# UL HPC Facility Workload Analysis

## Impact of updated model for Fairshare, Account Hierarchy and Limits for Iris/Aion. (draft) Workload characterization

High Performance Computing & Big Data Services

hpc.uni.lu

hpc@uni.lu

@ULHPC


**S. Varrette**, F. Pinel, E. Kieffer, G. Danoy

University of Luxembourg (UL), Luxembourg

https://hpc.uni.lu


Energumen Days June 24th, 2021, Lyon, France

# Summary

# Summary

# High Performance Computing @ UL

- **Started in 2007** under resp. of Prof P. Bouvry & Dr. S. Varrette
  - ↪ 2nd Largest HPC facility in Luxembourg...
    - ✓ after EuroHPC MeluXina ($\simeq$ 18 PFlops) system

`hpc.uni.lu`

Technical Docs:
`hpc-docs.uni.lu`

HPC/Computing Capacity

2794.23 TFlops

(incl. 748.8 GPU TFlops)

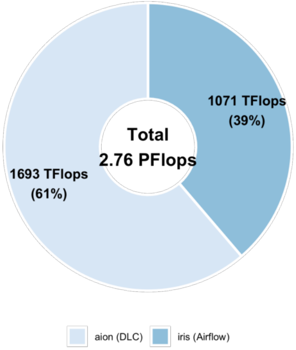
Shared Storage Capacity

10713.4 TB storage

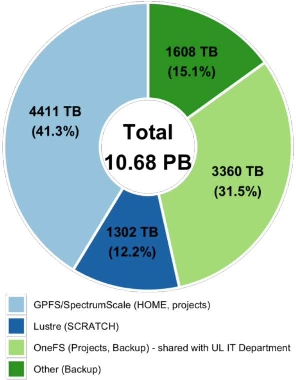
ULHPC Tutorials: `ulhpc-tutorials.rtfd.io`

# High Performance Computing @ UL

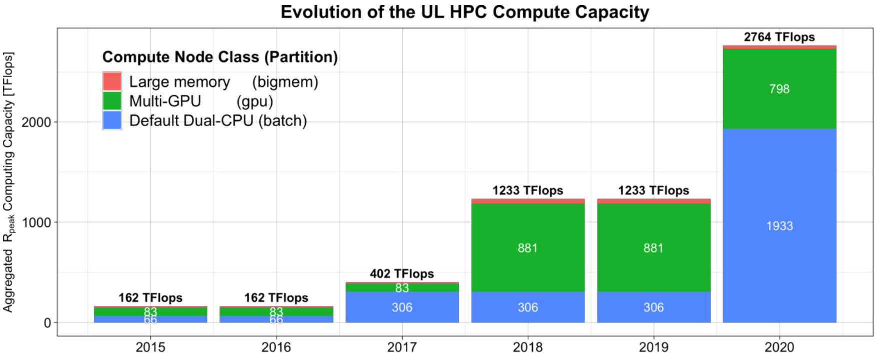
### UL HPC Cluster (2021)




### UL HPC Storage FileSystems (2020)



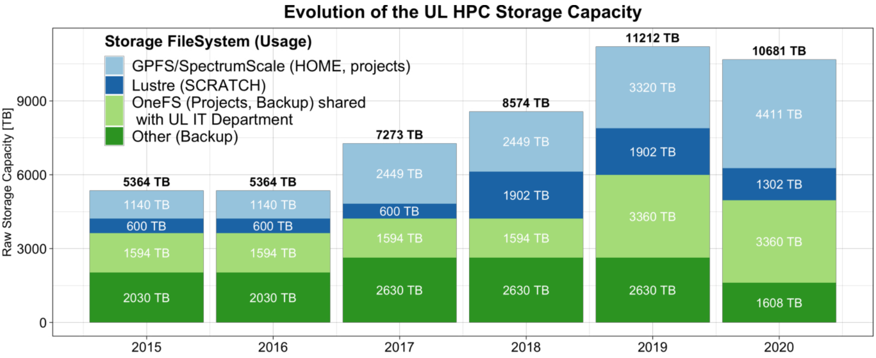

# High Performance Computing @ UL





- 3 types of computing resources across **2 clusters** (`aion`, `iris`)

# High Performance Computing @ UL




- 4 File Systems commons across the **2 clusters** (`aion`, `iris`)

# Uni.lu Data Center

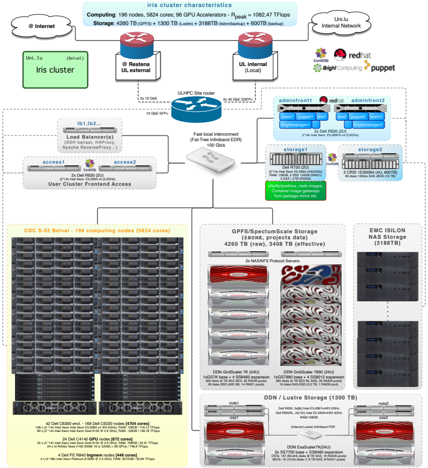

**Belval Campus**

Centre De Calcul (CDC)

- Power generation station for HPC floor:
  - ↪ up to **3 MW of electrical power**
  - ↪ **2.4 MW of cold water** at a 12-18°C regime
    - ✓ used for traditional Airflow with In-Row cooling.
  - ↪ Separate hot water circuit (between 30 and 40°C)
    - ✓ used for Direct Liquid Cooling (DLC): `aion`

| Location | Cooling | Usage |
|---|---|---|
| CDC S-02-001 | Airflow | **Future extension** |
| CDC S-02-002 | Airflow | **Future extension** |
| CDC S-02-003 | DLC | **Future extension** - High Density/Energy efficient HPC |
| **CDC S-02-004** | **DLC** | **High Density/Energy efficient HPC**: `aion` |
| **CDC S-02-005** | **Airflow** | **Storage / Traditional HPC**: `iris` and common equipment |

# UL HPC Supercomputers: iris cluster

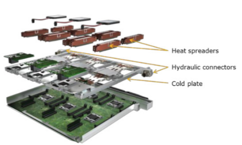



`hpc-docs.uni.lu/systems/iris/`

- **Dell/Intel** supercomputer, Air-flow cooling
  - ↪ 196 compute nodes, **5824 cores**, 52.2 TB RAM
  - ↪ $R_{peak}$: **1,072 PetaFLOP/s**
- Fast InfiniBand (IB) EDR network
  - ↪ **Fat-Tree** Topology     blocking factor 1:1.5

| Rack ID | Purpose | Description |
|---|---|---|
| D02 | Network | Interconnect equipment |
| D04 | Management | Management servers, Interconnect |
| D05 | Compute | iris-[001-056], interconnect |
| D07 | Compute | iris-[057-112], interconnect |
| D09 | Compute | iris-[113-168], interconnect |
| D11 | Compute | iris-[169-177,191-193](gpu), iris-[187-188](bigmem) |
| D12 | Compute | iris-[178-186,194-196](gpu), iris-[189-190](bigmem) |

# UL HPC Supercomputers: aion cluster




hpc-docs.uni.lu/systems/aion/

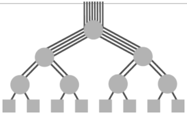
- **Atos/AMD** supercomputer, DLC cooling
  - ↪ 4 BullSequana XH2000 adjacent racks
  - ↪ 318 compute nodes, **40704 cores**, 81.4 TB RAM
  - ↪ $R_{\text{peak}}$: **1,693 PetaFLOP/s**
- Fast InfiniBand (IB) HDR network
  - ↪ **Fat-Tree** Topology    blocking factor 1:2


| | Rack 1 | Rack 2 | Rack 3 | Rack 4 | **TOTAL** |
|---|---|---|---|---|---|
| **Weight [kg]** | 1872,4 | 1830,2 | 1830,2 | 1824,2 | **7357 kg** |
| **#X2410 Rome Blade** | 28 | 26 | 26 | 26 | **106** |
| **#Compute Nodes** | 84 | 78 | 78 | 78 | **318** |
| **#Compute Cores** | 10752 | 9984 | 9984 | 9984 | **40704** |
| $R_{\text{peak}}$ **[TFlops]** | 447,28 TF | 415,33 TF | 415,33 TF | 415,33 TF | **1693.29 TF** |

# Fast Local Interconnect Network

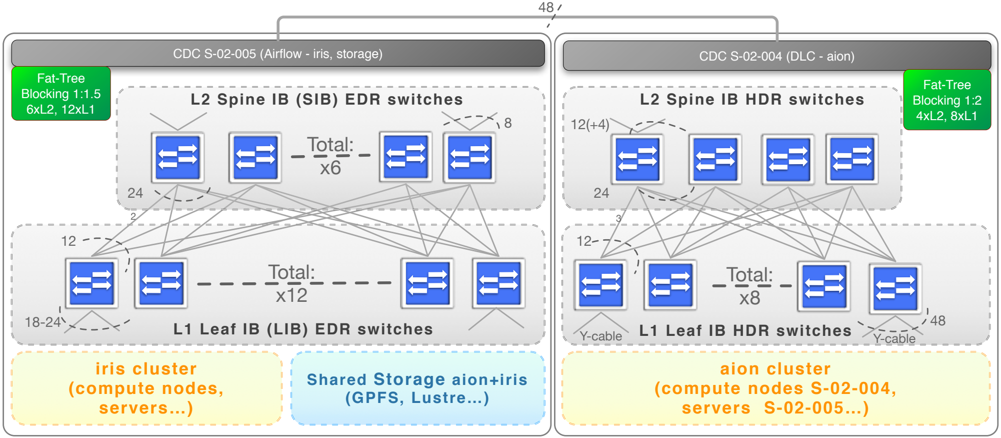
- HPC interconnect technologies nowadays divided into three categories
  1. Ethernet: dominant interconnect standard yet underlying protocol has inherent limitations
     - ✓ preventing low-latency deployments expected in real HPC environment
  2. InfiniBand: predominant interconnect technology in the HPC market
  3. Vendor specific interconnects: Cray/HPC Slingshot, Intel Omni-Path, Bull BXI...

