ICE 2.0: Restructuring and Growing an Instructional HPC Cluster

J. ERIC COULTER, MICHAEL D. WEINER, AARON JEZGHANI, MATT GUIDRY, RUBEN LARA, FANG (CHERRY) LIU, ALLAN METTS, RONALD RAHAMAN, KENNETH J. SUDA, PETER N. WAN, GREGORY WILLCOX, DEIRDRE WOMACK, and DAN (ANN) ZHOU, Georgia Institute of Technology Partnership for an Advanced Computing Environment (PACE), USA

The Partnership for an Advanced Computing Environment (PACE) at Georgia Tech (GT) has been running two campus-wide cluster resources available for academic courses and workshops for five years. The initial design focused on creating a federated resource for a wide range of educational topics, based on a PACE and College of Computing (COC) partnership. Due to funding, this took the form of separate resources, one funded by PACE, and another by COC. These "Instructional Cluster Environments", PACE-ICE and COC-ICE, became very popular with instructors at GT but led to a high maintenance cost due to the split nature of the environments. With the transition to the Slurm scheduler, PACE collaborated with COC to merge the two clusters into one, ICE. This work details the strategies used to sensibly merge the two production systems, including the storage architecture, shared system policies, and scheduler priority configurations that honor funding complexities.

CCS Concepts: • **Applied computing** → **Interactive learning environments**; **Computer-assisted instruction**; • **General and reference** → **Design**; • **Computer systems organization** → *Distributed architectures*; • **Information systems** → *Storage architectures*.

Additional Key Words and Phrases: HPC, Education, System design, Accounting, HPC Access, Instructional Infrastructure

ACM Reference Format:
J. Eric Coulter, Michael D. Weiner, Aaron Jezghani, Matt Guidry, Ruben Lara, Fang (Cherry) Liu, Allan Metts, Ronald Rahaman, Kenneth J. Suda, Peter N. Wan, Gregory Willcox, Deirdre Womack, and Dan (Ann) Zhou. 2023. ICE 2.0: Restructuring and Growing an Instructional HPC Cluster. 1, 1 (August 2023), 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The Instructional Cluster Environment (ICE) resource at Georgia Tech's (GT) Partnership for an Advanced Computing Environment (PACE) had its roots in a request and funding from the GT College of Computing (COC) [4]. The PACE team initially constructed two ICEs. The first was funded by COC, which would support COC courses. Building on its success, PACE funded and built a second cluster shortly after, available for use by any course at GT. These were dubbed COC-ICE and PACE-ICE, respectively. After five years, these resources continue to offer high-quality learning environments for a variety of courses. In the 2022-2023 academic year, 55 courses employed ICE, providing access to

Authors' address: J. Eric Coulter, j.eric@gatech.edu; Michael D. Weiner, mweiner3@gatech.edu; Aaron Jezghani, ajezghani3@gatech.edu; Matt Guidry, mguidry3@gatech.edu; Ruben Lara, rlara3@gatech.edu; Fang (Cherry) Liu, fliu67@gatech.edu; Allan Metts, ametts6@gatech.edu; Ronald Rahaman, rrahaman6@gatech.edu; Kenneth J. Suda, ksuda3@gatech.edu; Peter N. Wan, peter.wan@oit.gatech.edu; Gregory Willcox, gwillcox6@gatech.edu; Deirdre Womack, dowmack30@gatech.edu; Dan (Ann) Zhou, dzhou62@gatech.edu, Georgia Institute of Technology and Partnership for an Advanced Computing Environment (PACE), Atlanta, Georgia, USA.

over 5000 students. In addition, ICE serves regular workshops providing training in scientific computing, taught by PACE research faculty or other GT faculty, and it has also hosted two Open Hackathons in partnership with Nvidia [19].

