Uni.lu HPC School 2018

PS2: HPC workflow with sequential jobs

Uni.lu High Performance Computing (HPC) Team

H. Cartiaux

University of Luxembourg (UL), Luxembourg

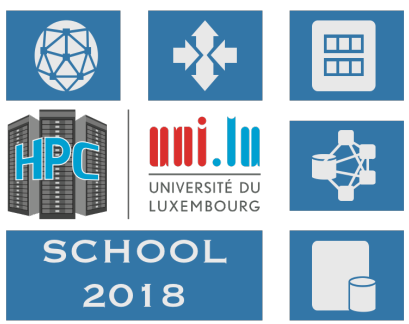
http://hpc.uni.lu

Latest versions available on Github:

UL HPC tutorials: https://github.com/ULHPC/tutorials

UL HPC School: http://hpc.uni.lu/hpc-school/

PS2 tutorial sources: ulhpc-tutorials.rtfd.io/en/latest/basic/sequential_jobs



Summary

Main Objectives of this Session

- Run **sequential**, **parametric programs** on the clusters
- Learn how-to use our set of launcher scripts
- Submit jobs
- use the cluster monitoring tools
  - ↪ Ganglia
  - ↪ Monika & Drawgantt

Tutorial Notes:

https://github.com/ULHPC/tutorials/tree/devel/basic/getting_started

http://git.io/5cYmPw

Summary

Getting started

```
# Connect to the cluster(s)
(laptop)$> ssh {iris,gaia,chaos}-cluster

#  Send files
(laptop)$> rsync -avz local_directory {iris,gaia,chaos}-cluster:

# Retrieve files
(laptop)$> rsync -avz {iris,gaia,chaos}-cluster:path/to/files local_dir
```

- **Submit jobs**

OAR on Chaos/Gaia	**Slurm** on Iris
`oarsub -I`	`srun -p interactive [--qos qos-interactive] --pty bash`
`oarsub ./program`	`sbatch program`

Summary

Gromacs

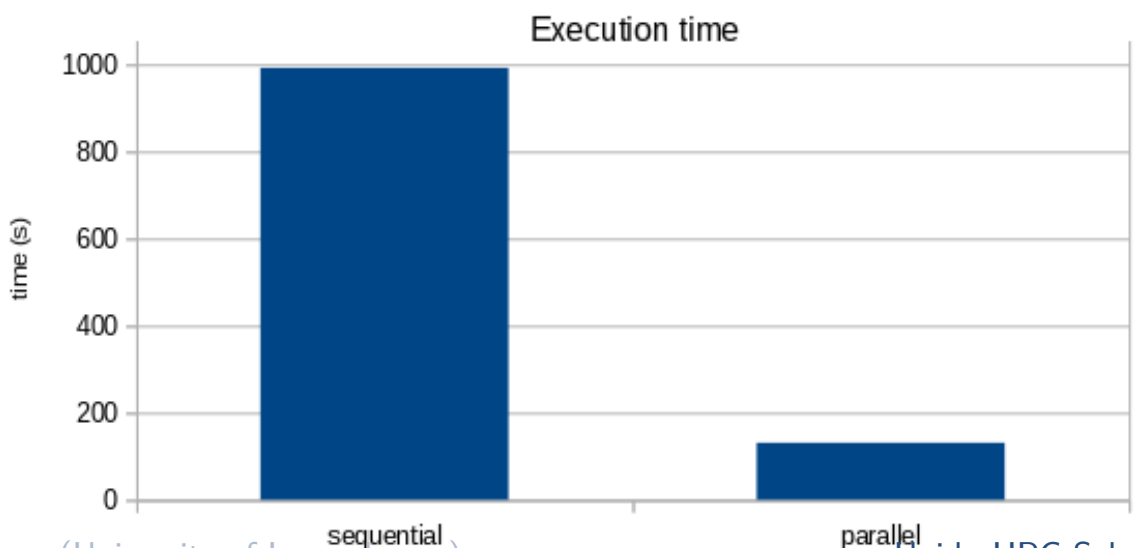
GROMACS: GROningen MAchine for Chemical Simulations
versatile package for molecular dynamics, primarily designed for biochemical molecules