- **On ULHPC**: **InfiniBand (IB)** solution
  - ↪ `iris`: Infiniband (IB) **EDR** Fabric in a **Fat-Tree** Topology
  - ↪ `aion`: Infiniband (IB) **HDR100** Fabric in a **Fat-Tree** Topology



- Up/Down InfiniBand Routing Algorithm
  - ↪ super-set of Fat-Tree with a tracker mode (allow each node to have dedicated route)
  - ↪ well adapted to IO traffic patterns

# Fast Local Infiniband Interconnect Network



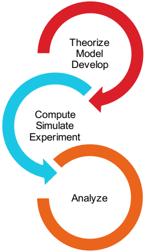
# UL HPC Software Stack

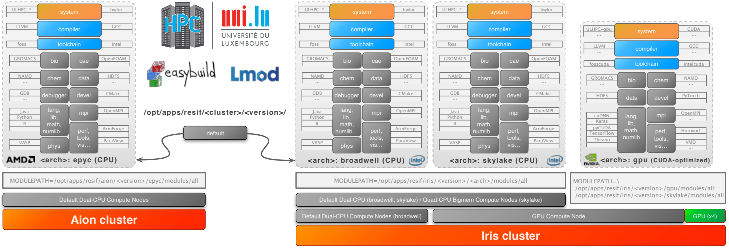
**Operating System: Linux CentOS/Redhat**

- **User Single Sign-on**: Redhat IdM/IPA
- **Remote connection & data transfer**: SSH/SFTP
  - ↪ **User Portal**: Open OnDemand
- **Scheduler/Resource management**: Slurm
- **(Automatic) Server / Compute Node Deployment**:
  - ↪ BlueBanquise, Bright Cluster Manager, Ansible, Puppet and Kadeploy
- **Virtualization and Container Framework**: KVM, Singularity
- **Platform Monitoring** (User level): Ganglia, SlurmWeb, OpenOndemand. . .
- **ISV software**: ABAQUS, ANSYS, MATLAB, Mathematica, Gurobi Optimizer, Intel Cluster Studio XE, ARM Forge & Perf. Report, Stata, . . .

# Accelerating Research - User Software Sets

- **Over 270 software packages** available for researchers
  - ↪ software environment generated using RESIF 3.0 framework over Easybuild
    - ✓ **optimized builds** organized by architecture, exposed through Environment Modules/Lmod
    - ✓ Categorized Naming Scheme <category>/<name>/<version>-<toolchain><versionsuffix>
  - ↪ containerized applications delivered with Singularity system




[ACM PEARC'21] S. Varrette, E. Kieffer, F. Pinel, E. Krishnasamy, S. Peter, H. Cartiaux, and X. Besseron. *RESIF 3.0: Toward a Flexible & Automated Management of User Software Environment on HPC facility*. In Practice & Experience in Advanced Research Computing (PEARC'21), July 2021

# ULHPC Bundles and Software Set Versioning

- **[bi-]Yearly** release of the ULHPC Software Set
  - ↪ following on Easybuid release of toolchains
    - ✓ see Component versions in foss and intel toolchains.
    - ✓ $\simeq$ 6 months of validation/import delay after EB release
- Public export of **RESIF 3 code** on Github: ULHPC/sw

| Toolchain Component | Software set release <version> 2019a (*deprecated*) | 2019b old | 2020a prod | 2021a* devel |
|---|---|---|---|---|
| GCCCore | 8.2.0 | 8.3.0 | 9.3.0 | 10.3.0 |
| foss | 2019a | 2019b | 2020a | 2021a |
| intel | 2019a | 2019b | 2020a | 2021a |
| binutils | 2.31.1 | 2.32 | 2.34 | 2.36 |
| Python | 3.7.2 (2.7.15) | 3.7.4 (2.7.16) | 3.8.2 (2.7.18) | 3.9.2 |
| LLVM | 8.0.0 | 9.0.1 | 10.0.1 | 11.1.0 |
| OpenMPI | 3.1.4 | 3.1.4 | 4.0.3 | 4.1.1 |
| RESIF version | 2.0 (*old*) | 3.0 | 3.0 | 3.1 |
| **#Modules:** | 229 | <arch>: 269 gpu: 135 | <arch>: 274 gpu: 151 | <arch>: n/a gpu: n/a |

| Bundle Name | Description | Featured applications |
|---|---|---|
| ULHPC-<version> | Default global bundle for 'regular' nodes | ULHPC-*-<version> (root bundle) |
| ULHPC-toolchains-<version> | Toolchains, compilers, debuggers, programming languages, MPI suits, Development tools and libraries | GCCcore, foss, intel, LLVM, OpenMPI, CMake, Go, Java, Julia, Python, Spack... |
| ULHPC-bd-<version> | Big Data | Apache Spark, Flink, Hadoop... |
| ULHPC-bio-<version> | Bioinformatics, biology and biomedical | GROMACS, Bowtie2, TopHat, Trinity... |
| ULHPC-cs-<version> | Computational science, incl. CAE, CFD, Chemistry, Earth Sciences, Physics and Materials Science | ANSYS, OpenFOAM, ABAQUS, NAMD, GDAL, QuantumExpresso, VASP... |
| ULHPC-dl-<version> | AI / Deep Learning / Machine Learning | TensorFlow, PyTorch, Horovod... |
| ULHPC-math-<version> | High-level mathematical software and Optimizers | R, MATLAB, CPLEX, GEOS, GMP, Gurobi... |
| ULHPC-perf-<version> | Performance evaluation / Benchmarks | ArmForge, PAPI, HPL, IOR, Graph500... |
| ULHPC-tools-<version> | General purpose tools | DMTC, Singularity, gocryptfs... |
| ULHPC-visu-<version> | Visualization, plotting, documentation & typesetting | OpenCV, ParaView... |
| ULHPC-gpu-<version> | Specific GPU/CUDA-accelerated software | {foss,intel}cuda, cuDNN, TensorFlow, PyTorch, GROMACS... |

# HPC in Luxembourg and Around in EU

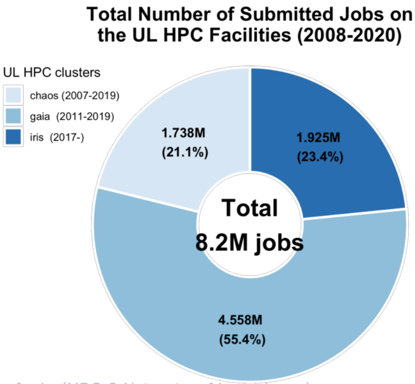
Tier 0: EU
Tier 1: National
Tier 2: Regional | Univ.

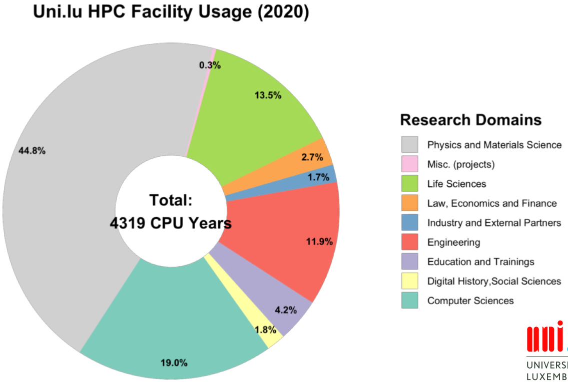
| Country | System(s) | Type | Institute | (CPU) #Nodes | #Cores | #[GPU]Accelerators | $R_{peak}$ | Shared Storage |
|---|---|---|---|---|---|---|---|---|
| | MeluXina (2021) | Euro-HPC **Petascale** *Tier 0/1 (EU,Nat)* | LuxProvide | 824 | ≃ 88 000 | 764 NVidia A100 | **17.57 PF** | ≃ 20 PB |
| Luxembourg | aion,iris | *Tier 2 (Univ)* | **Uni.lu HPC** | **552** | 46896 | **96** NVidia V100 | **2.79 PF** | **10.71 PB** |
| | | *Tier 2 (local)* | LIST | 40 | 1280 | 8 Nvidia V100 | 0.126 PF | 0.58 PB |
| | TGCC (Joliot-Curie) | *Tier 0 (EU)* | GENCI/CEA | 4808 | 430 448 | 828 Xeon Phi, 128 NVidia V100 | 22.26 PF | 35PB |
| France | JeanZay | *Tier 1 (Nat.)* | GENCI/Idris | 1 528 | 61 120 | 1292 NVidia V100 | 14.97 PF | 31.2 PB |
| | ROMEO | *Tier 2 (Reg.)* | Univ. Reims | 115 | 3 220 | 280 NVidia P100 | 1.75 PF | 0.634 |
| | Vlaams | *Tier 1 (Nat.)* | VSC | 988 | 27 664 | n/a | 1.63 PF | 1.3PB |
| Belgium | zenobe | *Tier 1 (Nat.)* | Cenaero | 584 | 14 016 | 4 NVidia K40 | 0.41 PF | 0.356PB |
| | Hortense | *Tier 2 (Reg.)* | Gent Univ. | n/a | ≃ 40 000 | 88 NVidia V100 | 3.3PF | 3PB |
| | JUWELS | *Tier 0 (EU)* | JSC | 2571 | 122 768 | 224 Nvidia V100 | 12.3 PF | 130.3PB |
| | JURECA | *Tier 0 (EU)* | JSC | 3524 | 156 736 | 1640 Xeon Phi | 7.24 PF | (as above) |
| Germany | Hawk | *Tier 0 (EU)* | HLRS, Univ. Stuttgart | 5632 | 720 896 | n/a | 26 PF | ≃25PB |
| | SuperMUC-NG | *Tier 0 (EU)* | LRZ, Munich | 6480 | 311 040 | n/a | 26.9 PF | 70.16PB |
| | CLAIX-2018 | *Tier 2 (Univ)* | Univ. Aachen | 1307 | 61 200 | 108 Nvidia V100 | 4.11 PF | 3PB |
| Bulgaria | PetaSC (2021) | Euro-HPC **Petascale** *Tier 0/1 (EU,Nat)* | SofiaTech | n/a | n/a | n/a | **4.5 PF** | n/a |
| Czech Republic | Barbora | *Tier 1 (Nat.)* | IT4Innovation | 201 | 7232 | 32 NVidia V100 | 0.85 PF | ≃ 1PB |
| | Karolina (2021) | Euro-HPC **Petascale** *Tier 0/1 (EU,Nat)* | IT4Innovation | 826 | ≃ 100K | 560 NVidia A100 | **9.4 PF** | 1PB |
| Finland | LUMI (2021) | Euro-HPC **Pre-exascale** *Tier 0 (EU)* | CSC | n/a | ≃ 200K (LUMI-C) | n/a | **375 PF** | 127PB |
| | Marconi-A3 | *Tier 0 (EU)* | Cineca | 3216 | 154 368 | n/a | 10.37 PF | 10PB |
| Italy | Galileo | *Tier 1 (Nat.)* | Cineca | 1022 | 36792 | n/a | 1.35 PF | 1.92PB |
| | Leonardo (2021) | Euro-HPC **Pre-exascale** *Tier 0 (EU)* | Cineca | 4992 | n/a | 13824 Nvidia A100 | **249.5 PF** | 100PB |
| Portugal | Deucalion (2021) | Euro-HPC **Petascale** *Tier 0/1 (EU,Nat)* | MACC | n/a | n/a | n/a | **7.2 PF** | n/a |
| Slovenia | VEGA (2021) | Euro-HPC **Petascale** *Tier 0/1 (EU,Nat)* | Maribor SC | 960 | 122,8K | 240 NVidia A100 | **10.1 PF** | 24 PB |
| Spain | MareNostrum 4 | *Tier 0 (EU)* | BSC | 3456 | 165 888 | n/a | 11.15 PF | 14PB |
| | MareNostrum 5 (2021) | Euro-HPC **Pre-exascale** *Tier 0 (EU)* | BSC | n/a | n/a | n/a | **≃ 200 PF** | n/a |
| Switzerland | Piz-Daint | *Tier 0 (EU)* | CSCS, ETH Zürich | 7517 | 387 872 | 5704 NVidia P100 | 29.34 PF | 8.8PB |