An intervening datacenter migration [14] allowed for an opportunity to refresh the hardware and software stack of the two clusters, while keeping the architecture constant. However, the separation of the resources began to lead to unnecessarily increased workload on the PACE team, when the purpose and design of each cluster was essentially the same. During a group-wide migration to the Slurm scheduler, we saw an opportunity to simplify the team-wide workload and generally improve the ICE resources, by combining them into a single cluster through our ICE 2.0 project.

The ICE cluster is distinctive in that it uses the same basic hardware types as our larger research-oriented clusters (although it has more variation in order to support student exploration of hardware architectures), and we hope to highlight here the ways in which the design for a campus-wide resource aimed at instructional use differs from those, which are more commonly documented in the cyberinfrastructure (CI) community. While there are a handful of systems purposefully built for educational purposes in the RCD community, for which publications exist, they often take the form of so-called "Microclusters" [1, 2, 6] using clusters of low-performance boards, cloud resources [18], or are discussed largely from a pedagogical perspective [3].

One of the key challenges in widely-available HPC systems is that of managing authentication - whether due to the size of the system itself [5], multiple layers of web software on top of HPC-system level software [15], or simply the struggles inherent in interfacing with separately controlled Identity and Access Management (IAM) systems [16].

The availability of HPC resources for use in non-CS and non-HPC-specific courses is described as a key method of propagating HPC knowledge at the university level by a recent working group on HPC education [23]. In addition to use in traditional courses, the ICE system supports a variety of workshops, similar in spirit to those at other institutions [7], which help grow and strengthen our HPC user community. Having a purpose-built resource and tightly integrated account management workflow for this greatly reduces the strain of providing new users access to a more research-focused cluster [9] with minimal human intervention in the account provisioning process [16].

Our migration from two separate clusters to one instructional resource provides an excellent opportunity to share design lessons learned over the last several years, which we have implemented in the final ICE 2.0 design and policies. Initially, we discuss our goals for the ICE project, including how we hope to improve resource utilization and wait times, goals of the merge, and migration from the Adaptive Computing Torque/Moab [8, 25] scheduler and resource manager to Slurm [26]. We describe the storage layout and data merge in detail, including an autofs configuration that allows us to interface cleanly with central IAM for account provisioning. Finally, we describe a floating partition and Quality-of-Service setup which allows us to satisfy a variety of stakeholder requirements while providing a straightforward set of QOS levels for new users.

2 PROJECT GOALS

The ICE 2.0 project aimed to restructure our instructional services and improve student experiences by increasing resource access, updating our scheduler and scientific software, and introducing faster storage. We also sought to streamline systems administration and report usage metrics more effectively.

The division between two ICE clusters, one for the College of Computing and one for other colleges, limited student access to resources, increasing wait times in periods of peak utilization, and increased the administrative effort to maintain the systems. In collaboration with COC, PACE merged the two clusters to form the new ICE. Overflow between nodes increases availability of resources for all courses and decreases wait times while preserving priority for COC courses on former COC-ICE nodes and for other colleges' courses on former PACE-ICE nodes. All students also gained the opportunity to experiment with the additional GPU architectures added by COC in recent years, including Nvidia A40 and A100 and AMD MI210 architectures. Merging the clusters also eliminated confusion, as students and instructors frequently attempted to access the wrong login node via ssh or browsed to the wrong Open OnDemand [10] interface between the similarly-named clusters.

ICE 2.0 introduced the Slurm scheduler to ICE, replacing the Moab and Torque schedulers previously in place. PACE's research clusters were in the progress of migrating schedulers in the same year, while also receiving a brand-new software stack with updated compilers, libraries, and scientific software applications [13]. ICE received the same software, aligning the research and instructional environments, which eases the transition between clusters for those engaged in advanced computing for both research and education while requiring the PACE team to maintain only one software stack for both systems.

