

Uni.lu HPC School 2018

PS4a: UL HPC Monitoring in practice: why, what, how, where to look



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/>

PS4a tutorial sources:





Summary

- 1 Objectives**
- 2 Experiment planning & workflow
- 3 Hardware knowledge
- 4 Live status
- 5 Common mistakes and pitfalls
- 6 Getting help



Objectives

- Understanding a standard HPC workflow (development and production run)
- Monitoring your job at the system level
- Monitoring your job at the job scheduler level
- Finding your bottlenecks



Summary

- 1 Objectives
- 2 Experiment planning & workflow**
- 3 Hardware knowledge
- 4 Live status
- 5 Common mistakes and pitfalls
- 6 Getting help



Workflow for Experiment Campaigns

1. Before the campaign
 - Send data to the clusters
 - Check/Install required software
2. Preparation
 - Test and debug
 - Prepare launcher script
3. Execution
 - Run the campaign
 - Monitor the execution
4. After a campaign
 - Retrieve output data
 - Archive and cleanup your data



Experiment campaign: Preparation

Goals

- Make sure everything will run OK
- Prepare submission script / launcher

Interactive approach

- Use option `-I` (Interactive) of `oarsub` command
- Allows to try commands one by one
- Work on a small case with a small number of cores
- Debug and check the results

Why prepare a submission script?

- Contains all commands and parameters
⇒ Easy re-execution
- No need to stay in front your computer

Experiment campaign: Execution

Submit the jobs

- Use the submission script / launcher
- Gaia/Chaos: submit to OAR with option `-S` (Script) of `oarsub`
- Iris: Submit to Slurm with `sbatch your_script.sh` (Slurm batch Script)
- Actual experiment execution with possibly many nodes
- Non interactive execution, it might not start immediately

Get resource usage of your program with time

- `time` tool is installed on the nodes which give execution time of a command or script
- You can use `time` command to get more info about CPU usage, memory usage, socket messages sent, etc...
- `/usr/bin/time -v ./stress-script.sh`



Experiment campaign: Monitor the execution

- Output/Logfile of your application
- Slurm scheduler information:
 - ↪ List your jobs and statuses: `squeue -u <yourlogin>`
 - ↪ Information of a running job: `sstat -j <jobid>`
 - ↪ Used resources information: `sacct`
- OAR
 - ↪ List your jobs and statuses: `oarstat -u <yourlogin>`
 - ↪ Get job information: `oarstat -f -j <jobid>`



Summary

- 1 Objectives
- 2 Experiment planning & workflow
- 3 Hardware knowledge**
- 4 Live status
- 5 Common mistakes and pitfalls
- 6 Getting help



Heterogeneity

- Heterogeneous clusters
- Special nodes ('bigmem', 'big SMP')
- Nodes must be targeted explicitly if reproducibility is important (benchmarks, performance evaluation, algorithm comparison, etc)

Heterogeneity (Iris)

Vendor	N	Proc. Description	Cores	Mem
Dell	108	Xeon E5-2680 v4 @ 2.4GHz	2 × 14C	128GB
Dell	60	Xeon Gold 6132 @ 2.6GHz	2 × 14C	128GB

- sbatch -C broadwell [...]
- sbatch -C skylake [...]

Heterogeneity (Gaia)

	N	Proc. Description	Cores	Mem
Bull	60	Xeon L5640 @ 2.26GHz	2 × 6C	48GB
Bull	12	Xeon L5640 @ 2.26GHz + GPUs	2 × 6C	96GB
Bull	1	Xeon E7-4850 @ 2GHz	16 × 10C	1TB
Dell	1	Xeon E5-4640 @ 2.4GHz	4 × 8C	1TB
Dell	5	Xeon E5-2660 @ 2.2GHz + GPUs	2 × 8C	64GB
SGI	1	Xeon E7-8880v2 @ 2.5GHz	8 × 15C	3TB
Bull	40	Xeon X5670 @ 2.93GHz	2 × 6C	48GB
Bull	32	Xeon X5675 @ 3.07GHz	2 × 6C	48GB
Dell	24	Xeon E5-2680v3 @ 2.5GHz	2 × 12C	128GB
Dell	4	Xeon E5-2680v3 @ 2.5GHz + GPUs	2 × 12C	128GB
Dell	2	Xeon E7-8867v3 @ 2.5GHz	2 × 16C	2TB
HP	90	Xeon E3-1284Lv3 @ 1.8GHz	1 × 4C	32GB