# Summary

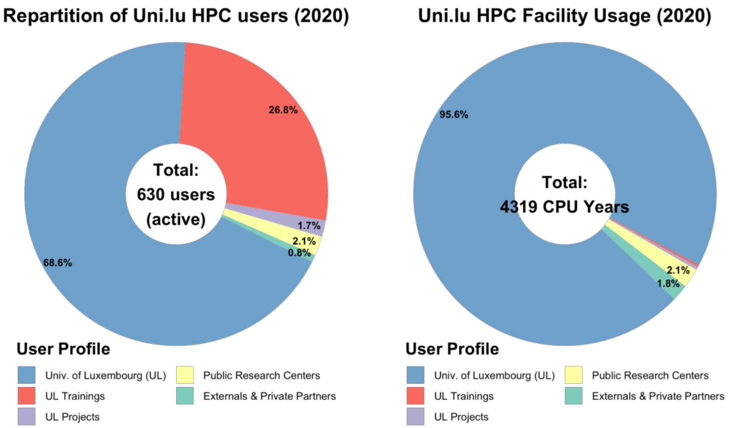
# Uni.lu HPC Users

- **1518** registered HPC Users - **630 actives** in 2020
  - ↪ 23 computational domains accelerated on UL HPC

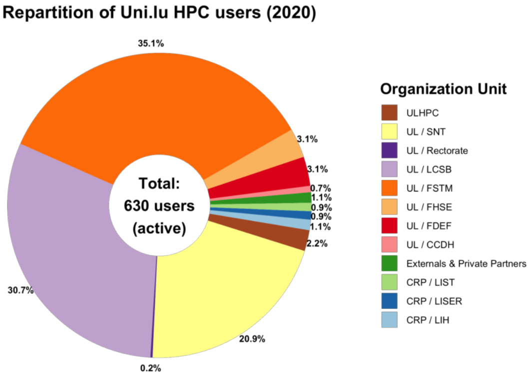


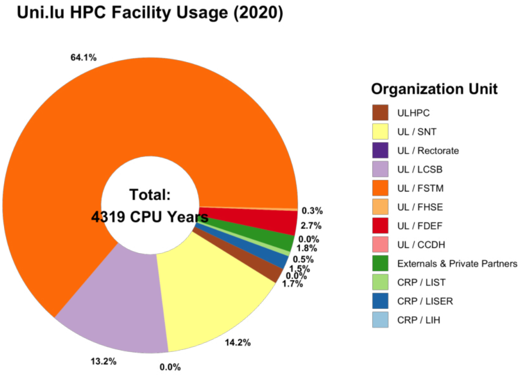



# Uni.lu HPC Users Past Year Statistics

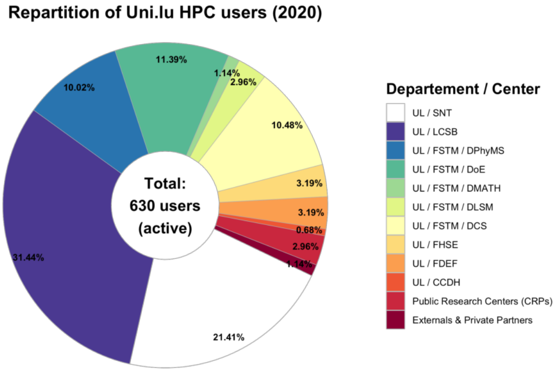


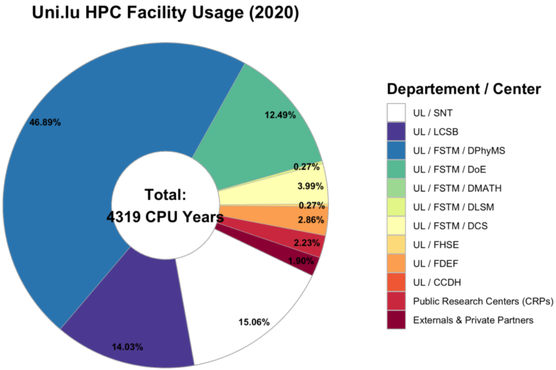
# Uni.lu HPC Users Past Year Statistics



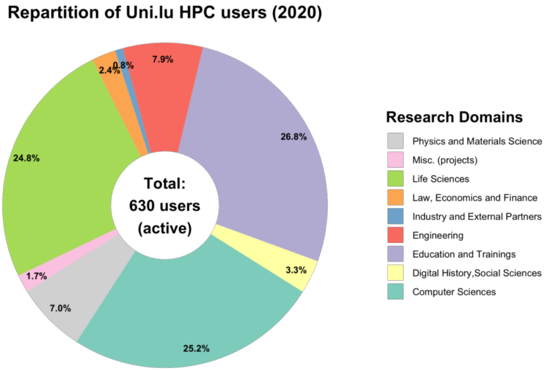



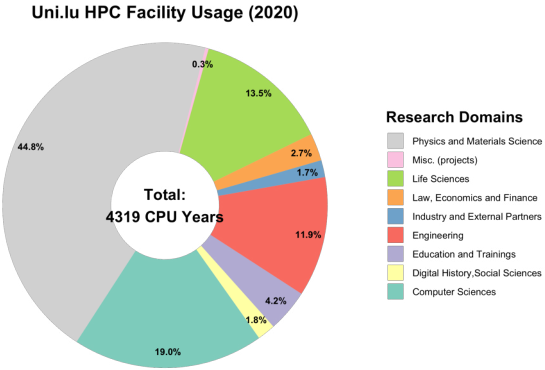
# Uni.lu HPC Users Past Year Statistics



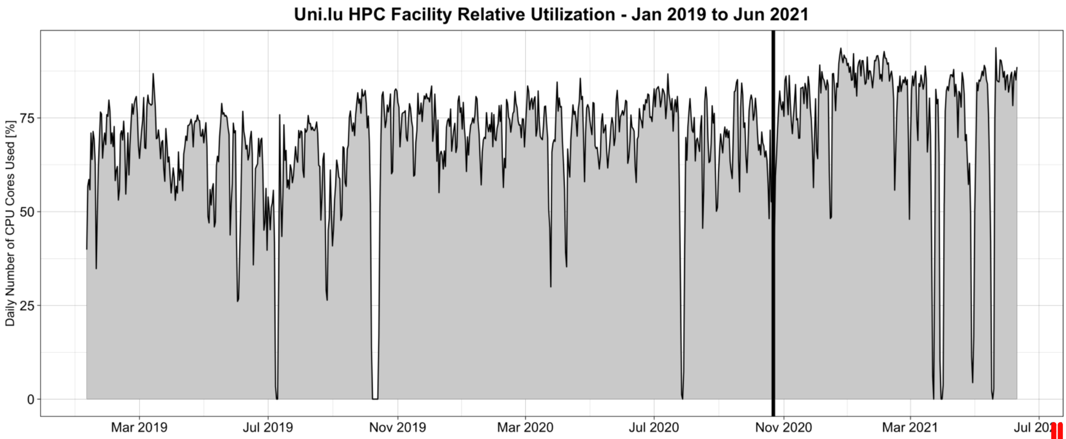



# Uni.lu HPC Users Past Year Statistics

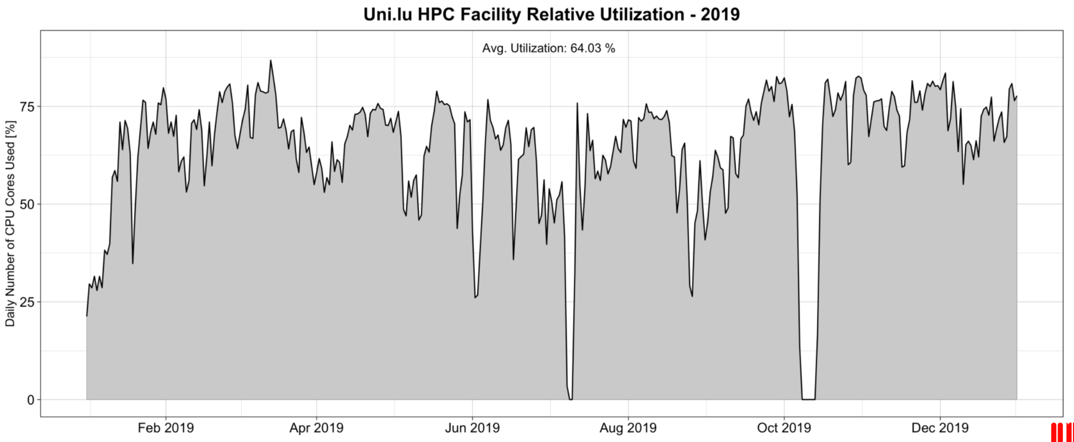


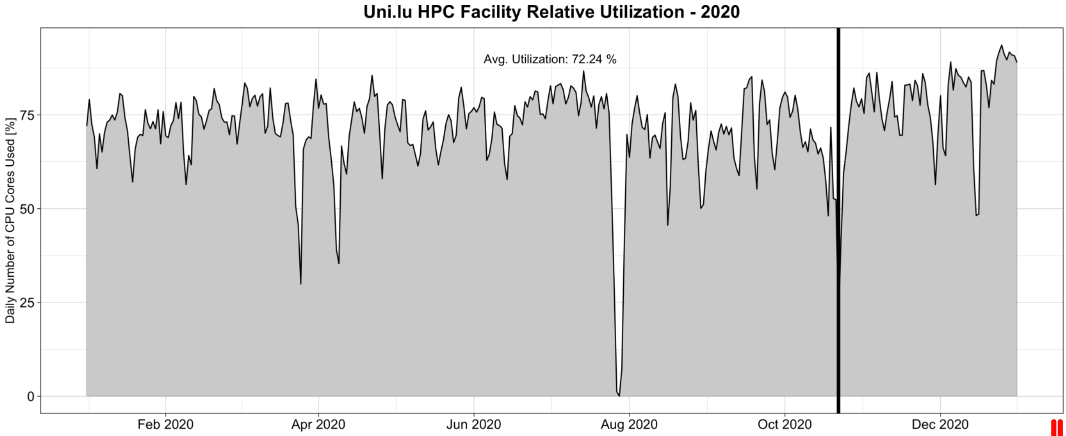
# Cluster Utilization (iris)