The third goal of ICE 2.0 was to improve storage speed and capacity, in response to feedback from faculty teaching courses on the cluster. COC and PACE collaborated to purchase a Lustre parallel filesystem, much like the one on the flagship Phoenix research cluster [12], to provide new faster storage for ICE. This device provides faster performance than the home directories on ICE and is set up as scratch space for temporary storage. We also aimed to improve provisioning of shared directories, increasingly popular as space for course materials or collaborative work, and to introduce fixed methods and schedules for the removal of old data from ICE storage systems.

Beyond student-facing aspects, we also aimed to improve the administration of the cluster. Merging the two instructional clusters simplified management of the systems, while new data collection and organization offers better metrics for evaluating the impact of ICE on students.

3 STORAGE DESIGN

3.1 Merging Old Home Directories

One of the key challenges with this project was to preserve existing home directory data in a way that will be accessible to students and instructors returning to the system. This became non-trivial due to the existence of some 16,000 home directories across both of the previous systems, some for users with accounts on both systems. The natural solution was to create sub-directories in each user's *new* home directory with the content of their previous home directories below, named to indicate the system of origin. With an enormous number of files and directories, doing this efficiently became a small scripting challenge resolved with a combination of xargs and find [21]. Given the lack of log data around time of last access or entitlement, our only option to cut down on the number of home directories we maintain was to check for whether or not a user still had an active entitlement in the GT LDAP system. Once this was accounted for, we only had to transfer current student and employee home directories to the new storage location. This was initially done by spawning an rsync process for each home directory, parallelized via xargs. Use of the '-S' flag during the initial sync was important, because we discovered a large number of students with huge sparse files, which if copied incorrectly would put them over quota on the new system.

An additional challenge we encountered on the old system was that home directories were all created in a single directory, which led to incredibly slow access times, when the number of accounts gradually ballooned to nearly 16,000 directories. This was resolved on the new system by implementing "bucketing" of home directories based on the last two digits of UID (which is more evenly distributed than switching on letters of names). This led to ten directories with ten subdirectories each (one hundred total "buckets") for home directories, ensuring that even in case every single student and faculty member at GT were to gain access to ICE, we need not expect more than 500 directories at a single level, much more manageable than 20,000!

However, this change wreaked havoc with the options available to us in using LDAP+SSSD [22] to handle authentication and user home directory mapping. The central IAM system has no way of providing us with correct home directories for each account specific to ICE, so mapping is necessary, which is typically handled through SSSD. Unfortunately, the available plugins for SSSD and AD only allow for mapping on *letters of username* which as we already noted, led to statistical aberrations where more popular letters result in heavily weighted "buckets". Thus, we investigated the use of AutoFS, to generate custom mappings for each user based on UID. There is a poorly-documented feature whereby it is possible to write a script to customize the available mapping from username to mount point. A custom entry to the AutoFS master map, added to `/etc/autofs.master.d`, is provided to map the NetApp home mount to `${HOMEROOT}/${UID: -2:1}/${UID: -1}/${USER}`, providing the aforementioned two-level bucketing scheme. In this way, we can accommodate the significant number of users required to serve all of campus.

3.2 Infrastructure

Home directories are hosted on a 13-TB NetApp [17] device with an SSD pool, which also holds a separate volume for shared directories, while scratch directories are located on a 1-PB Lustre parallel filesystem [11]. Daily snapshots on the NetApp provide protection against accidental deletion of important files.

3.3 Storage Policies

While upgrading ICE, we took the opportunity to revisit our storage and data retention policies, creating a clearer and more sustainable plan. Each user account on the cluster, whether student or instructor, receives a 15 GB home directory and a 100 GB scratch directory. Instructors may request quota exceptions for themselves or individual students if needed. For home directories, a snapshot is taken daily in case data needs to be retrieved after accidental file loss. No backups are made of scratch. Files in home directories are deleted after a user has not had access to ICE or logged in for one year. At the end of each semester, all files in scratch directories not touched in 120 days are deleted.

Shared directories are placed by default on a NetApp volume. As they are on a separate volume from the home directories, they do not count towards individual user quotas, greatly reducing the need for quota exceptions which we encountered frequently under the prior design which used a single volume for home and shared directories. Shared directories are deleted two years after the course is last taught using ICE.