Heterogeneity (Gaia)

Use OAR properties:

- `oarsub -l -p "nodemodel='Dell_FC430'"`
- `oarsub -l -p "nodemodel='Bull_B500'"`
- etc

See <https://hpc.uni.lu/users/docs/oar.html>



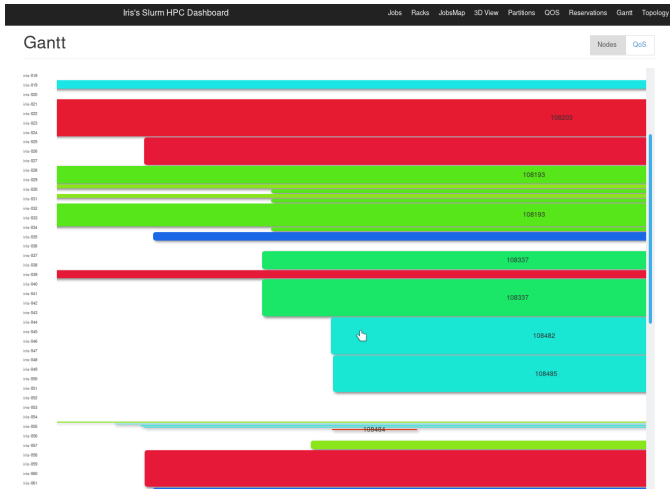
Summary

- 1 Objectives
- 2 Experiment planning & workflow
- 3 Hardware knowledge
- 4 Live status**
- 5 Common mistakes and pitfalls
- 6 Getting help



Slurm Web (Iris)

<https://access-iris.uni.lu/slurm/>





Slurm Web (Iris)

<https://access-iris.uni.lu/slurm/>Iris's Slurm HPC Dashboard Jobs Racks JobsMap 3D View Partitions QOS Reservations Gantt Topolog

