Management of an Academic HPC & Research Computing Facility: The ULHPC Experience 2.0

SEBASTIEN VARRETTE[*], HYACINTHE CARTIAUX, SARAH PETER[†], EMMANUEL KIEFFER, TEDDY VALETTE, and ABATCHA OLLOH, Faculty of Science, Technology and Medicine (FSTM) University of Luxembourg, Luxembourg

With the advent of the technological revolution and the digital transformation that made all scientific disciplines becoming computational, the need for High Performance Computing (HPC) has become and a strategic and critical asset to leverage new research and business in all domains requiring computing and storage performance. Since 2007, the University of Luxembourg operates a large academic HPC facility which remains the reference implementation within the country. This paper provides a general description of the current platform implementation as well as its operational management choices which have been adapted to the integration of a new liquid-cooled supercomputer, named *Aion*, released in 2021. The administration of a HPC facility to provide state-of-art computing systems, storage and software is indeed a complex and dynamic enterprise with the soul purpose to offer an enhanced user experience for intensive research computing and large-scale analytic workflows. Most design choices and feedback described in this work have been motivated by several years of experience in addressing in a flexible and convenient way the heterogeneous needs inherent to an academic environment towards research excellence. The different layers and stacks used within the operated facilities are reviewed, in particular with regards the user software management, or the adaptation of the Slurm Resource and Job Management System (RJMS) configuration with novel incentives mechanisms. In practice, the described and implemented environment brought concrete and measurable improvements with regards the platform utilization (+12,64%), jobs efficiency (average Wall-time Request Accuracy improved by 110,81%), the management and funding (increased by 10%). Thorough performance evaluation of the facility is also presented in this paper through reference benchmarks such as HPL, HPCG, Graph500, IOR or IO500. It reveals sustainable and scalable performance comparable to the most powerful supercomputers in the world, including for energy-efficient metrics (for instance, 5,19 GFlops/W (resp. 6,14 MTEPS/W) were demonstrated for full HPL (resp. Graph500) runs across all *Aion* nodes).

CCS Concepts: • **Computer systems organization → Architectures**; *Dependable and fault-tolerant systems and networks*; • **Networks**; • **Software and its engineering** → Software creation and management;

ACM Reference Format:
Sebastien Varrette, Hyacinthe Cartiaux, Sarah Peter, Emmanuel Kieffer, Teddy Valette, and Abatcha Olloh. 2022. Management of an Academic HPC & Research Computing Facility: The ULHPC Experience 2.0 . In 6^{th} *High Performance Computing and Cluster Technologies Conference (HPCCT 2022), July 08-10, 2022, Fuzhou, China.* ACM, New York, NY, USA, 15 pages.

1 INTRODUCTION

After 30 years of application in climate research, numerical weather prediction, particle simulation, astrophysics, earth sciences and chemistry, High Performance Computing (HPC) is now a cornerstone of all scientific fields

[*]Main contributor of this article, leading the developments of the research computing facility and HPC operations with the other co-authors. Other contributors are acknowledged at the end of the paper.
[†]Affiliated to the Luxembourg Centre for Systems Biomedicine (LCSB) within the University of Luxembourg.

HPCCT 2022, July 08-10, 2022, Fuzhou, China

ACM ISBN 978-1-4503-9664-6.

and widely recognised as a strategic tool for competitive science. For this reason, the last decade has seen massive investments in large-scale HPC and storage systems, aiming at meeting the surging demand for processing and data-analytic capabilities. This applies to R&D research centers and academic sites such as the University of Luxembourg (UL), which operates since 2007 a large research computing facility which remains a reference implementation within the country. This was made possible by an ambitious funding strategy since the University has invested tens of millions of euros into its HPC and Research Computing facility referred to hereafter as *ULHPC*, enabling researchers to push back the frontiers of traditional computing.

The ULHPC initiative now serves a wide user base, ranging from University staff and students to research partners and commercial users. They are given the possibility to run compute- and storage-intensive computations as part of their research or training, as illustrated in Figure 1. The University also offers access to its HPC facilities to the scientific staff of national public organizations, as well as to partners in joint research projects. Finally, dedicated service agreements established with local economic actors (*i.e.*, industry and external partners) allow access to the ULHPC resources, which includes supercomputers as well as expert consultants. Behind the scenes, the management of an HPC facility, its state-of-art computing systems, storage and software, is a complex enterprise and a constant area for discussions and improvements.

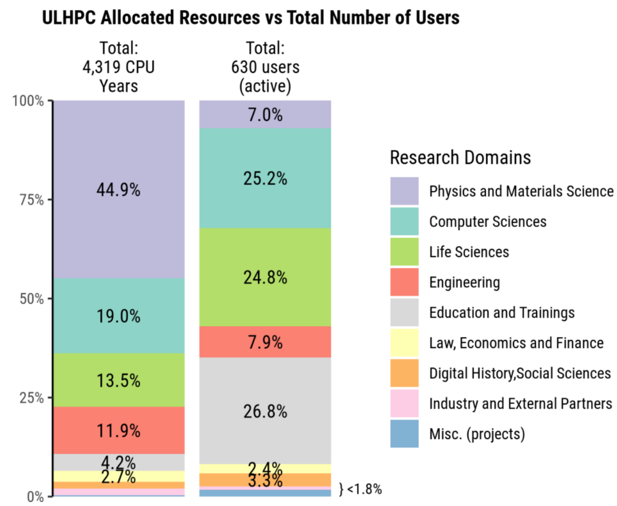

Fig. 1. User shares & cumulative usage per research domain.

To provide the best end-user experience while ensuring the efficient usage of the computing and storage resources, the dedicated team of operations personnel, system administrators and service manager need to design innovative solutions to consolidate the existing workflows submitted on the platform. It also includes adaptation to the emergence and sophistication of novel computing paradigms as well as new regulatory frameworks (such as the General Data Protection Regulation (GDPR) in Europe). These duties were assigned to a growing, yet relatively small, team of HPC experts. Due to the complexity of managing such a crucial infrastructure, a collaborative workflow, largely based on IT automation frameworks, was introduced in 2014 [28]. However, the described ecosystem was tied to computing clusters which have now been decommissioned. The acquisition of a new liquid-cooled supercomputer *Aion* in 2020 (with production release in 2021), to be federated with our previous flagship cluster *Iris* (in production since 2017), was the occasion to review the full stack of implemented approaches and procedures facilitating the operational administration of the infrastructure. This revision came with several changes, detailed in this state-of-practice article. Most design choices are motivated by several years of experience in addressing the heterogeneous needs inherent to an academic environment in a flexible and convenient way. This paper is organized as follows: Section 2 offers an overview of the hosting site and of the managed facility, as well as details of the implemented architectures (*e.g.* network, storage...) and surrounding services sustaining such a large-scale infrastructure. It covers also performance evaluation aspects. Section 3 reviews the management of the user software environment allowing automatic and optimized application builds exposed through the Lmod environment. Then the novel RJMS configuration applied on the facility (thus for Slurm [34] in our case), will be presented in Section 4. The introduced changes improve a seminal configuration in production on the *Iris* cluster, in an effort to offer a more flexible and easily understandable interface when federating both supercomputers with a uniform and transparent configuration. Finally, Section 5 concludes this article and provides some future directions and perspectives at the EuroHPC horizon.