Instructors may request shared directories on the Lustre (scratch) parallel filesystem for faster performance. These shared directories have no backup, so they are best used for data that could be retrieved from another location if it needed to be recreated. Files in Lustre shared directories count towards the scratch quota of the user who owns them, even though they are located outside the user's scratch directory.

4 SCHEDULER DESIGN

Originally, the COC-ICE and PACE-ICE clusters were managed as isolated resources, each with their own dedicated scheduler and resource manager. Per Table 1, COC-ICE totaled 51 servers with a collection of 2nd Generation Intel Scalable Xeon (Cascade Lake) and 3rd Generation AMD EPYC (Milan) processor CPU nodes and GPU nodes with an array of AMD MI210 Instinct and Nvidia Quadro RTX6000, Tesla V100, and Ampere A40 and A100 accelerators; meanwhile, PACE-ICE offered Cascade Lake CPU and Nvidia Quadro RTX6000 or Tesla V100 GPU nodes. While PACE-ICE was fairly simple in implementation, having only two Torque queues for CPU and GPU jobs respectively, COC-ICE provided a more robust queuing structure, with nine queues configured with different resource limits according to policy, such as job size, length, or purpose (grading versus coursework), and architecture (as Intel and AMD hardware had separate queues due to differing software stacks). With the merger of the two clusters and the migration to Slurm, we opted to simplify scheduling by:

(1) collapsing the COC-ICE queues by leveraging two-dimensional resource limits such as CPU-time,
(2) maintaining common priority policies for users coming from different colleges, and
(3) creating a modular infrastructure that could be augmented by additional resource additions.

Resource	Nodes	CPUs	GPUs
COC-ICE	51	2600	84
PACE-ICE	19	432	14
Total	**90**	**4312**	**258**

Table 1. Composition of the original COC-ICE and PACE-ICE clusters as used to set limits on floating partitions.

Priority and resource access are managed using a combination of partitions and QOS designations, which are applied passively via a Lua job submit filter to enforce policy. To preserve priority based on original hardware purchases while enabling access to a larger pool of resources for spillover capability, the partitions in Slurm were arranged as a two-level system using the "PriorityTier" attribute [24]: the higher priority partition defined as a floating resource with CPUs or GPUs based on the previous cluster using a partition QOS, and the lower priority partition mapping to all hardware. Within each priority tier, there are two partitions, one for CPU-only nodes and the other for GPU nodes. As summarized in Table 2, three levels of QOS designations manage scheduling priorities and resources limits for scheduling jobs within a partition: Instructor/TA for quick turnaround on grading, Enrolled Students to ensure assignments can be completed on time, and, in the future, General Students to allow independent work on the cluster's idle cycles. As described further in Section 5, user allowed and default QOS are determined based on entitlements in LDAP, meaning that this whole process is fully automated based on user management at the college and registrar level.

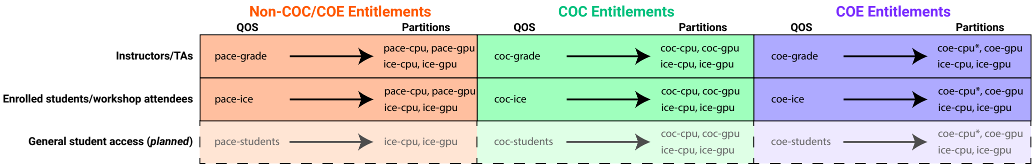
Following the finalization of the ICE 2.0 merger and Slurm configuration details, the College of Engineering (COE) proceeded with a hardware purchase to augment ICE and provide priority access to students from the college. As intended, the additional college was able to be implemented as a third vertical in the configurations, validating the templated approach almost immediately after its inception.