- very large codebase: 1.836.917 SLOC
- many applications in the package, several parallelization modes
- **mdrun**: computational chemistry engine, performing:
  - ↪ molecular dynamics simulations
  - ↪ Brownian Dynamics, Langevin Dynamics
  - ↪ Conjugate Gradient
  - ↪ L-BFGS
  - ↪ Steepest Descents energy minimization
  - ↪ Normal Mode Analysis
- **mdrun** - parallelized using MPI, OpenMP, pthreads and with support for GPU acceleration

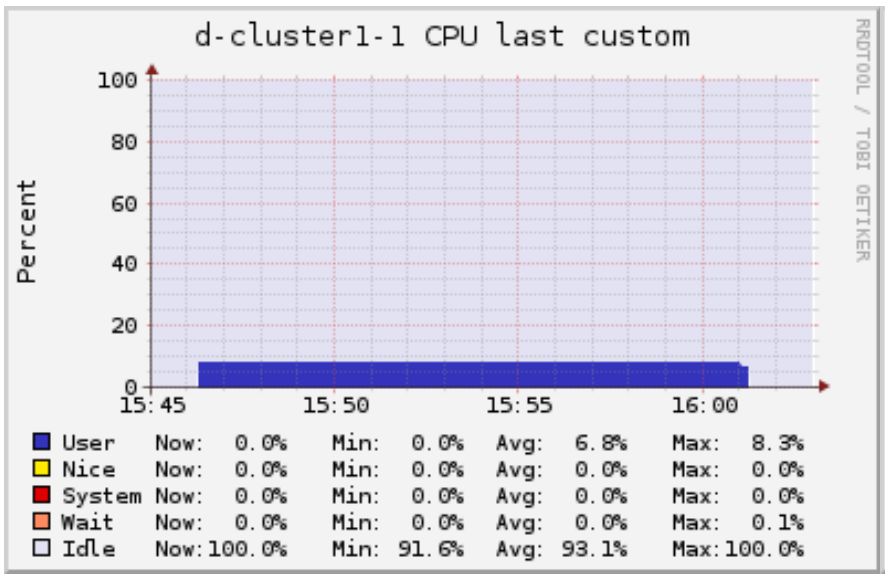
Comparison

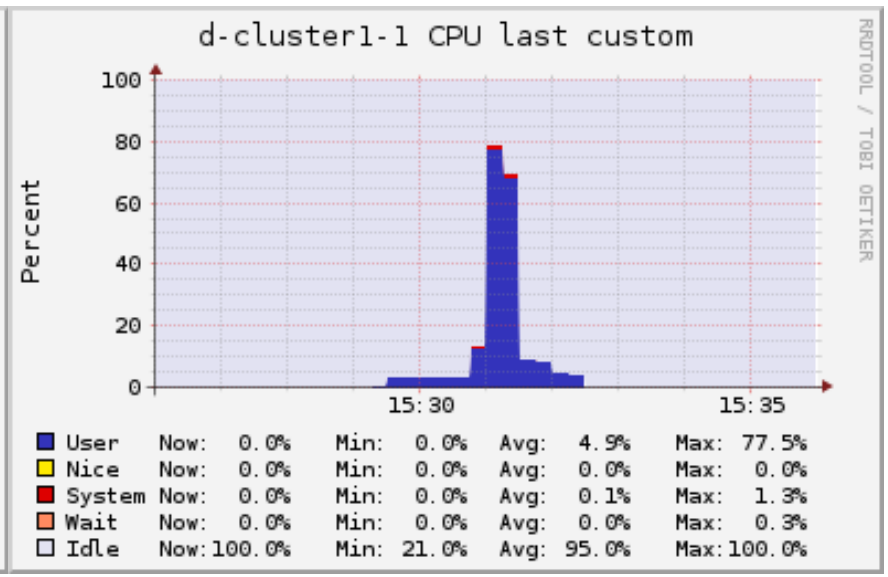
2 approaches

- Sequential (loop)
- Parallized (with GNU parallel)