2 OVERVIEW OF THE MANAGED FACILITY

Cluster	Date	Vendor	#Nodes	#Cores	#GPUs	R_{peak} [PFlops]
Aion	(2021-)	Atos	318*	40,704*	0	1.69*
Iris	(2017-)	Dell	196	5,824	96	1.07
Total in production (2022*)			**514***	**46,528***	**96**	**2.76 PFlops***
Gaia	(2011-2019)	Atos,Dell,HP	273	3,440	50	0.150
Chaos	(2007-2019)	Dell,HP	81	1,120	0	0.015

Table 1. Overview of the operated clusters and supercomputers.
∗: by 2023, 36 computing nodes will be added to Aion*, allowing it to reach a peak computing capacity of 1.89 PFlops (thus over 354 nodes totalling 45312 cores).*

Composed by several clusters of compute nodes detailed in the Table 1, the ULHPC platform has kept growing over time thanks to the continuous efforts of its core management team. While the *Chaos* and *Gaia* HPC clusters have been in production since 2007 and 2011 respectively, and although benefiting from several hardware upgrades over time, the obsolescence of equipment combined with the inability to further expand the support of the associated high-performance storage systems (hosting several critical data produced by the University's researchers and partners) and network components required the decommissioning of these computing platforms. The migration of data and computational campaigns toward the flagship production cluster *Iris* was a complex operation finalized in a step-wise approach in 2019. To sustain the load induced by this transition, it was essential to compensate for the loss of computing capacity induced by the decommissioning process. For this reason, the procurement and commissioning of a new leading-edge liquid-cooled supercomputer was initiated in 2020. Named *Aion*, the delivery and installation of this new supercomputer was largely impacted by the COVID pandemic and has seen its release for general availability on October 2021. In terms of hosting site and data center location, the research computing equipment are hosted since 2017 in the premises of the University's *Centre de Calcul* (CDC), which implements the cutting-edge approaches for data-center management, as well as the capacity to host state-of-the-art liquid-cooled HPC systems. This follows the trends and design choices implemented within all the recent HPC data-centers hosting the EuroHPC systems, or more generally all the biggest supercomputers listed in the Top500 around the world. In all cases, the CDC is established over two floors and features per level a global surface of around $1000m^2$ and five server rooms (totalling $518m^2$), each attached with a dedicated technical room. The first level is hosting administrative IT and research equipment, and the second floor is primarily dedicated to the HPC equipment (compute, storage and interconnect) and provided with a power supply of 3 MW where each IT room and racks have a dual power supply of type 2N. The power generation station supplies the whole CDC for 4.5MW of electrical power, and 4.5MW of cold water at a 12-18°C regime which are used for airflow cooled rooms hosting traditional HPC systems. Anticipating future HPC-cooling technologies, a separate hot water circuit (between 30 and 40°C) was planned from the early stage of the CDC design to meet the needs of liquid-cooled solutions and deployed in two dedicated server rooms which are room-neutrals. For a long time, these rooms were expected to host the national HPC facility. As a different hosting site was finally selected, *Aion* is the first and most recent DLC-enabled supercomputer installed in those premises and benefiting from this energy-efficient cooling infrastructure, which could sustain any similar expansion without any problem. With regards to the fire extinguishing system, each IT room and each electrical room is protected by extinguishing stations relying on Argonite gas batteries which are coupled to allow the generation of multiple extinguishments. The managed HPC ecosystem includes not only the computing nodes, but also a set of servers (eventually virtualized over the Kernel-based Virtual Machine (KVM) hypervisor) as well as networking equipment (*i.e.*, Ethernet on InfiniBand switches). In total, 756 servers are administered as of April 2022. All systems (except the network equipment) are running a Redhat-based Operating System (OS) *i.e.*, either RHEL or CentOS. A complex configuration management framwork based on Puppet [7] and Ansible (the latter implemented through the BlueBanquise [1] stack) has been developped to deploy and manage in an automatic and consistent way the operated systems. The general organization of all ULHPC supercomputers is kept consistent with best-practices to ensure a secure and redundant setup.

2.1 Network Organisation

HPC encompasses advanced computation over parallel processing, enabling faster execution of highly compute-intensive tasks which heavily rely on interconnect performance.

For this reason, the main high-bandwidth low-latency network of the ULHPC facility relies on the dominant interconnect technology in the HPC market *i.e., InfiniBand (IB)*, more specifically in the latest HDR (High Data Rate – 200Gbps) and EDR (Enhanced Data Rate – 100Gbps) flavors. There are several topologies commonly used in large-scale HPC deployments (*i.e.*, Fat tree, Hypercube, Torus or Dragonfly) [12], yet Fat-tree was always promoted on all ULHPC clusters due to its versatility, high bisection bandwidth and well understood routing which remains very efficient at avoiding superposition of routes on the same link for all to all or many to many communication patterns. It is also the only topology allowing for a non-blocking network at large-scale. In practice, a two-layer approach was sufficient and exhibits (1) a set of *leaf* IB switches labelled L1 LIB and (2) a set of spine/super-spine IB switches *e.g.,* L2 SIB. In its seminal installation, the *Iris* cluster IB interconnect was relying on a *non-blocking 1:1 Fat-Tree topology*, used to link all compute nodes with one link per node, the management servers and the high-performance storage systems hosting the SpectrumScale/GPFS and Lustre File Systems (FSs) (see section 2.2).

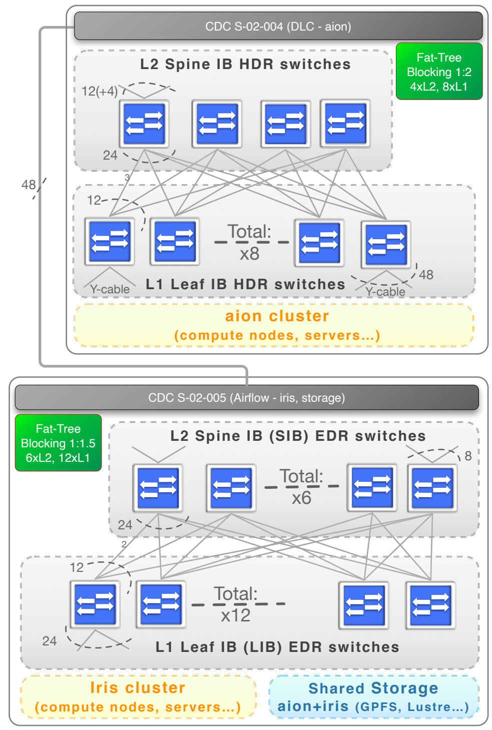

Fig. 2. Overview of the high-bandwidth low-latency IB network topology [29].

With the integration of *Aion*, an adaptation of the IB network was required and is detailed in [29], leading to the topology depicted in the Figure 2. To minimize the number of switches per nodes while keeping a good bisection bandwidth and minimizing cabling changes, a blocking configuration was set by freeing one of the L1 to L2 links available on *Iris*, to connect with 48 cables the L2 switches of *Aion*. Overall, this approach allowed to increase the leaf connection capacity from 216 to $12 \times 24 + 8 \times 48 = 672$ end-points (**+311%**). This changed the blocking factor for *Iris* from full non-blocking to 1:1.5. On *Aion*, the Fat-tree configuration was set with a blocking factor 2:1 due to the usage of specific "splitted" cables (also called "Y-cables"). The induced bandwidth penalty (100 Gb/s instead of 200, thus aligned to *Iris* capacities) was considered affordable as nowadays, very few applications are really able to fully exploit 200 Gb/s networks. Indeed, the merged IB network has brought very marginal performance penalties. For instance **less than 3% (resp. 0.3%) Read (resp. Write) bandwidth degradation** were observed when evaluating the impact on the parallel I/O GPFS performance of the shared storage infrastructure detailed in the section 2.2. The IB network configuration was first validated with the MPI Bisectional Bandwidth (BB) benchmark widely used to provide an evaluation of a topology's performance [18]. This measures the IB bandwidth between pairs of nodes. Considering the theoretical effective throughput of the implemented network at the compute nodes level (100 Gb/s), unidirectional (resp. bidirectional) point-to-point bandwidth evaluations are expected to reach 11,64 GiB/s (resp. 23,28 GiB/s). The measured performance is reported in the Figure 3, demonstrating stable and sustainable results for all possible pairs of nodes **with a point-to-point bandwidth efficiency above 95.45%** [29].