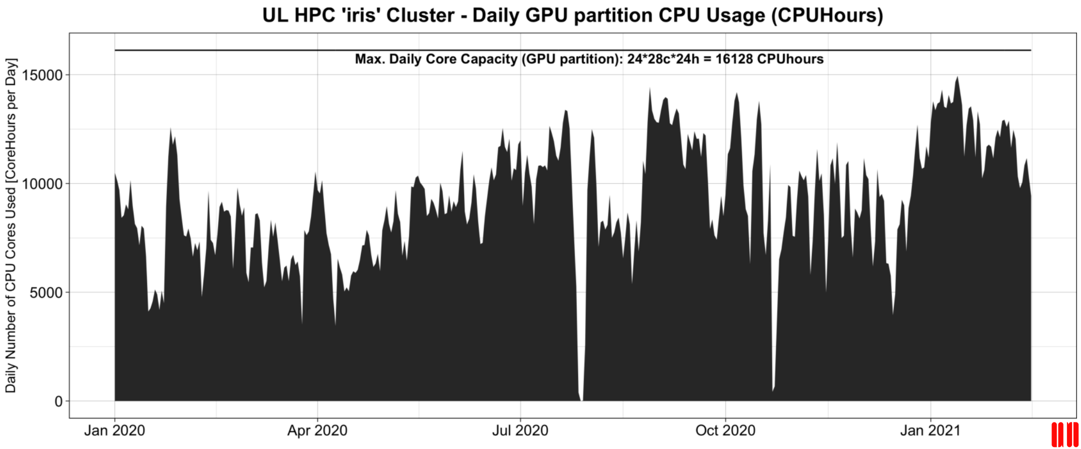
# Cluster Utilization (iris)



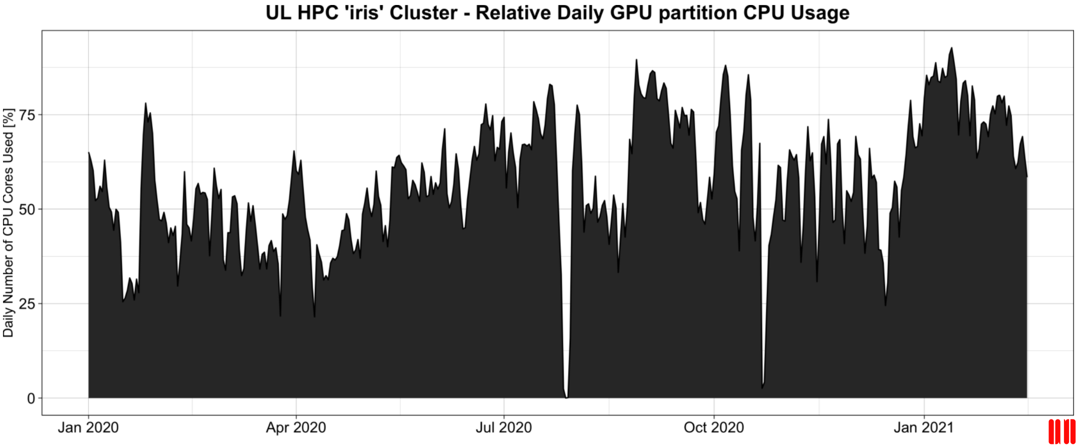
# Cluster Utilization (iris)