The Lua job submit plugin is leveraged to route jobs correctly to the partitions based on user QOS and resource requests, as shown in Figure 1. With the exception of the planned non-COC/COE general students, each user's job is submitted to the high priority floating CPU or GPU partition and the low priority CPU or GPU partition; additionally, unless a specific GPU architecture is requested, the job submit plugin appends the feature list to request an Nvidia GPU.

QOS Level	Priority	Wallclock Limit	Max Jobs	Max CPU-time	Max GPU-time	Preemptable
Instructor/TA	High	12 hours	10	768 CPU-hrs	24 GPU-hrs	No
Enrolled Students	Medium	8 hours	500	512 CPU-hrs	16 GPU-hrs	No
General Students	Zero	TBD	TBD	TBD	TBD	Yes

Table 2. The three QOS levels and their resource limits. These three levels are repeated for each vertical (e.g. college) with investments in ICE.

The effect of this configuration is to allow prioritization based on purchased hardware, but also leverage opportunistic cycles from the cluster as a whole, improving cluster utilization and efficiency.



	Non-COC/COE Entitlements QOS	Partitions	COC Entitlements QOS	Partitions	COE Entitlements QOS	Partitions
Instructors/TAs	pace-grade	pace-cpu, pace-gpu ice-cpu, ice-gpu	coc-grade	coc-cpu, coc-gpu ice-cpu, ice-gpu	coe-grade	coe-cpu*, coe-gpu ice-cpu, ice-gpu
Enrolled students/workshop attendees	pace-ice	pace-cpu, pace-gpu ice-cpu, ice-gpu	coc-ice	coc-cpu, coc-gpu ice-cpu, ice-gpu	coe-ice	coe-cpu*, coe-gpu ice-cpu, ice-gpu
General student access (*planned*)	pace-students	ice-cpu, ice-gpu	coc-students	coc-cpu, coc-gpu ice-cpu, ice-gpu	coe-students	coe-cpu*, coe-gpu ice-cpu, ice-gpu

Fig. 1. The hierarchy of QOS levels (instructor, enrolled student, or general student) for the three verticals, and the associated partitions to which jobs are submitted.

5 ACCOUNT MANAGEMENT

Account management for ICE is handled through GT's LDAP server. The details of enabling this setup have been described previously for the original ICE implementation [4] and remain in place. For the merged ICE, several new features were added to incorporate priority on different portions of the cluster, to improve the availability of usage metrics, and to ensure old data can be properly removed.

Collecting usage metrics is essential to understanding the use of valuable cyberinfrastructure resources and justifying investment in them. For an instructional cluster, this includes courses supported, numbers of students running compute jobs, utilization fraction of CPU and GPU resources, and more. We analyze much of this information through our instance of Open XDMoD [20].

Course enrollments are translated within the identity and access management system into school-based POSIX groups, which populate on the cluster. From here, we run scripts every 30 minutes (via cron) to identify new users added and create home directories and Slurm accounts for them. Slurm accounts, new in ICE 2.0, are based on the specific POSIX group, as described in Section 4, allowing us to tie an individual student's compute job to the school