Of course, having a single high-bandwidth and low-latency network to support efficient HPC and Big Data workloads would not provide the necessary flexibility brought by the Ethernet protocol. For this reason, an

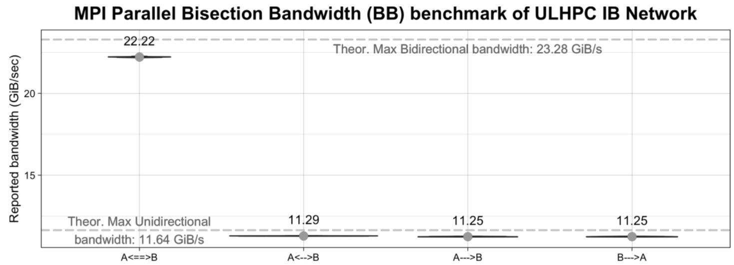

Fig. 3. MPI Bisectional Bandwidth (BB) IB performance between *Aion* compute nodes.

additional Ethernet-based network is defined for management tasks, external access and user's applications inside the research computing system in such cases. The different flows and streams are separated inside dedicated Virtual Local Area Network (VLAN) [29]. In particular, applications and user-level data transfer that do not support Infiniband natively can rely on either the non-routed IP-over-IB emulation layer (100 GbE) inside each cluster, or a 10 to 40 GbE "`prod`" VLAN. External access is protected within a 10 to 40 GbE "`DMZ`" VLAN when *management* tasks are isolated within a 1GbE "`mgmt`" VLAN. In practice, the Ethernet network has been reorganized as a 2-layer topology detailed in [29]: one upper level (*Gateway Layer*) with routing, switching features, network isolation and filtering (ACL) rules and meant to aggregate *only* switches. The bottom level (*Switching Layer*) is composed of *core* switches as well as the *TOR* (Top-of-rack) network equipments, meant to interface the HPC servers and compute nodes. The TOR switches are typically 1GbE switches with redundant 10GbE uplinks, possibly stacked, connecting all out-of-band interfaces for hardware management. The core switches are 10GbE switches with redundant 40GbE uplinks, stacked or clustered using Cisco vPC technology (Virtual Port Channel). This new topology aimed at tackling the limitations met with our previous HPC developments over the decommissioned clusters, *i.e.*, (1) enhanced service *availability* using fault tolerance techniques (critical network equipment are fully redundant; critical servers are connected using link aggregations etc.); (2) improved *maintainability*. For instance, it is easy to apply firmware and security updates on the switches, without requiring a service interruption or a maintenance window and (3) *scalability*: additional clusters or racks of computing equipment can be added in the coming years, without requiring any major topology change or physical cabling.

2.2 Tiered Shared Storage infrastructure

Due to their huge number of compute nodes, the largest supercomputers all deploy such parallel file systems for their external shared storage solution. The ULHPC facility relies on two types of distributed and parallel FS to deliver high-performance storage at a Big Data scale:

File System	Vendor	#Disks	Raw/Effective capacity
GPFS (2017-)	DDN	710 HDDs + 38 SSDs	4260 / 3408 TB
Lustre (2018-)	DDN	Object Storage Targets: 167 HDDs Meta-Data Targets: 19 SSDs	1300 / 920 TB
OneFS (2014-)	Dell/EMC	n/a (NDA)	7100 / 6400 TB

Table 2. Overview of the main ULHPC storage systems.

(1) **IBM Spectrum Scale**, formerly known as the General Parallel File System (GPFS) [27], a global high-performance clustered file system hosting home directories and projects data;
(2) **Lustre** [13], an open-source, parallel file system dedicated to large, local, parallel *scratch* storage.

These two FS remain the reference solutions deployed in large-scale HPC infrastructures – for instance, no other FS was ever present among the first 100 systems listed in the Top500 since the biannual release of the list. The decision to migrate to GPFS within the ULHPC facility was done in October 2014 to bypass the performance and scalability issues experimented with the initial NFS-based setup. The hereby described GPFS system, based on a DDN solution, was deployed in 2017 together with the release of the Iris cluster, and was extended in 2021. Lustre was present from the early developments of the facility (*e.g.*, since 2011) yet was never considered stable

	Directory	File System	Backup	Default **Quota**	Default **Inode quota**	**Purging time**
`$HOME`	`/home/users/<login>`	GPFS/Spectrumscale	yes (daily)	500 GB	1 M	-
	`/work/projects/<name>`	GPFS/Spectrumscale	yes (daily)	n/a	0	-
`$SCRATCH`	`/scratch/users/<login>`	Lustre	no	10 TB	1 M	60 days
	`/mnt/isilon/projects/<name>`	OneFS	yes (snapshot, weekly)	1.14 PB globally	-	-

Table 3. Overview of the ULHPC File-Systems backup and quota policy.

enough at that time to be used as storage backend for anything except what this FS was initially designed for *i.e.* temporary scratch I/O data. This kind of consideration no longer holds and more and more supercomputing systems rely exclusively on a Lustre-based FS. In all cases, the current Lustre storage system, also based on a DDN solution, was deployed in 2018. In addition, the ULHPC storage infrastructure relies on **OneFS**, a global low-performance Dell/EMC Isilon solution used to host project data, and serve for backup and archival purposes. Table 2 reports the characteristics of the three available storage systems. Then, each server and computational resources have access to these file systems, with different levels of performance, permanence and available space and quotas (including for inodes *i.e.,* number of files) as summarized in Table 3.

The performance evaluation of the two distributed and parallel FS available on the ULHPC facility is regularly assessed through IOR [5], the reference parallel IO benchmark that can be used to measure I/O throughput using various interfaces and access patterns subjected to a synthetic workload.

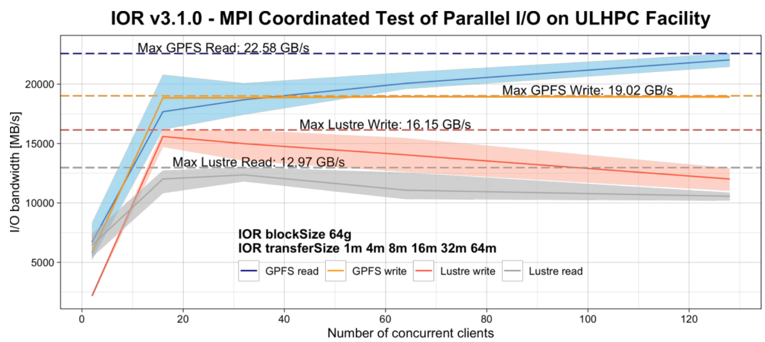

Fig. 4. IOR [5] performance evaluation of ULHPC parallel FS.