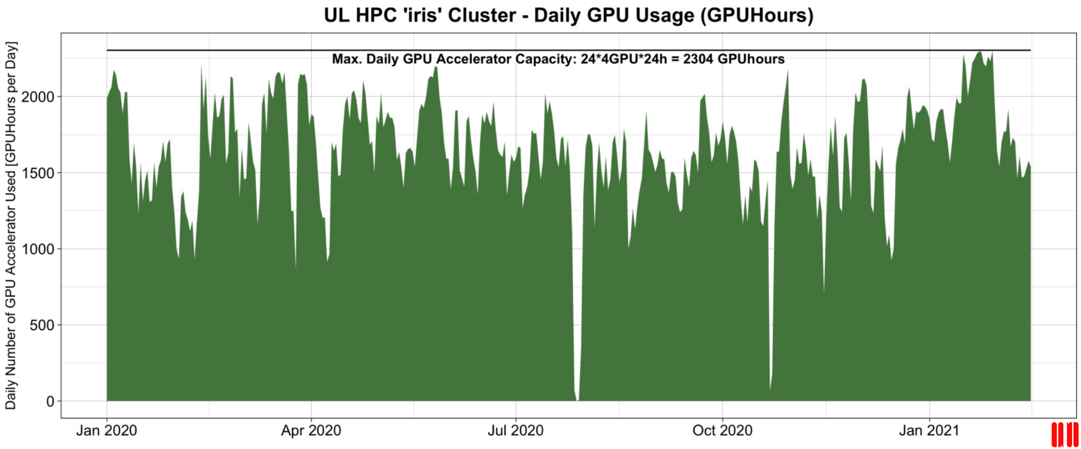
# GPU Accelerated Computing Node Usage



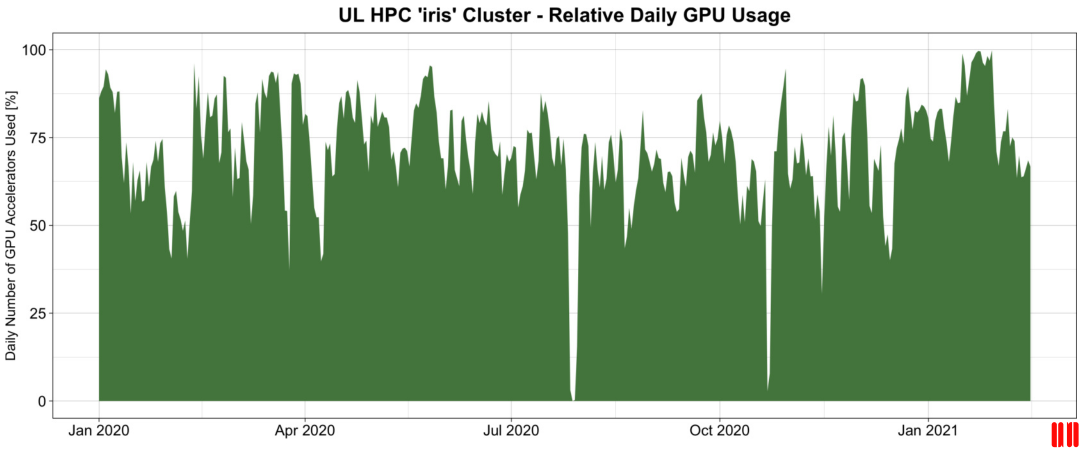
# GPU Accelerated Computing Node Usage



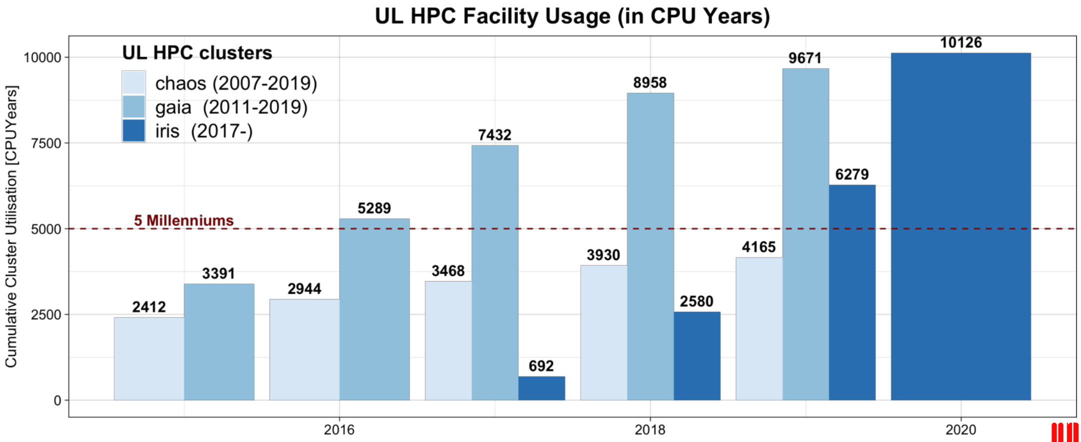
# GPU Accelerated Computing Node Usage



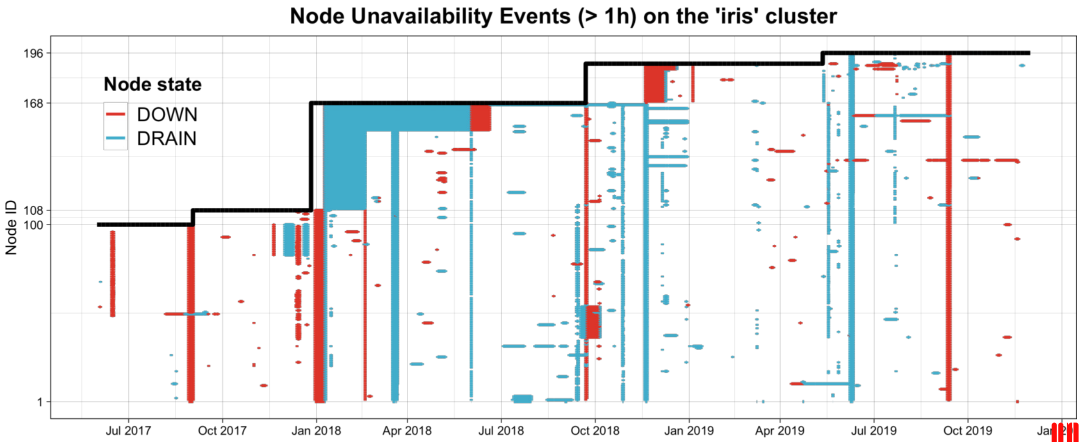
# GPU Accelerated Computing Node Usage



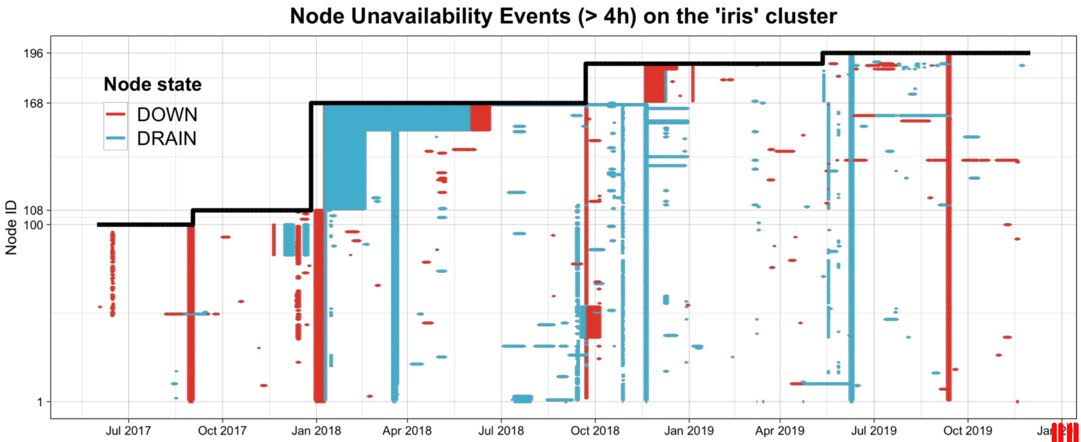
# Uni.lu HPC Cumulative Usage



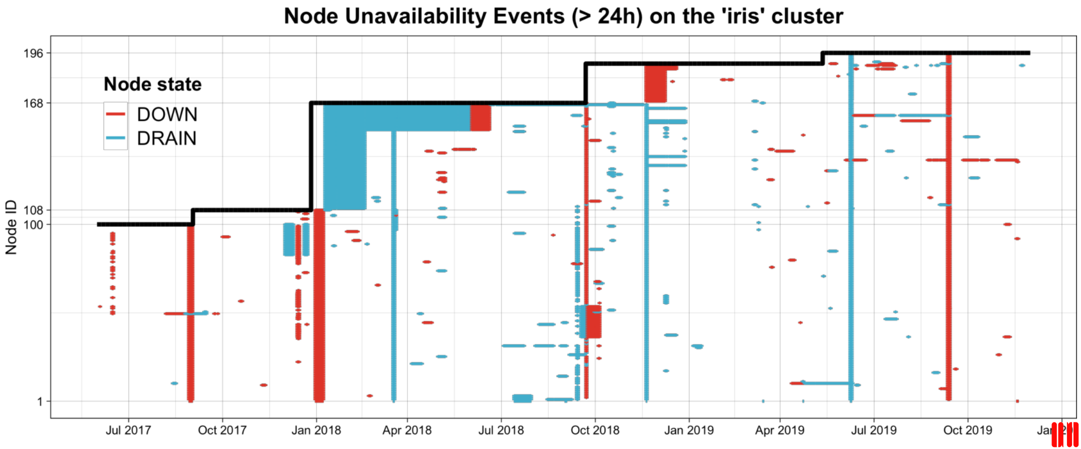
# Node Events and [un]Availability (iris, 2017-2019)



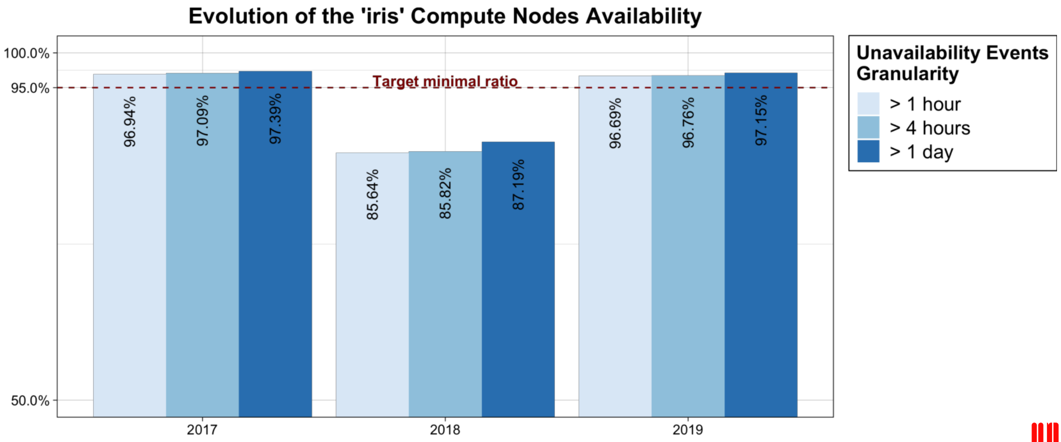
# Node Events and [un]Availability (iris, 2017-2019)



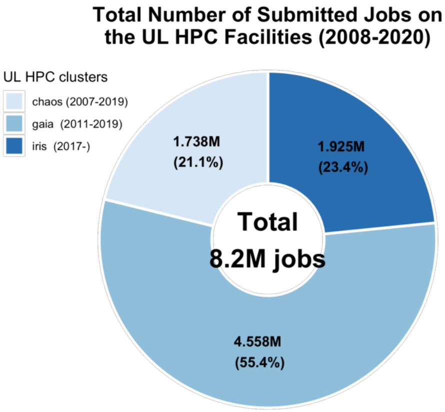
# Node Events and [un]Availability (iris, 2017-2019)



# Node Events and [un]Availability (iris, 2017-2019)



# Number of submitted Jobs

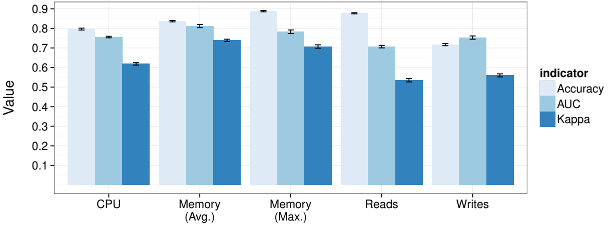


# OAR Jobs Characterization and Predictions

- First analysis on [2014] Gaia OAR traces [JSSPP'15]
  - ↪ Classification of Jobs Consumption
  - ↪ Supervised Learning Algorithm for the Complete Prediction of Job Resource Consumption

| | CPU | Memory | Disk IO |
|---|---|---|---|
| **Class 1** | 0 to 25% | up to 10MiB | up to 10KiB/s |
| **Class 2** | 25 to 50% | from 10 to 100MiB | from 10 to 100KiB/s |
| **Class 3** | 50 to 75% | from 100 to 1GiB | from 100 to 1MiB/s |
| **Class 4** | 75 to 100% | above 1GiB | above 1MiB/s |

Table 2: Clustering classes for consumption metrics.




[JSSPP'15] J. Emeras, S. Varrette, M. Guzek, and P. Bouvry, "*Evalix: Classification and Prediction of Job Resource Consumption on HPC Platforms*" in Proc. of the 19th Intl. Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP'15), part of the 29th IEEE/ACM Intl. Parallel and Distributed Processing Symposium (IPDPS 2015), Hyderabad, India, 2015.

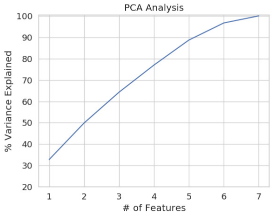
# [preliminary] Slurm Job Characterization

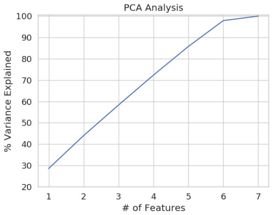
- New ***preliminary*** analysis on Iris Slurm traces (master training T. Gennuso 2019)
  - ↪ Slurm accounting brings ≥ 107 potential features

```
$ sacct --helpformat|wc -w
107
```

  - ↪ Dimensional reduction by PCA *(Principal Component Analysis)* to **7 features** (10K then 100K jobs)

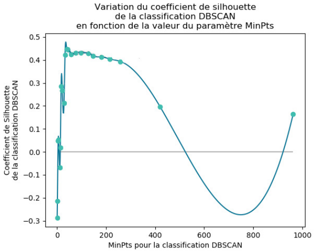
1. `elapsed`
2. `ncpus`
3. `ave_cpu`
4. `ave_rss`
5. `ave_disk_read`
6. `ave_disk_write`
7. `consumed_energy`

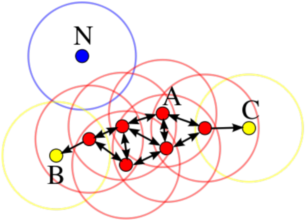


# [preliminary] Slurm Job Characterization

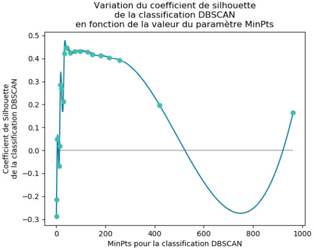
- Clustering based on `scikit-learn` **DBSCAN** Density-Based Spatial Clustering of Applications with Noise
  - ↪ unsupervised learning algorithm mostly influence by two hyperparameters:
    - ✓ $\varepsilon$: radius / max. distance between two samples to be considered neighbors
    - ✓ *MinPts* (`min_samples`): #samples (or total weight) in a neighborhood for a point to be considered as a core point. **Ex**: MinPts=4, red points are *core* points

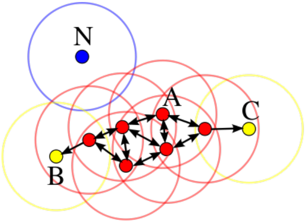


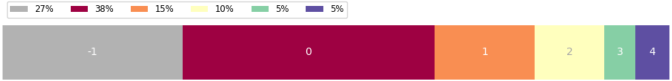
# [preliminary] Slurm Job Characterization

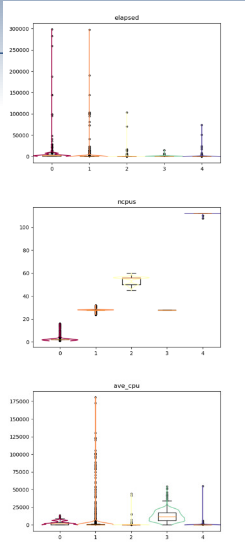
- Clustering based on `scikit-learn` **DBSCAN** Density-Based Spatial Clustering of Applications with Noise
  - ↪ unsupervised learning algorithm mostly influence by two hyperparameters:
    - ✓ $\varepsilon$: radius / max. distance between two samples to be considered neighbors
    - ✓ *MinPts* (`min_samples`): #samples (or total weight) in a neighborhood for a point to be considered as a core point. **Ex**: MinPts=4, red points are *core* points
  - ↪ (parallel) hyper-parameter tuning by maximizing **silhouette coefficient**
    - ✓ **Ex** for *fixed* $\varepsilon$ on 10K jobs



# [preliminary] Slurm Job Characterization

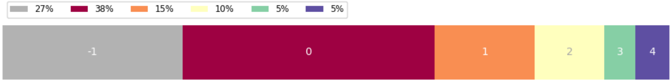



- 5 classes emerged by the clustering:
  - ↪ Tentative descriptions from main characteristics of the selected

