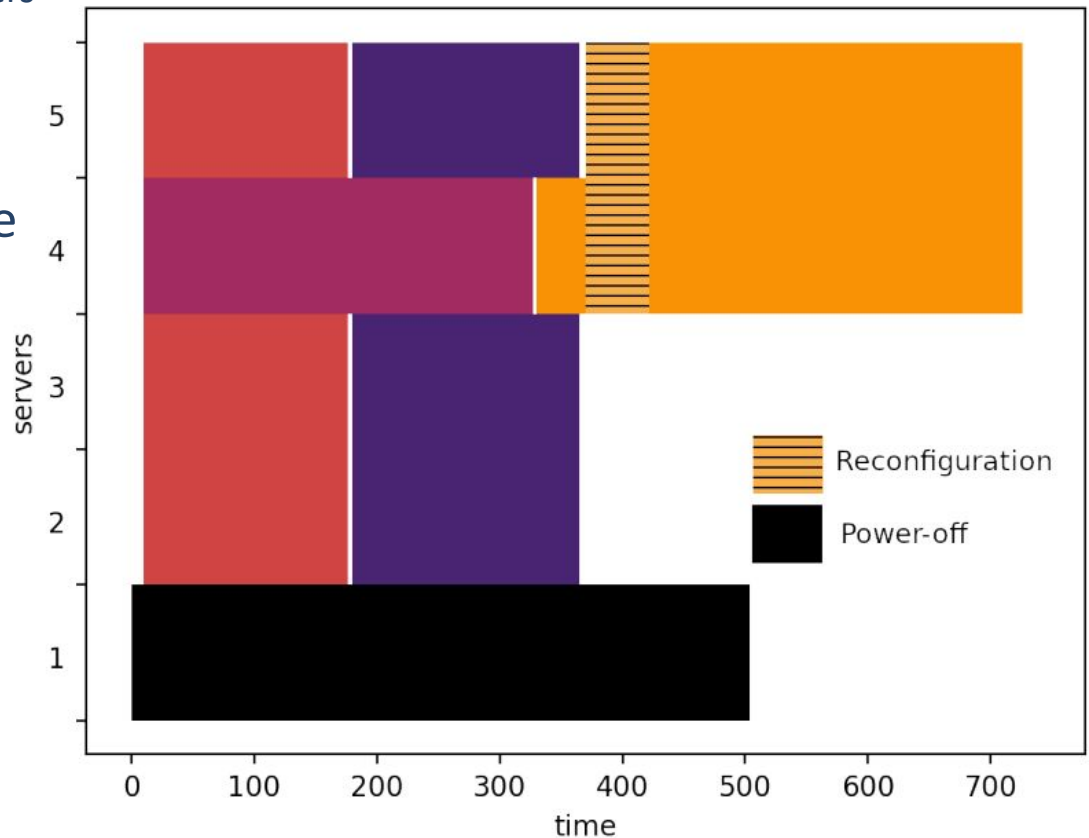
Current work on leverages and scheduling

Flexibility of tasks and servers

Real Time reconfiguration

- Tasks: increase or reduce resources
- Computers: Switch on or off

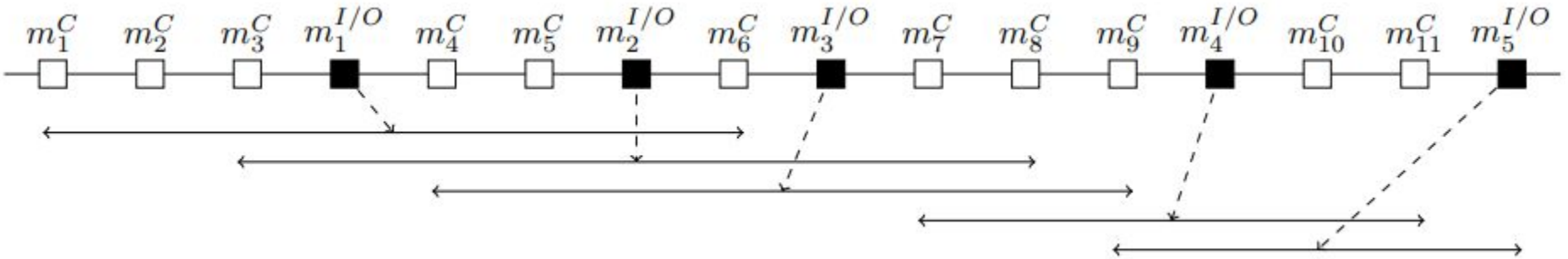
Energy-aware scheduling of malleable HPC applications using a Particle Swarm optimised greedy algorithm. Mejri et al.



Current work on topology

Multiple decisions

- Where to place the applications
- On how many servers



Scheduling Malleable Jobs Under Topological Constraints, Evripidis et al.

Current work on uncertainty

It is costly to be sure

Evaluation to know:

- Impact of DVFS

But costly

- Minimise the use of evaluation

Speed Scaling with Explorable Uncertainty, Evripidis et al.

		Energy	
		Lower Bound	Upper Bound
Offline	Oracle	ϕ^α	-
	CRCD	$\max\{\phi^\alpha, 2^{\alpha-1}\}$	$\min\{2^{\alpha-1}\phi^\alpha, 2^\alpha\}$
	CRP2D		$(4\phi)^\alpha$
	CRAD		$(8\phi)^\alpha$
Online	AVRQ	$(2\alpha)^\alpha$	$2^\alpha 2^{\alpha-1}\alpha^\alpha$
	BKPQ	$3^{\alpha-1}$	$(2+\phi)^\alpha 2(\frac{\alpha}{\alpha-1})^\alpha e^\alpha$
	AVRQ(m)	$(2\alpha)^\alpha$	$2^\alpha(2^{\alpha-1}\alpha^\alpha+1)$

Objective

Focus on Scheduling complex tasks

1. Improvement in the simulator/models
   a. Adding changing energy envelop
   b. Tasks with precise (and predicted) energy needs

2. Proposition of algorithms
   a. Multi-objective scheduling
      i. Performance and Energy
   b. Two leverages will be used
      i. DVFS and malleability

3. Linear programming model of the problem

```
int GetCount(int n, int v[])
{
    int total_count = 0;
    int count = 0;
    int id;          // process id, i.e. "r
    int p;           // number of concurren
    int low, high;   // this rank's partiti

    MPI_Comm_rank(MPI_COMM_WORLD, &id);
    MPI_Comm_size(MPI_COMM_WORLD, &p);

    low = id*(n-1)/p;
    high = (id+1)*(n-1)/p;

    for (int i = low; i < high; i++)
    {
        count += v[i];
    }

    if (p > 1)
        MPI_Reduce(&count, &total_count, 1
    else
        total_count = count;

    return total_count;
}
```