The latest results are summarised in the Figure 4, demonstrating over an increasing number of concurrent and distributed clients (1 MPI process per socket) **stable and scalable performance for GPFS (Max read: 22.58 GB/s, Max write: 19.02 GB/s)**, and sustained performance for Lustre which exhibits better write (16.16 GB/s) than read (12.97 GB/s) capabilities – a characteristic which was present from the seminal deployment. These results are to be compared to the theoretical I/O performance obtained from local SSD disks (between 300 and 400MB/s) and past evaluations done on NFS (below 100 MB/s with 64 clients) prior to the migration to GPFS, none of them allowing to reach the storage capacities featured in the Table 2.

With such storage capacities, novel challenges appear with, on one side the emerging paradigm of Open Science enabling an easier access to expert knowledge and material, and on the other hand the necessary compliance to the EU's General Data Protection Regulation (GDPR) [14]. We have carefully studied in [23] the interactions occurring during data processing on our facilities, and we were able to pinpoint, from a legal and technical point of view, the major data protection issues arising during HPC workflows. Possible solutions are out of the scope of the present article, but are also suggested in [23]. Furthermore, an organization-wide risk management analysis dedicated to the characterization of the compliance to both the GDPR and FAIR (*Findable, Accessible, Interoperable and Reusable* [33]) principles were promoted. It was conducted following the US National Institute of Standards and Technology (NIST) Guide for conducting Risk Assessments (NIST 800–30 Rev. 1) [26] grouped according to ISO/IEC 27002 [17]. One of the major challenges which is only partially addressed in [23] is related to the security enforcement in accordance with Art. 32 of the GDPR. It corresponds to the complex tracking of data movements within supercomputing facilities. Technically speaking, parallel and distributed file-systems used in HPC environments as the ULHPC are indeed not yet fully able to account and log internal data movements: changelogs-based auditing capabilities relevant for the GDPR compliance are featured in recently released versions of Lustre (2.11) and GPFS/Spectrumscale (5.0). Despite the witnessed stability and robustness of the latter, it is worth to report that the licensing model proposed by IBM evolved from the precedent versions, *i.e.,* from socket-based to a cost model based on the effective storage capacity. This renders the migration from v4 extremely expensive and prohibitive for HPC centres with large-scale storage capabilities aiming at enabling the changelogs-based auditing expected from the GDPR compliance. Without a more reasonable pricing model, it is anticipated that the design choices made at the ULHPC would be different when implementing a new site to favor a pure Lustre-based solution to make significant savings. This approach was already applied in the recent deployment of new EuroHPC systems across Europe, such as MeluXina in Luxembourg.

2.3 Computing Performance Evaluation and Acceptance Tests

The computing performance of the ULHPC supercomputers are continuously evaluated through a strict benchmarking campaign involving a set of synthetic as well as application-oriented benchmarks, each highlighting different aspects of the facility.

This includes the following benchmarks: **HPL** [24], a portable implementation of the High-Performance Linpack benchmark for distributed-memory computers which is used as reference benchmark to rank supercomputers in the Top500 list; **STREAM** [21], a simple synthetic benchmark program that measures sustainable memory bandwidth (in GB/s) and the corresponding computation rate for simple vector kernel; **HPCG** [3], the High Performance Conjugate Gradient benchmark, intended to complement HPL and designed to exercise computational and data access patterns that more closely match a different and broad set of important applications outside the ones caught by HPL patterns of execution, memory access, and global communication; Bisection Bandwidth (**BB**) test and **OSU Microbenchmarks (OMB)** [6] for low level network performance measures; **Graph'500** [8], a benchmark suite directed towards graph processing, which is a core part of most analytics workloads, and is thus well suited to reflect the performance achievable by data-intensive applications. Furthermore, storage-oriented benchmarks such as **IOR** (already presented in the section 2.2), or **IO500** [4] were executed. The latter, promoted by the Virtual Institute for I/O, also relies on IOR to evaluate workloads matching well-optimised I/O patterns and random I/O (`IOEasy`, `IOHard`) and metadata tests (`MDEasy`, `MDHard`). In addition to the above-mentioned synthetic benchmarks, supercomputers are also qualified against the application codes provided by the Unified European Application Benchmark Suite (UEABS) [9]. At this level, a current work in progress validates the user software applications performance built as part of the RESIF framework presented in the section 3 through *ReFrame* [19], a high-level framework for writing regression tests for HPC systems. In all cases, prior to large-scale runs, it is crucial to validate the single-node performance to detect potentially failing hardware components. For instance, HPL can easily track processors with degraded performance. The expected distribution of effective single-node performance against HPL is illustrated for the

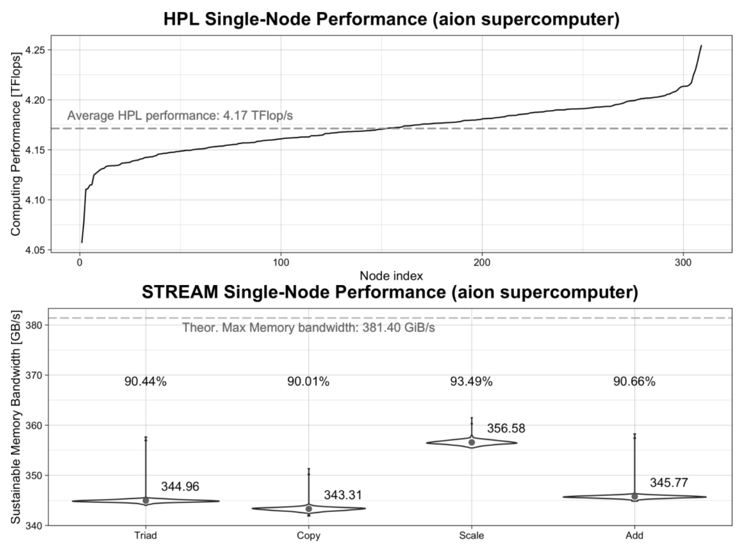

Fig. 5. Sorted distribution of HPL single node performance across *Aion*. Below: STREAM node performance distribution mesures within *Aion* nodes.

AMD Epyc processors featured on *Aion* computing nodes on the Figure 5 (top). The measured $R_{\max}$ values stand within acceptable range (±2.8% for all nodes) demonstrating "healthy" processors on those Dual-CPU nodes with consistent performances (see below side table). Similarly, STREAM allows to detect failing Memory DIMMs. Figure 5 (bottom) reports the measured sustainable memory bandwidth for each of the four computational kernel functions that compose STREAM when run across the *Aion* nodes: *Triad* (the most complex one), *Copy*, *Scale* and *Add*. As the maximum memory bandwidth supported by the AMD EPYC 7H12 processors is 190.7 GiB/s, it is possible to compute the efficiency of the benchmarks for these bi-sockets nodes, which is also reported. The obtained values are again demonstrating an excellent efficiency and healthy memory DIMMs. Finally, the sanity and performance of the individual Host Channel Adapter (HCA) IB network cards of each node can be assessed together with the sanity of the IB topology cabling through the BB benchmark as depicted in the Figure 3.