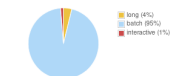
Jobs



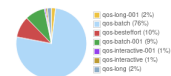
Allocated cores



Jobs/State



Cores/Partition

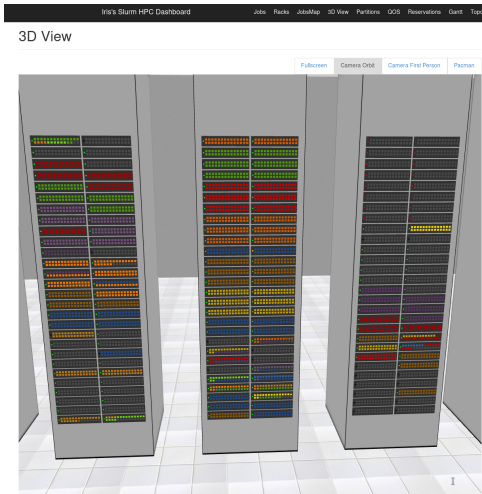


Cores/QOS

#	User	Resources	State	Reason	Start	QOS	Partition
103785	ipoltavskiy (Igor)	cpu=28,mem=112G,node=1,billing=28	RUNNING	-	since 14d 44min 12s	qos-long-001	long
106752	ipoltavskiy (Igor)	cpu=28,mem=112G,node=1,billing=28	RUNNING	-	since 5d 2h 6min	qos-long-001	long
107978	gpozetti (Gabriele)	cpu=14,mem=56G,node=1,billing=14	RUNNING	-	since 2d 23h 40min 59s	qos-batch	batch
107979	gpozetti (Gabriele)	cpu=14,mem=56G,node=1,billing=14	RUNNING	-	since 2d 23h 40min 48s	qos-batch	batch
107980	gpozetti (Gabriele)	cpu=14,mem=56G,node=1,billing=14	RUNNING	-	since 2d 23h 40min 45s	qos-batch	batch
107991	yliao (YuChung)	cpu=112,mem=448G,node=4,billing=112	RUNNING	-	since 2d 20h 32min 44s	qos-batch	batch
108107	yliao (YuChung)	cpu=28,mem=112G,node=1,billing=28	RUNNING	-	since 1d 13min 50s	qos-batch	batch
108108	yliao (YuChung)	cpu=28,mem=112G,node=1,billing=28	RUNNING	-	since 1d 13min 45s	qos-batch	batch
108193	amolinasanchez (Alejandro)	cpu=128,mem=512G,node=5,billing=128	RUNNING	-	since 7h 43min 11s	qos-besteffort	batch
108198	mbarborini (Matteo)	cpu=28,mem=112G,node=1,billing=28	RUNNING	-	since 7h 27min 21s	qos-besteffort	batch
108199	mbarborini (Matteo)	cpu=28,mem=112G,node=1,billing=28	RUNNING	-	since 7h 26min 34s	qos-batch	batch
108203	mbarborini (Matteo)	cpu=112,mem=448G,node=4,billing=112	RUNNING	-	since 7h 4min 2s	qos-batch	batch
108222	ipoltavskiy (Igor)	cpu=4,mem=16G,node=1,billing=4	RUNNING	-	since 5h 33min 8s	qos-batch-001	batch

Slurm Web (Iris)

<https://access-iris.uni.lu/slurm/>





Current node status - Monika (Chaos & Gaia)

<http://hpc.uni.lu/{gaia,chaos}/{monika}>

Chaos nodes

default summary			
	Free	Busy	Total
network_address	24	41	81
resource_id	320	604	1120

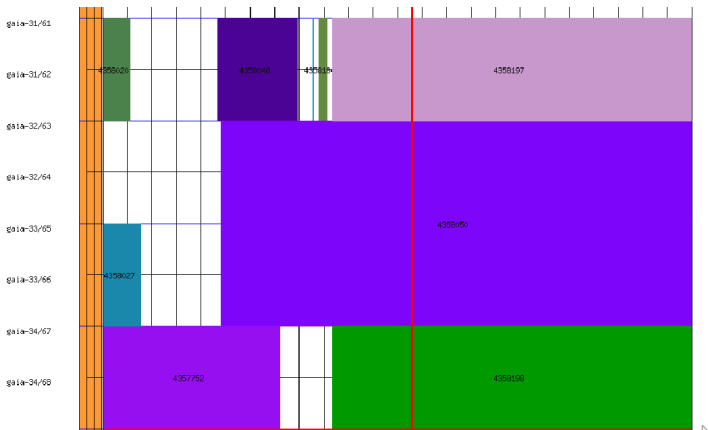
Reservations:

d-cluster1-1	1640999	1640999	1640999	1640999	1640999	1640999	1640968	1640971	1640971
d-cluster1-2	Free	Free	Free	Free	Free	Free	Free	Free	Free
d-cluster1-3	1640985	1640985	1640985	1640985	1640985	1640985	1640992	1640993	1640993
d-cluster1-4	Absent	Absent	Absent	Absent	Absent	Absent	Absent	Absent	Absent
d-cluster1-5	1640985	1640985	1640985	1640985	1640985	1640985	Free	Free	Free
d-cluster1-6	Free	Free	Free	Free	Free	Free	Free	Free	Free
d-cluster1-7	Free	Free	Free	Free	Free	Free	Free	Free	Free
d-cluster1-8	Free	Free	Free	Free	Free	Free	Free	Free	Free
d-cluster1-9	Free	Free	Free	Free	Free	Free	Free	Free	Free
d-cluster1-10	1640969	1640969	1640969	1640969	1640969	1640969	1640976	1640977	1640977
d-cluster1-11	1640969	1640969	1640969	1640969	1640969	1640969	1640982	1640983	1640983
d-cluster1-12	1640969	1640969	1640969	1640969	1640969	1640969	1640985	1640985	1640985
d-cluster1-13	1640999	1640999	1640999	1640999	1640999	1640999	1640999	1640969	1640969
d-cluster1-14	1640999	1640999	1640999	1640999	1640999	1640999	1640969	1640969	1640969
d-cluster1-15	1640999	1640999	1640999	1640999	1640999	1640999	1640985	1640985	1640985
d-cluster1-16	1640999	1640999	1640999	1640999	1640999	1640999	1640999	1640985	1640985
e-cluster1-1	1640911	1640911	1640911	1640911	1640911	1640911	1640911	1640911	1640911
e-cluster1-2	1640749	1640749	1640749	1640749	1640749	1640749	1640749	1640749	1640749
e-cluster1-3	1640749	1640749	1640749	1640749	1640749	1640749	1640749	1640749	1640749
e-cluster1-4	1640749	1640749	1640749	1640749	1640749	1640749	1640749	1640749	1640749
e-cluster1-5	1640949	1640949	1640949	1640949	1640949	1640949	1640949	1640949	1640949