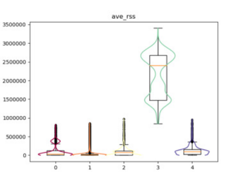
| Class | Elapsed | #Cores | CPU Load | Memory | I/O read | I/O write | Energy |
|---|---|---|---|---|---|---|---|
| 0 | ++ | + | + | ++ | +++ | +++ | +++ |
| 1 | ++ | ++ | + | + | + | + | ++ |
| 2 | + | ++ | + | ++ | + | ++ | + |
| 3 | + | + | ++ | +++ | + | +++ | ++ |
| 4 | + | +++ | + | ++ | + | + | + |

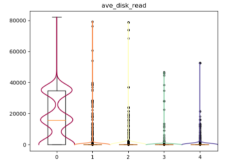
# [preliminary] Slurm Job Characterization

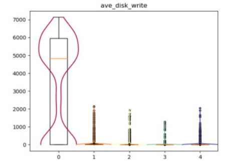



- 5 classes emerged by the clustering:
  - ↪ Tentative descriptions from main characteristics of the selected

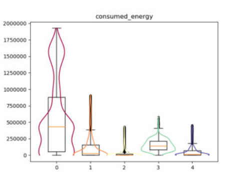
| Class | Elapsed | #Cores | CPU Load | Memory | I/O read | I/O write | Energy |
|---|---|---|---|---|---|---|---|
| 0 | ++ | + | + | ++ | +++ | +++ | +++ |
| 1 | ++ | ++ | + | + | + | + | ++ |
| 2 | + | ++ | + | ++ | + | ++ | + |
| 3 | + | + | ++ | +++ | + | +++ | ++ |
| 4 | + | +++ | + | ++ | + | + | + |

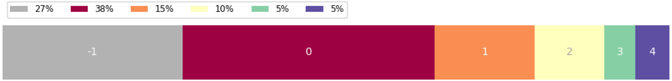


# [preliminary] Slurm Job Characterization



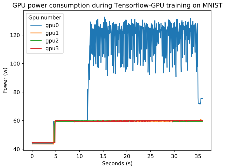




- 5 classes emerged by the clustering:
  - ↪ Tentative descriptions from main characteristics of the selected

| Class | Elapsed | #Cores | CPU Load | Memory | I/O read | I/O write | Energy |
|---|---|---|---|---|---|---|---|
| 0 | ++ | + | + | ++ | +++ | +++ | +++ |
| 1 | ++ | ++ | + | + | + | + | ++ |
| 2 | + | ++ | + | ++ | + | ++ | + |
| 3 | + | + | ++ | +++ | + | +++ | ++ |
| 4 | + | +++ | + | ++ | + | + | + |

# Going further for Energumen (and other projects)

- **Investigate 2017-2020 Slurm traces** (Iris and incoming Aion ones)
  - ↪ hierarchical DBSCAN (HDBSCAN*)
  - ↪ Going above Intel RAPL energy monitoring part of Slurm accounting
    - ✓ nvml-based GPU consumption
  - ↪ integrate LMod Module Usage Tracking DB




```
Module path                                          Distinct Users
-----------                                          --------------
[...]/modules/all/compiler/GCCcore/8.2.0.lua           200
[...]
[...]/modules/all/lang/Python/3.7.2-GCCcore-8.2.0.lua  122
[...]
[...]/modules/all/numlib/FFTW/3.3.8-gompi-2019a.lua    71
[...]/modules/all/toolchain/iimpi/2019a.lua            115
```

# Going further for Energumen (and other projects)

- Exploit new monitoring tools deployed in Luxembourg
  - ↪ **NVIDIA Data Center GPU Manager** (DCGM) telemetry and Job Statistics
    - ✓ WIP: Prometheus integration
- NVidia/Mellanox power savings on IB network? Integration of DC measures

# Going further for Energumen (and other projects)

- Exploit new monitoring tools deployed in Luxembourg
  - ↪ **NVIDIA Data Center GPU Manager** (DCGM) telemetry and Job Statistics
    - ✓ WIP: Prometheus integration
- NVidia/Mellanox power savings on IB network? Integration of DC measures

# Going further for Energumen (and other projects)

- Exploit new monitoring tools deployed in Luxembourg
  - ↪ **NVIDIA Data Center GPU Manager** (DCGM) telemetry and Job Statistics
    - ✓ WIP: Prometheus integration
- NVidia/Mellanox power savings on IB network? Integration of DC measures
- HPC I/O Monitoring with Darshan
  - ↪ *"Uncovering Access, Reuse, and Sharing Characteristics of I/O-Intensive Files on Large-Scale Production HPC Systems"* - PDF
  - ↪ Complement: NERSC `pytokio`
- Collaboration/contribution to extensions of BatSim, MojitO/S and COLMET
  - ↪ Slurm integration, validation via slurm-replay?
  - ↪ redefining SWF: integrate backfilling, fair sharing and raw share information

# Summary

# Context & Motivations

## Aion cluster installation

- Incoming arrival of Aion cluster is the occasion to review the Slurm configuration
  - ↪ take advantage of experience gained after 3 years of production
  - ↪ get rid of over-complex setup
    - ✓ KISS for users and HPC staff
  - ↪ **transparent** model for fair-sharing giving **incentives** to good practices
    - ✓ take into account monetary contribution to increase priority
    - ✓ tribute to past **efficient** usage
  - ↪ consolidate setup for HA and common accounting between the clusters
  - ↪ update account hierarchy to meet projects/training needs

# Overview of the Main Configuration Changes (1/3)

- Simplified partitions with max walltime reduction
- **Interactive Jobs** moved over **floating** partitions with feature selection
  - ↪ *Before*: guaranteed access to interative jobs on regular nodes even if `batch` partition full
  - ↪ *After*: no guarantee if partition is full **YET** backfilling and priority ensure first served

| Node Type | Slurm command | Helper script |
|---|---|---|
| regular | `srun -p interactive --qos debug -C batch [-C {broadwell,skylake}] [...] --pty bash` | `si [...]` |
| gpu | `srun -p interactive --qos debug -C gpu [-C volta[32]] -G 1 [...] --pty bash` | `si-gpu [...]` |
| bigmem | `srun -p interactive --qos debug -C bigmem [...] --pty bash` | `si-bigmem [...]` |

# Overview of the Main Configuration Changes (2/3)

- **Regular Jobs**: accounting consolidated, QOS redefinition for priority enforcement
  ↪ walltime restricted to **Max 2 days** / Max 64 nodes

| Node Type | Slurm command |
|---|---|
| regular | `sbatch [-A <project>] -p batch  [--qos {high,urgent}] [-C {broadwell,skylake}] [...]` |
| gpu | `sbatch [-A <project>] -p gpu    [--qos {high,urgent}] [-C volta[32]] -G 1 [...]` |
| bigmem | `sbatch [-A <project>] -p bigmem [--qos {high,urgent}] [...]` |

# Overview of the Main Configuration Changes (2/3)

- **Regular Jobs**: accounting consolidated, QOS redefinition for priority enforcement
  - ↪ walltime restricted to **Max 2 days** / Max 64 nodes

| Node Type | Slurm command |
|---|---|
| regular | `sbatch [-A <project>] -p batch [--qos {high,urgent}] [-C {broadwell,skylake}] [...]` |
| gpu | `sbatch [-A <project>] -p gpu [--qos {high,urgent}] [-C volta[32]] -G 1 [...]` |
| bigmem | `sbatch [-A <project>] -p bigmem [--qos {high,urgent}] [...]` |

- **Long Jobs** restricted Max walltime (`MaxWall`) to **14 days** EuroHPC/PRACE Recommendations
  - ↪ Max 6 nodes, Max 2 nodes per Job, Max **1** Job per User

# Overview of the Main Configuration Changes (2/3)

- **Regular Jobs**: accounting consolidated, QOS redefinition for priority enforcement
  - ↪ walltime restricted to **Max 2 days** / Max 64 nodes

| Node Type | Slurm command |
|---|---|
| regular | `sbatch [-A <project>] -p batch [--qos {high,urgent}] [-C {broadwell,skylake}] [...]` |
| gpu | `sbatch [-A <project>] -p gpu [--qos {high,urgent}] [-C volta[32]] -G 1 [...]` |
| bigmem | `sbatch [-A <project>] -p bigmem [--qos {high,urgent}] [...]` |

- **Long Jobs** restricted Max walltime (`MaxWall`) to **14 days** EuroHPC/PRACE Recommendations
  - ↪ Max 6 nodes, Max 2 nodes per Job, Max **1** Job per User
- **Slurm Federation configuration** between `iris` and `aion`
  - ↪ ensures global policy (coherent job ID, global scheduling, etc.) within ULHPC systems
  - ↪ easily submit jobs from one cluster to another `-M, --cluster aion|iris`

# Overview of the Main Configuration Changes (3/3)

- (complex) Depth-Oblivious Fairshare $\Longrightarrow$ **Fair tree** Algorithm (rooted plane tree)
  - ↪ if accounts A and B are siblings and A has a higher fairshare factor than B, then all children of A will have higher fairshare factors than all children of B.
- **Review backfilling params**: updated windows, refresh rate to opt. interactive/small jobs

```
SchedulerParameters=bf_window=14400,bf_resolution=600,bf_interval=60,bf_continue,\
                    bf_max_job_test=500,sbatch_wait_nodes,salloc_wait_nodes
```

- **Cross-partition QOS**, mainly tied to **priority level** (low → urgent)
  - ↪ special **preemptible QOS** kept for **best-effort Jobs** **YET** renamed: ~~qos-~~besteffort
  - ↪ **restricted `urgent` QOS** for ultra-high priority jobs (Ex: covid-19)
- **New cost model and account hierarchy**
  - ↪ incentives for User groups/Projects contributing to the HPC budget line
  - ↪ end-User raw-share increased based on past year efficiency

# ULHPC Slurm Partitions 2.0

`-p, -partition=<partition>`

```
$> {srun|sbatch|salloc|sinfo|squeue...} -p <partition> [...]
```

| **AION** Partition | Type | #Node | PriorityTier | DefaultTime | MaxTime | MaxNodes |
|---|---|---|---|---|---|---|
| interactive | floating | 318 | 100 | 30min | 2h | 2 |
| batch | | 318 | 1 | 2h | 48h | 64 |

| **IRIS** Partition | Type | #Node | PriorityTier | DefaultTime | MaxTime | MaxNodes |
|---|---|---|---|---|---|---|
| interactive | floating | 196 | 100 | 30min | 2h | 2 |
| batch | | 168 | 1 | 2h | 48h | 64 |
| gpu | | 24 | 1 | 2h | 48h | 4 |
| bigmem | | 4 | 1 | 2h | 48h | 1 |