Processor/GPU Model	#Cores	Freq.	R_{peak}	Avg. R_{max}
AMD ROME 7H12 *(epyc)*	64	2.6 GHz	2.66 TFlops	2.09 TFlops
Intel Xeon E5-2680v4 *(broadwell)*	14	2.4 GHz	0.54 TFlops	0.46 TFlops
Intel Xeon Gold 6132 *(skylake)*	14	2.3 GHz*	1.03 TFlops	0.94 TFlops
Intel Xeon Platinum 8180M *(skylake)*	28	2.3 GHz*	2.06 TFlops	1.76 TFlops
NVidia Tesla V100-SXM2	5120+*640*	1.3 GHz	7.80 TFlops	5.59 TFlops

*: AVX-512 Turbo Frequency

Once computing processor capabilities are evaluated (a summary is proposed in the side table) and that single-node performance is assessed, full cluster runs can be foreseen. This requires complex tuning to optimize the benchmark parameters, a process out of the scope of this article and largely debated and reported in the litterature. Table 4 simply depicts the best results obtained for this evaluation campaign for the most important benchmarks, together with the corresponding worldwide rank from the latest list releases available at the time of writing. A few take-away lessons can yet be expressed. With regards HPL CPU efficiency, the obtained values are consistent with the expectations (≥ 72% efficiency) even for large-scale runs. Yet the R_{peak} performance for the Intel skylake Gold processors takes into account the fact that those CPU embed two AVX512 units, thus they are capable of performing 32 Double Precision operations per cycle but *only* upon AVX-512 Turbo Frequency (*i.e.*, the maximum all-core frequency in turbo mode) in place of the base non-AVX core frequency generally advertized. With regards the HPL performance over the Nvidia GPU accelerators, the depicted results were obtained through private Nvidia binaries optimized for the V100 cards that could not be redistributed. We indeed obtained very poor results (below 13% efficiency on 1 node/4 GPU cards) when relying on the public CUDA-enabled HPL code. Nevertheless, even when using the optimized binaries, we could not reach the expected efficiency (above 65%) known for these cards for runs exceeding 8 nodes (32 GPU cards). We have not been able to track the origin of this problem. Finally, with regards the Graph500 benchmark and its two flavors (BFS and SSSP), scalable performance for the second problem resolution (Shortest Path SSSP) cannot be obtained with the reference open-source code and requires custom developments we did not afford for the moment.

	Benchmark	#N	*(Main parameters)*	Best Performance	Efficiency	Improvement*	Equivalent Worldwide Rank	
Aion	**HPL** (Top500)	**318**	*(NB=192,P×Q=48×53)*	$R_{\max}$ = **1255.36 TFlops**	**74.10%**	+1.9%	>500 (Nov 2021)	**#490** (Jun 2020)
	Green500	318		5.19 GFlops/W		+12.83%	**#71** (Jun 2022)	**#56** (Jun 2021)
	HPCG	318		16.842 TFlops		+15.35%	**#144** (Nov 2021)	**#135** (Jun 2021)
	Graph500 BFS	2^8=256	*(Scale: 36,Edge:16)*	975 GTEPS		+64%	**#31** (Jun 2022)	**#23** (Jun 2021)
	GreenGraph500	2^8=256		6.14 MTEPS/W		+180%	**#43** (Jun 2022)	**#36** (Jun 2021)
				*: *performance improvement with the minimal acceptance threshold set in the* Aion *tender document*				
	IO500 (`isc21` release)	128		**11.345219**			**#42** (Nov 2020 - latest release)	
Iris	HPL *(CPU/broadwell)*	108		84.75 TFlops	72.98%			
	HPL *(GPU/V100 16G)*	72	*(NB=320,P×Q=12×6)*	283.6 TFlops	52.87%			
	HPCG *(GPU/V100 16G)*	72		8.74 TFlops				
	HPL *(GPU/V100 32G)*	24	*(NB=288,P×Q=6×4)*	135.2 TFlops	75.61%			
	HPCG *(GPU/V100 32G)*	24		2.90 TFlops				

Table 4. Overview of the global computing capacity performance for ULHPC supercomputers.

3 USER SOFTWARE ENVIRONMENT

The ULHPC facility provides a large set of pre-installed scientific applications covering various research domains. Users can navigate through and use the software of their choice using the standard environment module system Lmod [22]. To build and deploy the software stack, we have developed and been using using since 2014 an in-house tool called RESIF. In preparation for the supercomputer *Aion*, the RESIF framework received a significant update in 2020 with the release of RESIF 3.0 which is described in detail in [31].

In summary, RESIF is a tool designed for the automated deployment of scientific software on an HPC cluster, and is based on the EasyBuild installation framework [15] for the compilation and installation of the software and generation of the modules. RESIF pilots EasyBuild installations in order to apply a consistent set of configurations and customization on group of HPC clusters. More precisely, RESIF provides the following services in a unified framework: definition of thematic software bundles; organization of installation paths (release versioning, architecture, etc.); customization of software (source and configuration); global, cluster- and node-specific configurations; scripts and launchers for setup, deployment and testing; workflow for contributions to upstream EasyBuild; and documentation. For the ULHPC team, the use of RESIF reduced the manual operations needed for deployment of new software and consequently the risk of errors. Furthermore, RESIF 3.0 simplified our workflow to contribute our changes to the EasyBuild community, which allow us to remove about 90% of our custom software configurations. In practice, a new software stack is released every year with a new major `<version>` number, and is based on the toolchains released by EasyBuild twice a year: `foss` (Free and Open Source Software) and `intel` toolchains. For a given toolchain, all the major components (such as GCC, MPI, BLAS, Python, etc.) are fixed to a specific version as illustrated in Table 5. This toolchain is then used to build all the software on top of it. *ULHPC bundles*, defined in the RESIF configuration, are the software sets to be built for a given release and structured according to the layout depicted in Table 6. Each software is built and optimized for each supported processor architecture, namely `broadwell` and `skylake` for the *Iris* cluster, and `epyc` for *Aion*. For GPU nodes, the `ULHPC-gpu` bundle provides CUDA-enabled toolchains and additional GPU-accelerated builds of specific software as listed in Table 6. The `MODULEPATH` environment variable is automatically populated based on the computing node to provide the software in the correct processor architecture, and with the GPU-enabled software proposed as first choices on a GPU node. This operation, transparent for the users, allows them to benefit from the most optimized build for their software. Concerning the modules themselves, they are generated automatically by EasyBuild. On ULHPC, we configured RESIF to follow

	Software set release <version>		
	2019b	**2020b**	**2021b**
Component	legacy	prod	devel
binutils	2.32	2.35	2.37
GCCCore	8.3.0	10.2.0	11.2.0
foss	2019b	2020b	2021b
- OpenMPI	3.1.4	4.0.5	4.1.2
intel	2019b	2020b	2021a
- Compilers/MKL	2019.5.281	2020.1.217	2021.4.0
- Intel MPI	2018.5.288	2019.7.217	2021.4.0
Python	3.7.4	3.8.6	3.9.6
RESIF version	3.0	3.0	3.1
#Software Modules	<arch>: 269 gpu: 135	<arch>: 274 gpu: 151	<arch>: 282 gpu: 157

Table 5. ULHPC software set releases characteristics.

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

Table 6. Overview of ULHPC Bundles.

the *categorized naming scheme* (<category>/<app>/<version>-<toolchain><suffix>) that we implemented for our needs and contributed to EasyBuild upstream back in 2016. One peculiarity in our configuration is that the module utility is *only* available on the compute nodes, which is a simple way to prevent users from running applications from the login nodes inadvertently. sers can also take advantage of *Singularity* [20] to create a full environment tailored to their needs. Finally, a current work in progress is to combine the ULHPC software environment with the one provided within the EESSI project [2], an initiative fostering collaboration between European HPC sites and industry partners to set up a shared repository of scientific software installations suitable for a variety of systems, regardless of which flavor/version of Linux distribution or processor architecture is used.