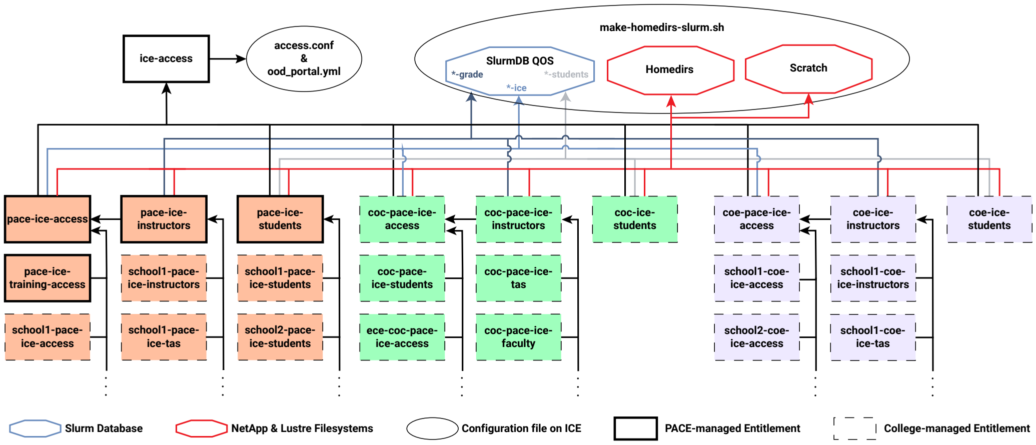


Fig. 2. The home directory and account population script for ICE. School- and College-owned POSIX groups feed into PACE-owned POSIX groups using PACE-managed rules, simplifying cluster access and storage management. Additionally, intermediate-level entitlements are used to automate Slurm QOS.

offering the course in which they are enrolled. For privacy reasons, individual course enrollments are not tracked. The full updated process of account provisioning is shown in Figure 2, including a top-level POSIX group for enabling ssh to the login nodes via $access.conf$ and access to the Open OnDemand portal.

With the addition of Slurm accounts, we can now determine metrics in XDMoD by school offering the course, providing much more fine-grained detail beyond the prior information, which was limited to only two buckets: COC and all other colleges. To further increase our understanding of the students on ICE, we began recording from the identity management database student majors, which reflects the diversity of student population learning scientific computing skills through ICE beyond only the schools offering ICE courses.

The POSIX groups for instructors and TAs are also recognized by the updated account creation script, populating them into the higher-priority "grade" QOS on the scheduler. This priority was previously introduced for TAs in COC to grade student coding assignments and is now expanded to support any instructor efforts campuswide to prepare or grade course materials.

The account management scripts have also been updated to track membership in the relevant access POSIX groups each day. This allows us to implement the storage policies in Section 3.3 and remove data for former students after sufficient time has passed.

6 CONCLUSIONS AND FUTURE WORK

Building on the five-year-old Instructional Cluster Environments (ICE), Georgia Tech's PACE completed the ICE 2.0 project to update and improve our instructional environment, merging two clusters, adopting the Slurm scheduler, updating scientific software, compilers, and libraries, adding a parallel filesystem, and improving account management and metrics. The upgraded cluster is designed to provide an increasing student population with a better educational experience, with more resource availability, more compute architectures, faster storage, and current software.

ACKNOWLEDGMENTS

The authors would like to acknowledge our colleagues at PACE for their support of these efforts. We thank the GT College of Computing for their ongoing collaboration focused on student success as well as financial contributions to ICE. The GT Student Technology Fee has funded much of ICE's compute and storage hardware through a series of grants, with additional support from the Georgia Tech Research Institute. We thank the GT College of Engineering for its recent investment to expand ICE and increase the number of students it can support. Finally, we acknowledge the faculty and students of courses employing ICE for their suggestions and feedback guiding its continued improvement.

REFERENCES

[1] Joel C. Adams, Suzanne J. Matthews, Elizabeth Shoop, David Toth, and James Wolfer. 2017. Using Inexpensive Microclusters and Accessible Materials for Cost-Effective Parallel and Distributed Computing Education. *The Journal of Computational Science Education* 8 (Dec. 2017), 2–10. Issue 3. https://doi.org/10.22369/issn.2153-4136/8/3/1

[2] Lluc Alvarez, Eduard Ayguade, and Filippo Mantovani. 2018. Teaching HPC Systems and Parallel Programming with Small-Scale Clusters. In *2018 IEEE/ACM Workshop on Education for High-Performance Computing (EduHPC)*. 1–10. https://doi.org/10.1109/EduHPC.2018.00004

[3] Michael E. Baldwin, Xiao Zhu, Preston M. Smith, Stephen Lien Harrell, Robert Skeel, and Amiya Maji. 2016. Scholar: A Campus HPC Resource to Enable Computational Literacy. In *2016 Workshop on Education for High-Performance Computing (EduHPC)*. 25–31. https://doi.org/10.1109/EduHPC.2016.009