Comparison - Ganglia

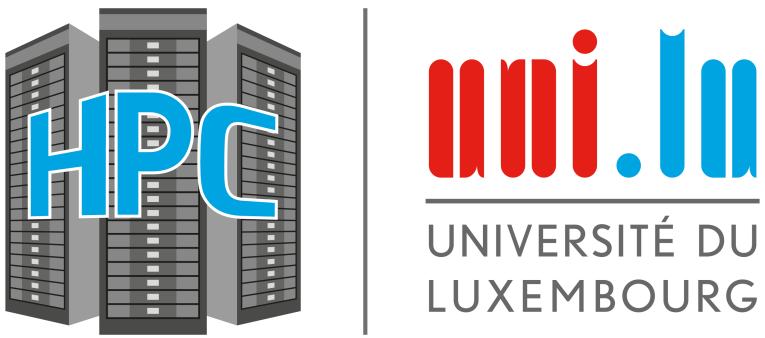




Summary

Watermark Application

- **Objective**: Apply a watermark to a given set of pictures
  - ↪ Simple Python script
  - ↪ Generic parallel launcher
  - ↪ Distribute the work on several nodes



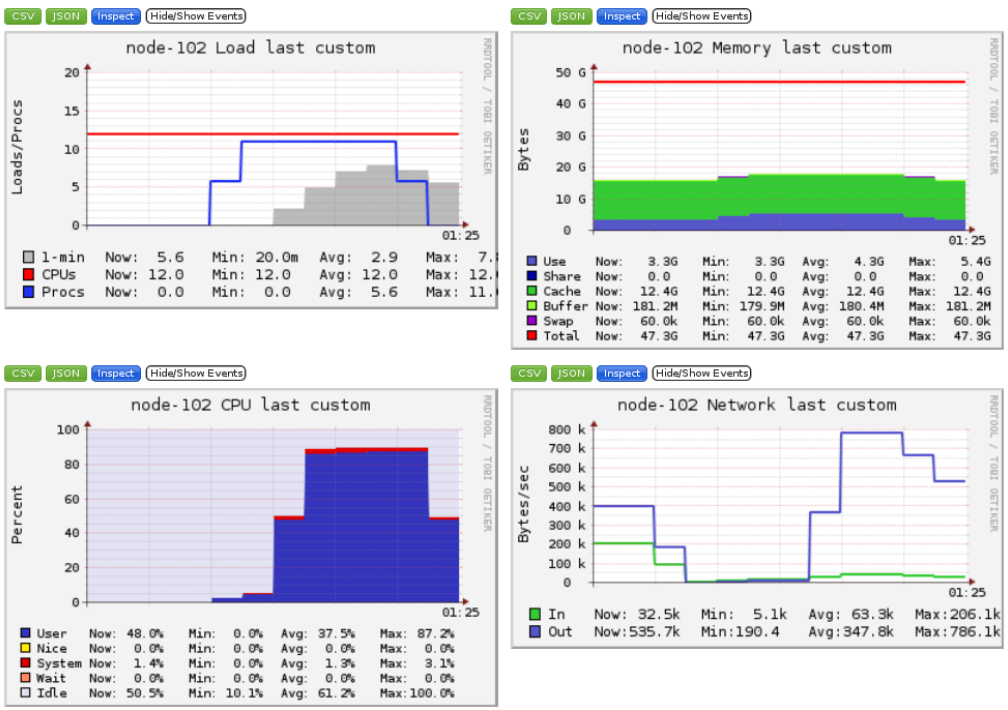
Source image

Watermarked image

Summary

Jcell & cGAs

- **JCell**: a Java framework for working with genetic algorithms
  - ↪ Ex: Generational algorithm for the Combinatorial ECC problem
- Test the variations of these parameters:
  - ↪ *Mutation probability* and *Crossover probability*



Summary

Conclusion

- We have covered one of the most common workflow:
  - ↪ **parametric jobs**
- Our launchers can be improved!

Perspectives

- Array jobs
- Best effort jobs
- Checkpoint/Restart mechanism

Questions?

http://hpc.uni.lu

High Performance Computing @ uni.lu

Prof. Pascal Bouvry
Dr. Sebastien Varrette
Valentin Plugaru
Sarah Peter
Hyacinthe Cartiaux
Clement Parisot

University of Luxembourg, Belval Campus
Maison du Nombre, 4th floor
2, avenue de l'Université
L-4365 Esch-sur-Alzette
mail: hpc@uni.lu