4 USER JOB MANAGEMENT AND THE SLURM INFRASTRUCTURE

The ULHPC infrastructure relies on Slurm (Simple Linux Utility for Resource Management) [34] as RJMS for cluster/resource management and job scheduling. This middleware is responsible for allocating resources to users, providing a framework for starting, executing and monitoring work on allocated resources and scheduling work for future execution. Slurm provides superior scalability and performance. Multiple distributed libraries (*e.g.* Dask [25], IPyparallel [16]) offer the possibility to users to submit distributed jobs directly through their API – the compatibility between the workload manager and these tools makes the use of the platform more convenient for non-specialists. The seminal Slurm configuration put in place with *Iris* was deeply reviewed to prepare the federation with the *Aion* supercomputer while offering a simplified user experience. Details on the implemented policy and configuration changes are provided in [30] together with a thorough performance evaluation on the ULHPC supercomputer workloads. Below is an summary of the modifications performed. First of all, the partitions (detailed in Table 7) were set to match the 3 types of computing resources:

- batch is intended for running parallel scientific applications on "regular" nodes (Dual CPU, no accelerators, 128 to 256 GB of RAM);
- gpu is intended for running GPU-accelerated scientific applications on "gpu" nodes (Dual CPU, 4 Nvidia accelerators, 768 GB RAM);
- bigmem is dedicated for memory intensive data processing jobs on Large-memory nodes (Quad-CPU, no accelerators, 3072 GB RAM).

	Partition	#Nodes (core/node)	Default-Max Job Time	Max Nodes	Prio-rity
Aion	interactive (*floating*)	318	30min - 2h	2	100
Aion	batch (*default*)	318 (128c)	2h - 48h	64	1
Iris	interactive (*floating*)	196	30min - 2h	2	100
Iris	batch (*default*)	168 (28c)	2h - 48h	64	1
Iris	gpu	24 (28c)	2h - 48h	4	1
Iris	bigmem	4 (112c)	2h - 48h	1	1

Slurm QOS	(partition)	Priority	GrpTRES	MaxJobs PerUser	Max Wall
besteffort	(*)	1		50	
low	(*)	10		2	
normal	(*)	100		50	
long	(*)	100	node=6	4	14 days
debug	(interactive)	150	node=8	10	
high	(*)	200		50	
urgent	(*)	1000		100	

Table 7. Non-hidden Slurm partitions, QOS and their limits.

In addition, a *floating* partition named interactive was set. Intended for quick interactive jobs, it allows for quick tests and compilation/preparation work. This is the only partition crossing all type of nodes (thus floating), the selection of the expected resource type being left to the specification of the matching *feature*. Then, the Slurm QOS used to constrain or modify the characteristics of a submitted job were completely redefined with the introduction of *Aion*. Previously specific to each partition and named qos-<partition>, we now favor cross-partition QOSs, mainly tied to priority level (low → urgent). A special *preemptible* QOS exists for best-effort jobs, and the long QOS allows to run jobs for up to 14 days (instead of the default 2 days walltime limit). Further limits on Trackable RESources (TRES *i.e.,* a resource (cpu,node,etc.) tracked for usage or used to enforce limits against) are defined and summarized in Table 7 where GrpTRES stands for the total count of TRES able to be used at any given time from jobs running within the considered QOS. If this limit is reached, new jobs are queued but are only allowed to run once the resources have been relinquished from this group. MaxJobsPerUser is the maximum number of jobs a user can have running at

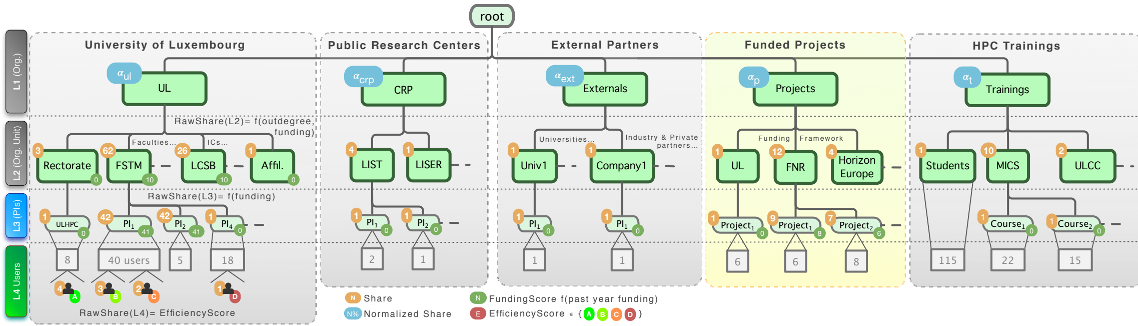

Fig. 6. New ULHPC Account Tree Hierarchy [30].

a given time and `MaxWall` corresponds to the maximum wall clock time any individual job can run for in the given QOS. The parameters of the backfilling scheduling enabled within the RJMS engine were also reviewed with updated resolution windows and a refreshed lock rate to optimize and favor interactive and/or small jobs.

Furthermore, the fair sharing configuration was completely reworked. Fairshare allows past resource utilization information to be taken into account into job feasibility and priority decisions to ensure a fair allocation of the computational resources between all ULHPC users. In practice, we moved from the "*Depth Oblivious*" algorithm in favor of "*Fair Tree*", which prioritizes users such that 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. One advantage is that new jobs are immediately assigned a priority and the fairshare levels are more easily understandable: all users from a higher priority account receive a higher fairshare factor than all users from a lower priority account and we made tremendous efforts to render the account hierarchy more consistent with transparent and innovative way of assigning the shares. Of course, the migration to this fair-sharing scheme renders the job priority resolution quite sensitive to the raw shares associated to each user account. For this reason, the associations and shares defined in the accounting database were deeply restructured and formalized with consistent rules designed to attribute these raw shares. More specifically, accounting records were re-organized as a hierarchical tree as depicted in Figure 6. End users, defined by their ULHPC login `<login>`, stand as leaves of the structure and different rules are applied to define the raw shares of an account depending on its level in this hierarchy:

- L_1 (Organization): arbitrary shares $\alpha_{<\text{org}>}$ to dedicate at least 85% of the platform to serve UL needs;
- L_2 (Organizational Unit): shares are function of the *out-degree* of the tree nodes representing the number of active research groups, together with a *funding score* reflecting the past year(s) budget contribution (normalized on a per-month basis) for the year in exercise;
- L_3 (Principal Investigator (PI), project or training course): shares are function of the same *funding score* (with eventually a different weight) reflecting the past year(s) budget contribution of the PI/project.
- *End user*: share levels are defined as a function of an user's *efficiency score*, giving incentives for more efficient usage of the platform.