[4] Mehmet Belgin, Trever C. Nightingale, David A. Mercer, Fang Cherry Liu, Peter Wan, Andre C. McNeill, Ruben Lara, Paul Manno, and Neil Bright. 2018. ICE: A Federated Instructional Cluster Environment for Georgia Tech. In *Proceedings of the Practice and Experience on Advanced Research Computing* (Pittsburgh, PA, USA) *(PEARC '18)*. Association for Computing Machinery, New York, NY, USA, Article 16, 7 pages. https://doi.org/10.1145/3219104.3219112

[5] Brett Bode, Tim Bouvet, Jeremy Enos, and Sharif Islam. 2016. Account Management of a Large-Scale HPC Resource. In *In HPCSYSPROS16: HPC System Professionals Workshop*. Salt Lake City, UT. https://doi.org/10.5281/zenodo.4327649

[6] Caughlin Bohn and Carrie Brown. 2020. Legion: A K-12 HPC Outreach and Education Cluster. In *Practice and Experience in Advanced Research Computing* (Portland, OR, USA) *(PEARC '20)*. Association for Computing Machinery, New York, NY, USA, 448–451. https://doi.org/10.1145/3311790.3400845

[7] Dhruva K. Chakravorty, Marinus "Maikel" Pennings, Honggao Liu, Zengyu "Sheldon" Wei, Dylan M. Rodriguez, Levi T. Jordan, Donald "Rick" McMullen, Noushin Ghaffari, and Shaina D. Le. 2019. Effectively Extending Computational Training Using Informal Means at Larger Institutions. *The Journal of Computational Science Education* 10 (Jan. 2019), 40–47. Issue 1. https://doi.org/10.22369/issn.2153-4136/10/1/7

[8] Adaptive Computing. 2023. Moab HPC Suite. Retrieved August 3, 2023 from https://adaptivecomputing.com/moab-hpc-suite/

[9] Henry A. Gabb, Alexandru Nicolau, Satish Puri, Michael D. Shah, Rahul Toppur, Neftali Watkinson, Weijia Xu, and Hui Zhang. 2021. Lightning Talks of EduHPC 2021. In *2021 IEEE/ACM Ninth Workshop on Education for High Performance Computing (EduHPC)*. St. Louis, MO.

[10] Dave Hudak, Doug Johnson, Alan Chalker, Jeremy Nicklas, Eric Franz, Trey Dockendorf, and Brian L. McMichael. 2018. Open OnDemand: A web-based client portal for HPC centers. *Journal of Open Source Software* 3, 25 (2018), 622. https://doi.org/10.21105/joss.00622

[11] Cluster File Systems Inc. 2002. Lustre : A Scalable , High-Performance File System Cluster. Retrieved March 28, 2022 from https://cse.buffalo.edu/faculty/tkosar/cse710/papers/lustre-whitepaper.pdf

[12] Aaron Jezghani, Semir Sarajlic, Michael Brandon, Neil Bright, Mehmet Belgin, Gergory Beyer, Christopher Blanton, Pam Buffington, J. Eric Coulter, Ruben Lara, Lew Lefton, David Leonard, Fang Cherry Liu, Kevin Manalo, Paul Manno, Craig Moseley, Trever Nightingale, N. Bray Bonner, Ronald Rahaman, Christopher Stone, Kenneth J. Suda, Peter Wan, Michael D. Weiner, Deirdre Womack, Nuyun Zhang, and Dan Zhou. 2022. Phoenix: The Revival of Research Computing and the Launch of the New Cost Model at Georgia Tech. In *Practice and Experience in Advanced Research Computing* (Boston, MA, USA) *(PEARC '22)*. Association for Computing Machinery, New York, NY, USA, Article 13, 9 pages. https://doi.org/10.1145/3491418.3530767

[13] Fang (Cherry) Liu, Ronald Rahaman, Michael D Weiner, J Eric Coulter, Deepa Phanish, Jeffrey Valdez, Semir Sarajlic, Ruben Lara, and Pam Buffington. 2023. Semi-Automatic Hybrid Software Deployment Workflow in a Research Computing Center. In *Practice and Experience in Advanced Research Computing* (Portland, OR, USA) *(PEARC '23)*. Association for Computing Machinery, 9 pages.