Platform occupation - Drawgantt (Chaos & Gaia)

<http://hpc.uni.lu/{gaia,chaos}/{drawgantt}>
<http://hpc.uni.lu/{gaia,chaos}/{drawgantt-svg}>





Resource usage - Ganglia

http://hpc.uni.lu/{iris,gaia,chaos}/ganglia

Main Search Aggregate Graphs Compare Hosts Reports Live Dashboard Cubism Mobile

ULHPC Infrastructure Grid Report at Mon, 11 Jun 2018 15:47:07 +0000 Get Fresh Data

Last 2hr 4hr day week month year job or from

Sorted descending by name by hosts up by hosts down

ULHPC Infrastructure Grid > --Choose a Source v

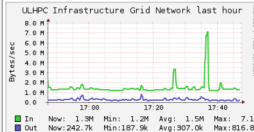
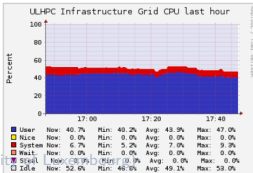
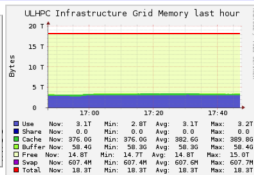
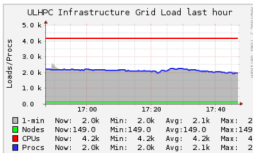
ULHPC Infrastructure Grid (1 sources) (tree view)

CPUs Total: **4172**
 Hosts up: **149**
 Hosts down: **1**

Current Load Avg (15, 5, 1m):
 50%, 49%, 48%

Avg Utilization (last hour):
 51%

Localtime:
 2018-06-11 15:47



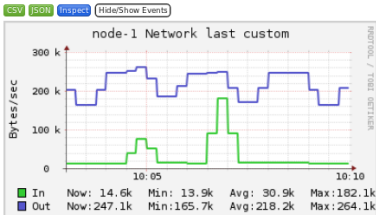
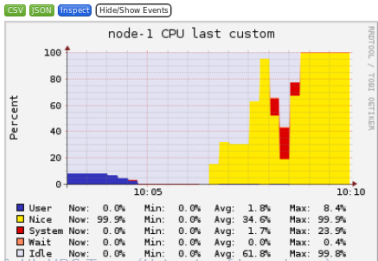
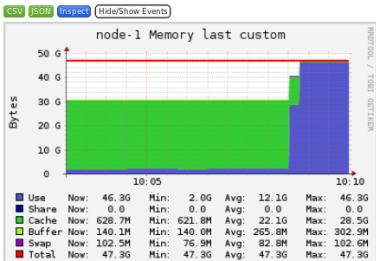
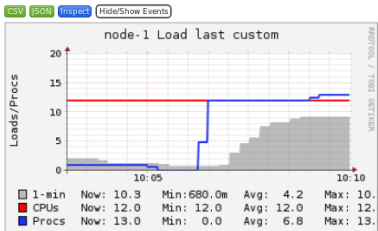