The proposed scoring schemes are further discussed in [30] with examples, yet the general ideas are summarized below. The *funding score* associated with an account A belonging to a level L in the hierarchy is yearly updated based on past funding and a level threshold β_L: $\texttt{FundingScore}_L(A) = \left\lfloor \beta_L \frac{Investment_A(Year-1)}{\#months} \right\rfloor$. This funding score is added to the out-degree of an account at the $L2$ level of the hierarchy to define the raw share of the considered account A. At $L3$ level, it simply comes as an addition to the default raw share value (1), still to favor through the Fair-Tree algorithm accounts with past budget contributions. Including the out-degree within $L2$ accounts raw shares provides a consistent way to reflect the weight (in terms of research groups or projects using the platform) for the considered organisation units (faculty, interdisciplinary center etc.). At the end-user level, the *efficiency score* $S_{\text{efficiency}}$ is an integer (between 0 and 3) added to the default raw share value (1) as an incentive for each user to improve their efficiency, either in terms of CPU, GPU, Memory or Wall time estimation when interacting with the HPC facility. For the latter, we propose to use for a given user U the average *Wall-time Request Accuracy* (WRA), computed as a reduction over a pre-defined time range (1 year typically) for N *completed* jobs by U as follows: $S_{\text{efficiency}}(U, Year) = \text{WRA}(U, Year) = \frac{1}{N}\sum_{JobID}\frac{T_{\text{elapsed}}(JobID)}{T_{\text{asked}}(JobID)}$. Providing a meaningful efficiency metric to capture CPU, GPU and memory usage within allocated jobs is more complex to adjust as it would be unfair to privilege one component over the other - some user workflow are indeed either CPU-bound or memory-bound. The `seff` utility coming with Slurm provides the necessary inputs with regards CPU and memory usage associated to each jobs. With regards GPU usage, setting up GPU telemetry through NVIDIA Data Center GPU Manager (DCGM) [10] is required. One difficulty is to correctly define the DCGM group for the set of allocated GPUs within each job to be able to start collecting the performance metrics accordingly. This is done by adapting in Slurm the job `epilog` and `prolog` scripts.

All the many RJMS configuration and policy changes described in this section and in [30] were applied at once within the production systems on Oct 22, 2020 during a maintenance session. The impact was nearly immediate on the global system performance. For instance, the average utilization (daily number of CPU cores used) is depicted in Figure 7 and aggregates traces from several months of *uninterrupted* HPC services (*i.e.*, between two maintenance sessions) prior and after the introduction of these changes. It follows that the supercomputer relative utilization was **increased by 12.64%** to reach an average of **81.56% of daily utilization** after 6 months in production. Figure 8 reports the evolution of the average Wall-time Request Accuracy (WRA) metric over a 1 year period before and after the configuration changes. As can be seen, the **average WRA for the processed jobs was increased by 110,81%**, moving from 14,8% on average to 31.3%. More metrics are analysed in [30], together with the implemented billing policies judged out of scope for the present article. Yet the introduced funding score coupled with this job billing and accounting has already permitted to **increase the HPC budget incomes in 2021 by 10%**. Finally, all new research project proposals submitted to the national funding instrument are asked to budget the expected HPC computing expenses from the proposed cost model, demonstrating the validity of the approach endorsed by national agencies such as the "*Fond National de la Recherche*" (FNR) in Luxembourg. In all cases, this new setup is now in production for 18 months across all ULHPC supercomputers.

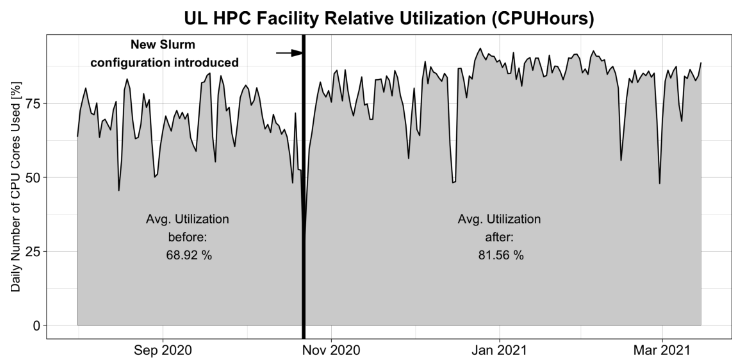

Fig. 7. Impact of the updated Slurm configuration on the ULHPC relative utilization (in CPUhours) restricted to the *Iris* supercomputer.

5 CONCLUSION AND PERSPECTIVES

This article reports on the design choices at the hardware and middleware levels, as well as the configuration changes introduced at the occasion of the acquisition of an novel leading-edge supercomputer, *Aion*. The objective was to allow for a smooth integration with the existing HPC ecosystem, as well as to simplify the experience of users across multiple research domains. First, the University's data center specifications were reviewed to comply with both traditional air-cooled systems and cutting-edge liquid-cooled solutions through a separate hot water circuit. This compliance is essential to guarantee the freedom to host all possible types of research

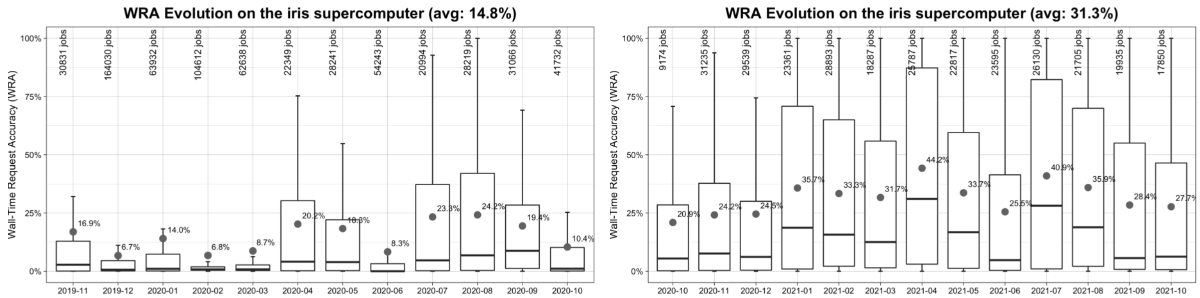

Fig. 8. Impact of the updated Slurm configuration on the average Wall-time Request Accuracy (WRA) for all jobs completed on the Iris supercomputer before (left) and after (right) the configuration changes.

computing equipment in the near future without sacrificing the energy efficiency of the hosting infrastructure. Then, the general organization of each ULHPC supercomputer was described, with a specific attention to the bi-level network topologies implemented for both the fast IB interconnect and the Ethernet network [29]. The tiered shared storage infrastructure was briefly introduced, revealing different types of distributed and parallel File Systems able to deliver high-throughput data storage capabilities at a Big Data scale. The backup, purging and quota policy was also reported. Furthermore, GDPR and FAIR principles compliance were discussed [23]. The performance of the ULHPC supercomputers reveals sustainable and leading-edge performance comparable to the most powerful supercomputers in the world, including for energy-efficient metrics. With regards to the software environment, a large and rich variety of scientific applications has been provided to the user community. A complex workflow relying on the novel RESIF 3.0 framework [31] allows for a yearly release of the complete software set relying on architecture-optimised builds of bundles exploiting multiple toolchains. Then, the changes operated to the RJMS configuration were depicted, either at the partition, QOS or fairsharing levels [30]. The updated account hierarchy, together with the rules set to ensure transparent and representative raw shares for each account set on our facilities allowed for significant improvement on the workload processing efficiency. It is worth to note that the implemented strategy takes into account not only the number of active PIs within a given organization, but also the past funding or past jobs efficiency to give incentives for further contributions and/or higher efficiency when interacting with the ULHPC facility. Due to space restrictions, several practical aspects are not detailed in this article. This includes for instance the server and research computing services management operated in practice by a distributed infrastructure relying on the Puppet [7] and BlueBanquise [1] frameworks. Future directions include the federation with EuroHPC infrastructures to permit research computing and data analytic workflows to be "transparently" migrated from the University facility toward larger Tier-0 systems. Another complementary perspective concerns the possibility to offload some of the less-demanding jobs (typically embarrassingly parallel single-core tasks) onto dynamically allocated virtual cloud instances to free local HPC resources for (more) massively parallel jobs.