# ULHPC Slurm QOS 2.0

--qos=<qos>

```
$> {srun|sbatch|salloc|sinfo|squeue...} [-p <partition>] --qos <qos> [...]
```

| QOS | Partition | Allowed [L1] Account | Prio | GrpTRES | MaxTresPJ | MaxJobPU | Flags |
|---|---|---|---|---|---|---|---|
| besteffort | * | **ALL** | 1 | | | 100 | NoReserve |
| low | * | **ALL** (default for CRP/externals) | 10 | | | 2 | DenyOnLimit |
| normal | * | **Default** (UL,Projects,. . . ) | 100 | | | 50 | DenyOnLimit |
| long | * | UL,Projects,etc. | 100 | node=6 | node=2 | 1 | DenyOnLimit,PartitionTimeLimit |
| debug | interactive | **ALL** | 150 | node=8 | | 2 | DenyOnLimit |
| high | * | (**restricted**) UL,Projects,Industry | 200 | | | 10 | DenyOnLimit |
| urgent | * | (**restricted**) UL,Projects,Industry | 1000 | | | 100 ? | DenyOnLimit |

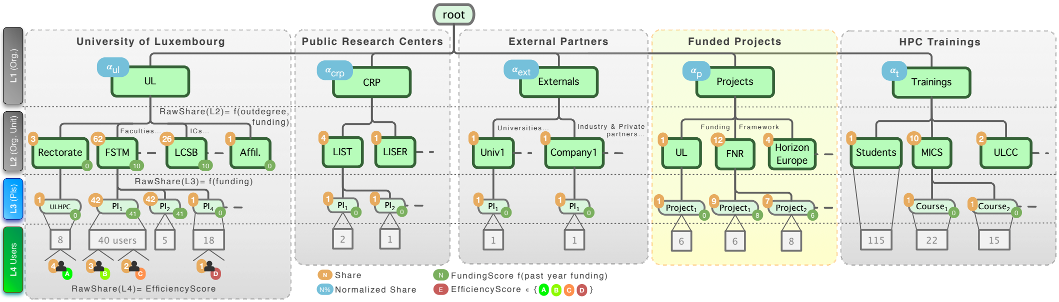
- **Cross-partition QOS**, mainly tied to **priority level** (low → urgent)
  - ↪ Simpler names than before (i.e. no more qos- prefix)
  - ↪ special **preemptible QOS** for best-effort jobs: `besteffort`

# Account Hierarchy 2.0

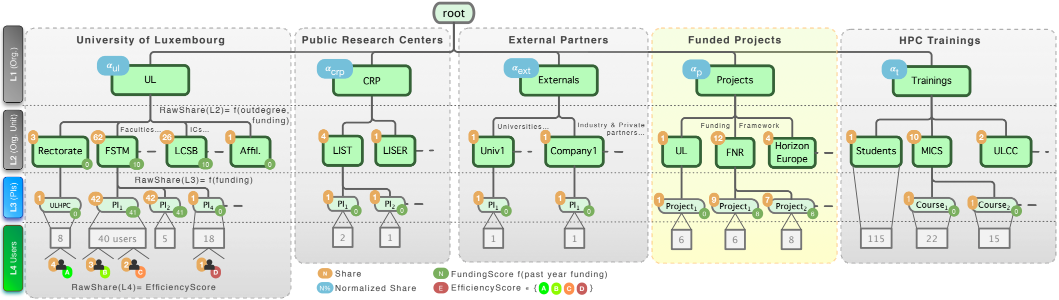
- Every user job runs under a group account
  - ↪ granting access to specific QOS levels.
  - ↪ default raw share for accounts: 1

- **L1**: Organization Level: UL, CRPs, Externals, Projects, Trainings
  - ↪ guarantee 85% of the shares for core UL activities
- **L2**: Organizational Unit (Faculty, ICs, External partner, Funding program. . . )
  - ↪ Raw share depends on **outdegree** and **funding score**
- **L3**: Principal Investigator (PIs), Projects, Course
  - ↪ Raw share depends on **funding score** (different weight)
  - ↪ Eventually restricted **only** to projects and courses
- **L4**: End User (ULHPC login)
  - ↪ Raw share based on **efficiency score**

# Account Hierarchy 2.0



# Account Hierarchy 2.0




```
# L1,L2 or L3 account /!\ ADAPT <name> accordingly
sacctmgr show association tree where accounts=<name> format=account,share
# End user (L4)
sacctmgr show association where users=$USER  format=account,User,share,Partition,QOS
```

# Funding Score (L2/L3)

- Associated with an account $A$ belonging to a level $L$ in the hierarchy
  - ↪ yearly updated at the beginning of the year
  - ↪ depreciation based on contribution type

$$\texttt{FundingScore}_L(A) = \left\lfloor \beta_L \frac{Investment_A(Year-1)}{\#months} \right\rfloor$$

- **Ex1**: Exceptional contribution of 120K€ performed in 2020 by a faculty (L2 account $A$)
  - ↪ depreciation: 12 months (*default*)
  - ↪ **funding score in 2021**: $\left\lfloor \beta_{L_2} \frac{120000}{12} \right\rfloor = \lfloor \beta_{L_2} \times 10000 \rfloor$.
- **Ex2**: let $P$ be a project granted in 2020 to start in 2021 for a duration of 36 months
  - ↪ **budget**: 27K€ allocated for HPC costs
  - ↪ **funding score for the years 2021, 2022 and 2023**: $\left\lfloor \beta_{L_3} \frac{27000}{36} \right\rfloor = \lfloor \beta_{L_3} \times 750 \rfloor$

# Efficiency Score (L4)

- **Updated every year based on past jobs efficiency.**
  - ↪ Similar notion of "nutri-score": A(very good - 3), B (good: 2), C (bad, 1), D(very bad - 0)
- Proposed Metric for **user** $U$: **Average Wall-time Accuracy (WRA)** (higher the better)
  - ↪ Defined for a given time period (past year)

```
sacct -u <U> -X -S <start> -E <end> [...] # --format User,JobID,state,time,elapsed
```

  - ↪ Reduction for $N$ `COMPLETED` jobs:

$$S_{\text{efficiency}}(U, Year) = \frac{1}{N} \sum_{JobID \in (U, Year)} \frac{T_{\text{elapsed}}(JobID)}{T_{\text{asked}}(JobID)}$$

- Default thresholds

| Score | Avg. WRA |
| --- | --- |
| A | $S_{\text{efficiency}} \geq 75\%$ |
| B | $50\% \leq S_{\text{efficiency}} < 75\%$ |
| C | $25\% \leq S_{\text{efficiency}} < 50\%$ |
| D | $S_{\text{efficiency}} < 25\%$ |

- **WIP**: integrate other efficiency metrics (CPU, mem, GPU efficiency)

# Job Priority, Fairsharing and Fair Tree

- **Fairsharing**: way of ensuring that users get their appropriate portion of a system
  - ↪ **Share**: portion of the system users have been granted.
  - ↪ **Usage**: amount of the system users have actually **used**.
  - ↪ **Fairshare score**: value the system calculates based off of user's usage.
    - ✓ difference between the portion of the computing resource that has been promised and the amount of resources that has been consumed
  - ↪ **Priority score**: priority assigned based off of the user's fairshare score.
- ULHPC Slurm configuration with Multifactor Priority Plugin and **Fair tree** algorithm
  - ↪ rooted plane tree (rooted ordered tree) being created then sorted by Level Fairshare
  - ↪ All users from a higher priority account receive a higher fair share factor than all users from a lower priority account

```
$> sshare -l                                                # See Level FS
```

# ULHPC Job Prioritization Factors

- **Age**: length of time a job has been waiting (PD state) in the queue
- **Fairshare**: difference between the portion of the computing resource that has been promised and the amount of resources that has been consumed
- **Partition**: factor associated with each node partition
  - ↪ Ex: privilege `interactive` over `batch`
- **QOS** A factor associated with each Quality Of Service (`low` ⟶ `urgent`)

```
Job_priority =
    PriorityWeightAge         * age_factor +
    PriorityWeightFairshare   * fair-share_factor+
    PriorityWeightPartition   * partition_factor +
    PriorityWeightQOS         * QOS_factor +
    - nice_factor
```

# ULHPC Job Prioritization Factors

- **Age**: length of time a job has been waiting (PD state) in the queue
- **Fairshare**: difference between the portion of the computing resource that has been promised and the amount of resources that has been consumed
- **Partition**: factor associated with each node partition
  - ↪ Ex: privilege `interactive` over `batch`
- **QOS** A factor associated with each Quality Of Service (`low` ⟶ `urgent`)

```
Job_priority =
    PriorityWeightAge         * age_factor +
    PriorityWeightFairshare   * fair-share_factor+
    PriorityWeightPartition   * partition_factor +
    PriorityWeightQOS         * QOS_factor +
    - nice_factor
```

```
# Show current weights
sprio -w
# List pending jobs, sorted by jobid
sprio [-n]
# List pending jobs, sorted by  priority
sprio [-n] -S+Y
sprio [-n] | sort -k 3 -n
sprio [-n] -l | sort -k 4 -n
```

# Fairshare Factor and Job billing

https://hpc-docs.uni.lu/policies/usage-charging/

- Utilization of the University computational resources is charged in **Service Unit (SU)**
  - ↪ 1 SU $\simeq$ 1 hour on 1 physical processor core on regular computing node
  - ↪ Usage charged **0,03€ per SU (VAT excluded)** (external partners, funded projects etc.)
- A Job is characterized (and thus billed) according to the following elements:
  - ↪ $T_{\text{exec}}$: Execution time (in hours)
  - ↪ $N_{\text{Nodes}}$: number of computing nodes, and **per node**:
    - ✓ $N_{\text{cores}}$: number of CPU cores allocated per node
    - ✓ $Mem$: memory size allocated per node, in GB
    - ✓ $N_{\text{gpus}}$: number of GPU allocated per node
  - ↪ associated weighted factors $\alpha_{cpu}, \alpha_{mem}, \alpha_{GPU}$ defined as `TRESBillingWeight` in Slurm
    - ✓ account for consumed resources other than just CPUs
    - ✓ taken into account in fairshare factor
    - ✓ $\alpha_{cpu}$: normalized relative perf. of CPU processor core (reference: skylake 73,6 GFlops/core)
    - ✓ $\alpha_{mem}$: inverse of the average available memory size per core
    - ✓ $\alpha_{GPU}$: weight per GPU accelerator