Getting faster: Identify performance bottlenecks

Note for code developers: **The first bottleneck is your algorithm!**
Know the hardware

- Computer nodes are connected using a fast interconnect
- Different types of resources:
Processors, GPU, Memory, Storage, Network

Identify your bottleneck (memory)



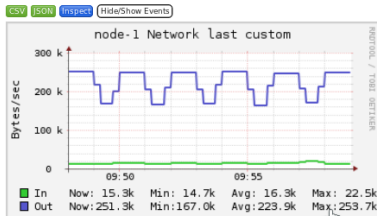
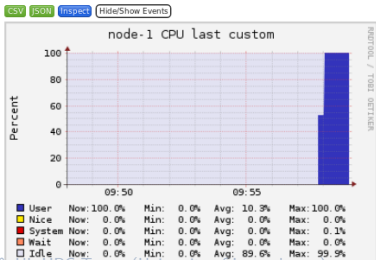
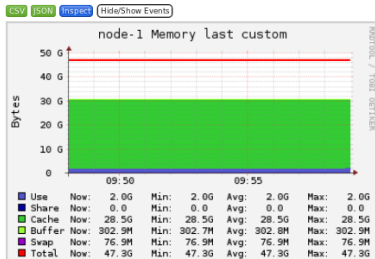
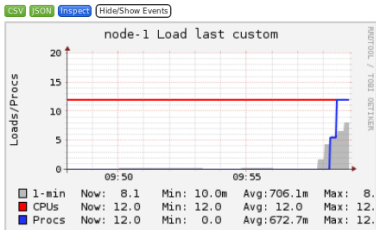


Identify your bottleneck (memory)

Application is limited by the size of the memory

- Reserve all CPUs on a single node, to get access to all memory banks
- Use a node with a bigger memory (bigmem)
- Distributed execution on multiple nodes (MPI)

Identify your bottleneck (CPU)





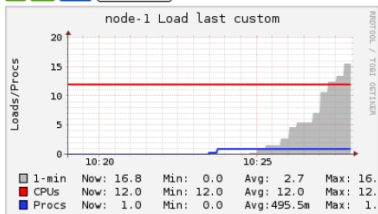
Identify your bottleneck (CPU)

Application is limited by the speed of the processor

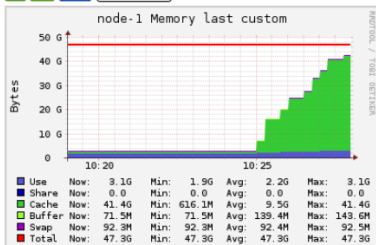
- Optimize your code
- Use GPU accelerator (CUDA)
- Parallel execution on multiple nodes (MPI)
- Parallel execution on multiple nodes with GPUs (MPI+CUDA)

Identify your bottleneck (I/O)

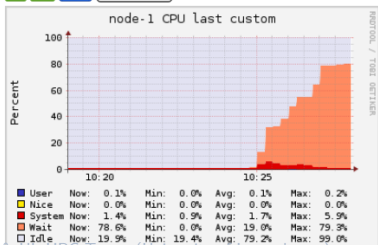
CSV JSON Inspect Hide/Show Events



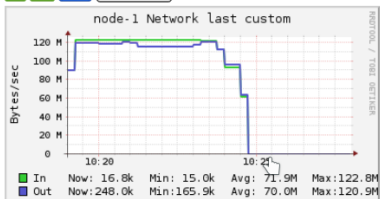
CSV JSON Inspect Hide/Show Events



CSV JSON Inspect Hide/Show Events



CSV JSON Inspect Hide/Show Events





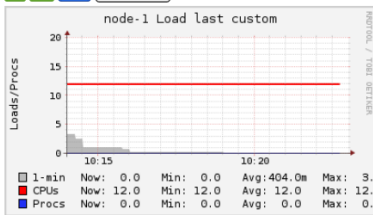
Identify your bottleneck (I/O)