Acknowledgments: The experiments presented in this paper were carried out using the HPC facilities of the University of Luxembourg [28] (see `hpc.uni.lu`). The authors are grateful to the alumni staff members of the ULHPC operational team, Valentin Plugaru and Clément Parisot, for their contributions to the concepts developed in this article prior to their departure from the University of Luxembourg. They would like also to thank Dr. Frederic Pinel, Dr. Ezhilmathi Krishnasamy, Dr. Xavier Besseron and Dr. Aurelien Ginolhac for their valuable feedbacks and suggestions for improving this paper and generally for their contributions to user mentoring activities. All plots were generated with `R` [11] and `ggplot2` [32]. Fig. 1, initially generated as piecharts by Dr. Varrette, were adapted in its current concise form by Dr. Ginolhac.

REFERENCES

[1] [n.d.]. BlueBanquise: An open source cluster deployment and management stack. [online]. https://bluebanquise.com/.

[2] [n.d.]. European Environment for Scientific Software Installations Community. [online]. https://www.eessi-hpc.org/.
[3] [n.d.]. High Performance Conjugate Gradients (HPCG) Benchmark. https://www.hpcg-benchmark.org/.
[4] [n.d.]. IO-500 Benchmark. https://www.vi4io.org/io500/.
[5] [n.d.]. IOR: HPC IO Benchmark Repository. https://ior.readthedocs.io/.
[6] [n.d.]. OSU Micro-Benchmarks (OMB). https://mvapich.cse.ohio-state.edu/benchmarks/.
[7] [n.d.]. Puppet: Powerful infrastructure automation and delivery. [online]. https://puppet.com/.
[8] [n.d.]. The Graph 500 Benchmarks BFS (Search) and SSSP(Shortest Path) Specifications. https://graph500.org/.
[9] [n.d.]. The Unified European Application Benchmark Suite (UEABS). https://repository.prace-ri.eu/git/UEABS/ueabs/.
[10] 2022. NVIDIA Data Center GPU Manager (DCGM). https://developer.nvidia.com/dcgm.
[11] 2022. *R: A Language and Environment for Statistical Computing*. https://www.R-project.org/
[12] A. Bhatele, N. Jain, M. Mubarak, and T. Gamblin. 2019. Analyzing Cost-Performance Tradeoffs of HPC Network Designs under Different Constraints Using Simulations. In *Proc. of the ACM SIGSIM Conf. on Principles of Advanced Discrete Simulation (SIGSIM-PADS'19)* (Chicago, IL, USA). ACM, 1–12. https://doi.org/10.1145/3316480.3325516
[13] Peter Braam. 2019. The Lustre Storage Architecture. arXiv:1903.01955 [cs.OS]
[14] European Parliament and Council. 2016. Regulation (EU) 2016/679. http://data.europa.eu/eli/reg/2016/679/oj.
[15] M. Geimer, K. Hoste, and R. McLay. 2014. Modern Scientific Software Management Using EasyBuild and Lmod. In *2014 First International Workshop on HPC User Support Tools (HUST)*. 41–51. https://doi.org/10.1109/HUST.2014.8
[16] IPython. 2015. *ipyparallel*. https://github.com/ipython/ipyparallel
[17] ISO. 2013. *ISO/IEC 27002:2013: Information technology, Security techniques, Code of practice for information security controls* (2nd ed.). ISO. 80 pages. https://www.iso.org/standard/54533.html
[18] S. A. Jyothi, A. Singla, G. P. Brighten, and A. Kolla. 2016. Measuring and Understanding Throughput of Network Topologies. In *SC '16: Proceedings of the Intl. Conf. for HPC, Networking, Storage and Analysis*. 761–772. https://doi.org/10.1109/SC.2016.64
[19] V. Karakasis and al. 2019. Enabling Continuous Testing of HPC Systems Using ReFrame. In *Tools and Techniques for HPC - HUST Workshops, part of SC 2019 (CCIS, Vol. 1190)*. Springer, 49–68. https://doi.org/10.1007/978-3-030-44728-1_3
[20] G. M. Kurtzer, V. Sochat, and M. W. Bauer. 2017. Singularity: Scientific containers for mobility of compute. *PloS one* 12, 5 (2017).
[21] John D. McCalpin. [n.d.]. STREAM: Sustainable Memory Bandwidth in High Performance Computers. cs.virginia.edu/stream/.
[22] Robert McLay. 2015. Lmod: Environmental modules system.
[23] L. Paseri, S. Varrette, and P. Bouvry. 2021. Protection of Personal Data in High Performance Computing Platform for Scientific Research Purposes. In *Proc. of the EU Annual Privacy Forum (APF) 2021 (LNCS, Vol. 12703)*. Springer International Publishing, 123–142.
[24] A. Petitet, R. C. Whaley, J. Dongarra, and A. Cleary. [n.d.]. HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers. https://www.netlib.org/benchmark/hpl/.
[25] Matthew Rocklin. 2015. Dask: Parallel Computation with Blocked algorithms and Task Scheduling. In *Proceedings of the 14th Python in Science Conference*, Kathryn Huff and James Bergstra (Eds.). 130 – 136.
[26] Ronald Ross. 2012. *Guide for Conducting Risk Assessments*. Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD. https://doi.org/10.6028/NIST.SP.800-30r1
[27] F. Schmuck and R. Haskin. 2002. GPFS: A Shared-Disk File System for Large Computing Clusters. In *Proceedings of the 1st USENIX Conference on File and Storage Technologies* (Monterey, CA) *(FAST '02)*. USENIX Association, USA, 19–es.
[28] S. Varrette, P. Bouvry, H. Cartiaux, and F. Georgatos. 2014. Management of an Academic HPC Cluster: The UL Experience. In *Proc. of the 2014 Intl. Conf. on High Performance Computing & Simulation (HPCS 2014)*. IEEE, Bologna, Italy, 959–967.
[29] S. Varrette, H. Cartiaux, T. Valette, and A. Olloh. 2022. Aggregating and Consolidating two High Performant Network Topologies: The ULHPC Experience. In *ACM Practice and Experience in Advanced Research Computing (PEARC'22)*. Association for Computing Machinery (ACM), Boston, USA. https://doi.org/10.1145/3491418.3535159
[30] S. Varrette, E. Kieffer, and F. Pinel. 2022. Optimizing the Resource and Job Management System of an Academic HPC and Research Computing Facility. In *21st IEEE Intl. Symp. on Parallel and Distributed Computing (ISPDC'22)*. IEEE Computer Society, Basel, Switzerland.
[31] S. Varrette, E. Kieffer, F. Pinel, E. Krishnasamy, S. Peter, H. Cartiaux, and X. Besseron. 2021. RESIF 3.0: Toward a Flexible & Automated Management of User Software Environment on HPC facility. In *ACM Practice and Experience in Advanced Research Computing (PEARC'21)*. Association for Computing Machinery (ACM), Virtual Event. https://doi.org/10.1145/3437359.3465600
[32] Hadley Wickham. 2016. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org
[33] M. D. Wilkinson, M. Dumontier, I.J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J-W. Boitenand L.N. da Silva Santos, P.E. Bourne, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* 3, 160018 (2016). https://www.nature.com/articles/sdata201618
[34] A. B. Yoo, M. A. Jette, and M. Grondona. 2003. SLURM: Simple Linux Utility for Resource Management. In *Proc. of Job Scheduling Strategies for Parallel Processing (JSSPP 2003) (LNCS, Vol. 2862)*. Springer Verlag, 44–60.