# Fairshare Factor and Job billing

## Number of SU associated to a job

$$N_{\text{Nodes}} \times [\alpha_{cpu} \times N_{\text{cores}} + \alpha_{mem} \times Mem + \alpha_{gpu} \times N_{\text{gpus}}] \times T_{\text{exec}}$$

| Cluster | Node Type | Partition | #Cores/node | CPU | $\alpha_{\text{cpu}}$ | $\alpha_{\text{mem}}$ | $\alpha_{\text{GPU}}$ |
|---|---|---|---|---|---|---|---|
| Iris, Aion | Regular | `interactive` | 28/128 | n/a | 0 | 0 | 0 |
| Iris | Regular | `batch` | 28 | `broadwell` | 1.0* | $\frac{1}{4} = 0,25$ | 0 |
| Iris | Regular | `batch` | 28 | `skylake` | 1.0 | $\frac{1}{4} = 0,25$ | 0 |
| Iris | GPU | `gpu` | 28 | `skylake` | 1.0 | $\frac{1}{27}$ | 50 |
| Iris | Large-Mem | `bigmem` | 112 | `skylake` | 1.0 | $\frac{1}{27}$ | 0 |
| Aion | Regular | `batch` | 128 | `epyc` | 0,57 | $\frac{1}{1.75}$ | 0 |

```
#  Billing rate for running job <jobID>
scontrol show job <jobID> | grep -i billing
# Billing rate for completed job <jobID>
sacct -X --format=AllocTRES%50,Elapsed -j <jobID>
```

# Fairshare Factor and Job billing

## Number of SU associated to a job

$$N_{\text{Nodes}} \times [\alpha_{cpu} \times N_{\text{cores}} + \alpha_{mem} \times Mem + \alpha_{gpu} \times N_{\text{gpus}}] \times T_{\text{exec}}$$

| Cluster | Node Type | Partition | #Cores/node | CPU | $\alpha_{\text{cpu}}$ | $\alpha_{\text{mem}}$ | $\alpha_{\text{GPU}}$ |
|---|---|---|---|---|---|---|---|
| Iris, Aion | Regular | `interactive` | 28/128 | n/a | 0 | 0 | 0 |
| Iris | Regular | `batch` | 28 | `broadwell` | 1.0* | $\frac{1}{4} = 0,25$ | 0 |
| Iris | Regular | `batch` | 28 | `skylake` | 1.0 | $\frac{1}{4} = 0,25$ | 0 |
| Iris | GPU | `gpu` | 28 | `skylake` | 1.0 | $\frac{1}{27}$ | 50 |
| Iris | Large-Mem | `bigmem` | 112 | `skylake` | 1.0 | $\frac{1}{27}$ | 0 |
| Aion | Regular | `batch` | 128 | `epyc` | 0,57 | $\frac{1}{1.75}$ | 0 |

- Continuous use of **2 regular skylake nodes** (56 cores, 224GB Memory) on `iris` cluster
  - ↪ 28 cores per node, 4 GigaByte RAM per core i.e., 112GB per node
  - ↪ **For 30 days**: 2 nodes$\times[\alpha_{cpu} \times 28 + \alpha_{mem} \times 4 \times 28 + \alpha_{gpu} \times 0]\times$ 30 days$\times$ 24 hours
    - ✓ Total: $2 \times [(1.0 + \frac{1}{4} \times 4) \times 28] \times 720 = 80640$ SU = **2419,2€ VAT excluded**

# Fairshare Factor and Job billing

## Number of SU associated to a job

$$N_{\text{Nodes}} \times [\alpha_{cpu} \times N_{\text{cores}} + \alpha_{mem} \times Mem + \alpha_{gpu} \times N_{\text{gpus}}] \times T_{\text{exec}}$$

| Cluster | Node Type | Partition | #Cores/node | CPU | $\alpha_{\text{cpu}}$ | $\alpha_{\text{mem}}$ | $\alpha_{\text{GPU}}$ |
|---|---|---|---|---|---|---|---|
| Iris, Aion | Regular | `interactive` | 28/128 | n/a | 0 | 0 | 0 |
| Iris | Regular | `batch` | 28 | `broadwell` | 1.0* | $\frac{1}{4} = 0,25$ | 0 |
| Iris | Regular | `batch` | 28 | `skylake` | 1.0 | $\frac{1}{4} = 0,25$ | 0 |
| Iris | GPU | `gpu` | 28 | `skylake` | 1.0 | $\frac{1}{27}$ | 50 |
| Iris | Large-Mem | `bigmem` | 112 | `skylake` | 1.0 | $\frac{1}{27}$ | 0 |
| Aion | Regular | `batch` | 128 | `epyc` | 0,57 | $\frac{1}{1.75}$ | 0 |

- Continuous use of **2 regular epyc nodes** (256 cores, 448GB Memory) on `aion` cluster
  - ↪ 128 cores per node, 1,75 GigaByte RAM per core i.e., 224 GB per node
  - ↪ **For 30 days**: 2 nodes$\times[\alpha_{cpu} \times 128 + \alpha_{mem} \times 1.75 \times 128 + \alpha_{gpu} \times 0]\times$ 30 days$\times$ 24 hours
    - ✓ Total: $2 \times [(0.57 + \frac{1}{1.75} \times 1.75) \times 128] \times 720 = 289382{,}4$ SU = **8681,47€ VAT excluded**

# Fairshare Factor and Job billing

## Number of SU associated to a job

$$N_{\text{Nodes}} \times [\alpha_{cpu} \times N_{\text{cores}} + \alpha_{mem} \times Mem + \alpha_{gpu} \times N_{\text{gpus}}] \times T_{\text{exec}}$$

| Cluster | Node Type | Partition | #Cores/node | CPU | $\alpha_{\text{cpu}}$ | $\alpha_{\text{mem}}$ | $\alpha_{\text{GPU}}$ |
|---|---|---|---|---|---|---|---|
| Iris, Aion | Regular | `interactive` | 28/128 | n/a | 0 | 0 | 0 |
| Iris | Regular | `batch` | 28 | `broadwell` | 1.0* | $\frac{1}{4} = 0,25$ | 0 |
| Iris | Regular | `batch` | 28 | `skylake` | 1.0 | $\frac{1}{4} = 0,25$ | 0 |
| Iris | GPU | `gpu` | 28 | `skylake` | 1.0 | $\frac{1}{27}$ | 50 |
| Iris | Large-Mem | `bigmem` | 112 | `skylake` | 1.0 | $\frac{1}{27}$ | 0 |
| Aion | Regular | `batch` | 128 | `epyc` | 0,57 | $\frac{1}{1.75}$ | 0 |

- Continuous use of **1 GPU nodes** (28 cores, 4 GPUs, 756GB Memory) on `iris` cluster
  - ↪ 28 cores per node, 4 GPUs per nodes, 27 GigaByte RAM per core, 756 GB per node
  - ↪ **For 30 days**: 1 node$\times[\alpha_{cpu} \times 28 + \alpha_{mem} \times 27 \times 28 + \alpha_{gpu} \times 4$ GPUS$]\times$ 30 days$\times$ 24 hours
    - ✓ Total: $1 \times [(1.0 + \frac{1}{27} \times 27) \times 28 + 50.0 \times 4] \times 720 = 184320$ SU $=$ **5529,6€ VAT excluded**

# Fairshare Factor and Job billing

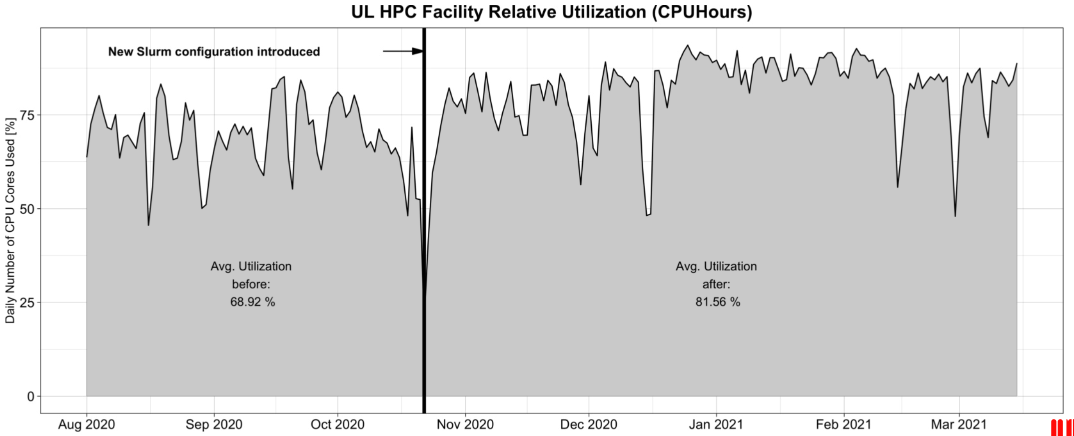
## Number of SU associated to a job

$$N_{\text{Nodes}} \times [\alpha_{cpu} \times N_{\text{cores}} + \alpha_{mem} \times Mem + \alpha_{gpu} \times N_{\text{gpus}}] \times T_{\text{exec}}$$

| Cluster | Node Type | Partition | #Cores/node | CPU | $\alpha_{\text{cpu}}$ | $\alpha_{\text{mem}}$ | $\alpha_{\text{GPU}}$ |
|---|---|---|---|---|---|---|---|
| Iris, Aion | Regular | `interactive` | 28/128 | n/a | 0 | 0 | 0 |
| Iris | Regular | `batch` | 28 | `broadwell` | 1.0* | $\frac{1}{4} = 0,25$ | 0 |
| Iris | Regular | `batch` | 28 | `skylake` | 1.0 | $\frac{1}{4} = 0,25$ | 0 |
| Iris | GPU | `gpu` | 28 | `skylake` | 1.0 | $\frac{1}{27}$ | 50 |
| Iris | Large-Mem | `bigmem` | 112 | `skylake` | 1.0 | $\frac{1}{27}$ | 0 |
| Aion | Regular | `batch` | 128 | `epyc` | 0,57 | $\frac{1}{1.75}$ | 0 |

- Continuous use of **1 Large-Memory nodes** (112 cores, 3024GB Memory) on `iris` cluster
  - ↪ 112 cores per node, 27 GigaByte RAM per core i.e. 3024 GB per node
  - ↪ **For 30 days**: 1 node$\times[\alpha_{cpu} \times 112 + \alpha_{mem} \times 27 \times 112 + \alpha_{gpu} \times 0]\times$ 30 days$\times$ 24 hours
    - ✓ Total: $1 \times [(1.0 + \frac{1}{27} \times 27) \times 112] \times 720 = 161280$ SU = **4838,4€ VAT excluded**

# Impact on the ULHPC total utilization



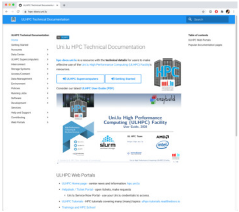
# Questions?

**Dr. Sebastien Varrette**
**Dr. Frederic Pinel**
**Dr. Emmanuel Kieffer**
**Dr. Grégoire Danoy**

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

## High Performance Computing @ Uni.lu

www: hpc[-docs].uni.lu



ulhpc-tutorials.rtfd.io