

# Developper Meetup - CLEPS allocations optimization

Simon Legrand

16/04/24

## Contents

<b>1</b>	<b>Allocations optimization on CLEPS</b>	<b>1</b>
1.1	Why should you care? . . . . .	1
1.2	The tricky parts . . . . .	2
1.2.1	<b>Node architecture</b> . . . . .	2
1.2.2	<b>CPU allocation(1)</b> . . . . .	2
1.2.3	<b>CPU allocation(2)</b> . . . . .	2
1.3	The workflow . . . . .	3

## 1 Allocations optimization on CLEPS

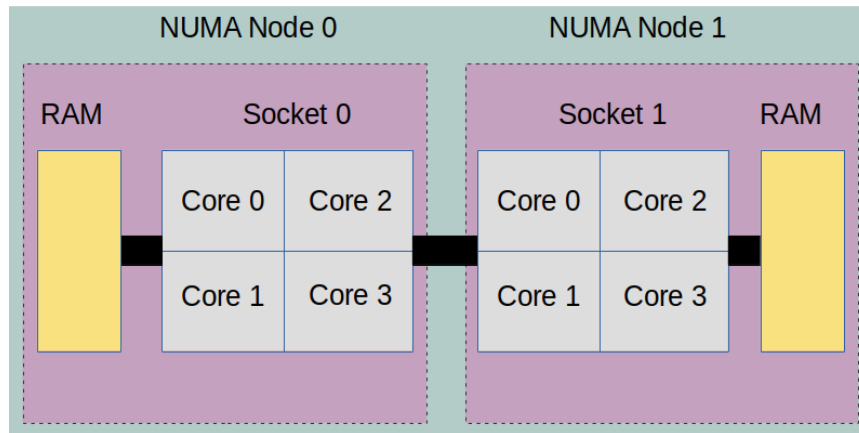
### 1.1 Why should you care?

- **Improve the performances**  
Well used resources lead to better performances
- **Maximize the occupation** of CLEPS  
Hardware ROI depends on its usage
- **Avoid priority penalties**  
Users priority decrease when decreasing 'Used time'/'Wall time'
- (To ease the scheduling)

## 1.2 The tricky parts

### 1.2.1 Node architecture

NUMA: Non Uniform Memory Access



**Extra cost** to access RAM of another NUMA node.

### 1.2.2 CPU allocation(1)

- Your code is parallel:

```
--cpus-per-task    # Shared memory jobs  
--ntasks-per-node  # Distributed memory jobs
```

**BUT** why should you allocate more resources?

- Time constraints (i.e. job longer than walltime max)
- Benchmarking

Otherwise, **best parallel efficiency** achieved with **several instances** of your job with **small allocations**.

### 1.2.3 CPU allocation(2)

Control where resources are allocated

```
--cores-per-socket # Shared memory  
--ntasks-per-socket # Distributed memory
```

- Your code is **memory bound**

Increase the memory bandwidth by using less core per socket

```
--hint=memory_bound # use only one core in each socket
```

- Your code is **compute bound**

Maximize the core occupation

```
--hint=compute_bound # use all cores on each socket
```

More info on Slurm documentation

### 1.3 The workflow

1. **Prepare cpus allocation** interactively

```
salloc ...  
numactl -H  
srun grep Cpus_allowed_list /proc/self/status
```

2. Start with a **large allocation**

3. **Check** your resources consumption

Example: Get the memory peak and elapsed time.

```
sacct -j <job_id> -o MaxRSS,Elapsed
```

4. Adapt future allocations

```
...  
#SBATCH --mem=...  
#SBATCH --time=...  
...
```