Application is limited by the speed of the storage

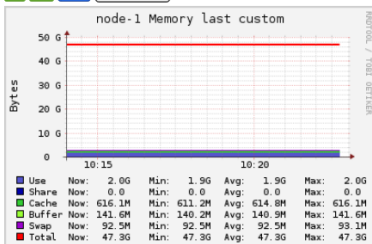
- Use local storage, eg /tmp, instead of network storage
- Use local memory, eg /dev/shm
- Use \$SCRATCH (no backup)

Identify your bottleneck (Network)

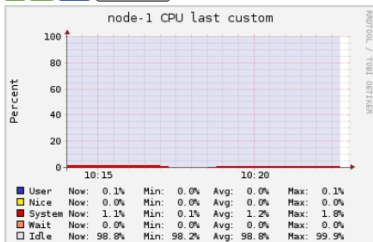
CSV JSON Inspect Hide/Show Events



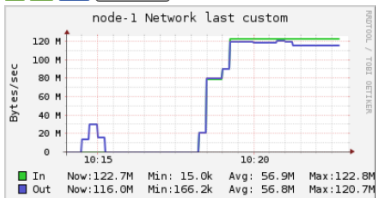
CSV JSON Inspect Hide/Show Events



CSV JSON Inspect Hide/Show Events



CSV JSON Inspect Hide/Show Events





Identify your bottleneck (Network)

Application is limited by the speed of the network (too many communications)

- Use Infiniband instead of Ethernet
- Reduce the number of nodes



Command line tools

Connect to the node during your passive job

- OAR: `oarsub -C <jobid>`
- SLURM: `srun -jobid <jobid> -pty bash`
- memory usage: `free -m`
- list current processes and statistics: `ps aux`
- processes ordered by CPU usage: `top`
- like top, but interactive: `htop`
- filesystem usage: `df-ulhpc`
- live system statistics (including I/O and network): `dstat`



Summary

- 1 Objectives
- 2 Experiment planning & workflow
- 3 Hardware knowledge
- 4 Live status
- 5 Common mistakes and pitfalls**
- 6 Getting help

Common pitfalls

My job has been terminated, why ?

① Maximum memory usage exceeded

- ↪ The Linux *Out Of Memory Killer* (OOMK) mechanism killed your processes silently.
- ↪ The available memory depends on the number of cores/CPU's reserved

② Requested walltime exceeded

- ↪ The walltime specified with your submission command (sbatch, srun, oarsub) was too short
- ↪ Your job walltime cannot be extended after its submission
- ↪ The job duration must be estimated before its submission



Summary

- 1 Objectives
- 2 Experiment planning & workflow
- 3 Hardware knowledge
- 4 Live status
- 5 Common mistakes and pitfalls
- 6 Getting help**



Know the basics!

Access the clusters, access and reserve nodes

- Use SSH and public key authentication
<https://hpc.uni.lu/users/docs/access.html>
- Learn how to use the OAR resource manager
<https://hpc.uni.lu/users/docs/oar.html>
- Learn how to use the SLURM batch scheduler (Iris cluster only)
<https://hpc.uni.lu/users/docs/slurm.html>

Transfer files between your computer and the clusters

- Learn how to use tools like scp, rsync, etc.
<https://hpc.uni.lu/users/docs/filetransfer.html>

Use pre-installed software

- Search and use software with the module command
<https://hpc.uni.lu/users/docs/modules.html>



In this order

- 1 Check the UL HPC quick reference
`https://hpc.uni.lu/download/documents/ulhpc-quickref.pdf`
- 2 Read The Fine Manual at `https://hpc.uni.lu/docs`
- 3 `$ man man`
- 4 Google is your friend!
- 5 Open a ticket on
`hpc-tracker.uni.lu`
- 6 Ask the HPC sysadmins
`hpc-sysadmins@uni.lu`
- 7 Bonus: ask the users community mailing list
`hpc-users@uni.lu`

Questions?

<http://hpc.uni.lu>

High Performance Computing @ uni.lu

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

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



- 1 Objectives
- 2 Experiment planning & workflow
- 3 Hardware knowledge

- 4 Live status
- 5 Common mistakes and pitfalls
- 6 Getting help