[14] Fang Cherry Liu, Michael D. Weiner, Kevin Manalo, Aaron Jezghani, Christopher J. Blanton, Christopher Stone, Kenneth Suda, Nuyun Zhang, Dan Zhou, Mehmet Belgin, Semir Sarajlic, and Ruben Lara. 2021. Human-in-the-Loop Automatic Data Migration for a Large Research Computing Data Center. In *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*. 1752–1758. https://doi.org/10.1109/CSCI54926.2021.00068

[15] Ping Luo, Benjamin Evans, Tyler Trafford, Kaylea Nelson, Thomas Langford, Jay Kubeck, and Andrew Sherman. 2021. Using Single Sign-On Authentication with Multiple Open OnDemand Accounts: A Solution for HPC Hosted Courses. In *Practice and Experience in Advanced Research Computing* (Boston, MA, USA) *(PEARC '21)*. Association for Computing Machinery, New York, NY, USA, Article 15, 6 pages. https://doi.org/10.1145/3437359.3465575

[16] Junya Nakamura and Masatoshi Tsuchiya. 2018. Automated User Registration Using Authentication Federation on Academic HPC System. In *2018 7th International Congress on Advanced Applied Informatics (IIAI-AAI)*. 61–67. https://doi.org/10.1109/IIAI-AAI.2018.00022

[17] NetApp. 2022. Data Management Solutions for the Cloud. Retrieved March 28, 2022 from https://www.netapp.com/

[18] Linh B. Ngo and Jon Kilgannon. 2020. Virtual Cluster for HPC Education. *J. Comput. Sci. Coll.* 36, 3 (oct 2020), 20–30.

[19] OpenACC. 2023. Open Hackathons. Retrieved August 3, 2023 from https://www.openhackathons.org

[20] Jeffrey T. Palmer, Steven M. Gallo, Thomas R. Furlani, Matthew D. Jones, Robert L. DeLeon, Joseph P. White, Nikolay Simakov, Abani K. Patra, Jeanette Sperhac, Thomas Yearke, Ryan Rathsam, Martins Innus, Cynthia D. Cornelius, James C. Browne, William L. Barth, and Richard T. Evans. 2015. Open XDMoD: A Tool for the Comprehensive Management of High-Performance Computing Resources. *Computing in Science & Engineering* 17, 4 (2015), 52–62. https://doi.org/10.1109/MCSE.2015.68

[21] GNU Project. 2021. findutils. Retrieved August 3, 2023 from https://www.gnu.org/software/findutils/

[22] SSSD Project. 2021. System Security Services Daemon. Retrieved August 3, 2023 from https://sssd.io/

[23] Rajendra K. Raj, Carol J. Romanowski, John Impagliazzo, Sherif G. Aly, Brett A. Becker, Juan Chen, Sheikh Ghafoor, Nasser Giacaman, Steven I. Gordon, Cruz Izu, Shahram Rahimi, Michael P. Robson, and Neena Thota. 2020. High Performance Computing Education: Current Challenges and Future Directions. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education* (Trondheim, Norway) *(ITiCSE-WGR '20)*. Association for Computing Machinery, New York, NY, USA, 51–74. https://doi.org/10.1145/3437800.3439203

[24] SchedMD. 2023. slurm.conf. Retrieved August 3, 2023 from https://slurm.schedmd.com/slurm.conf.html#OPT_PriorityTier

[25] Garrick Staples. 2006. TORQUE Resource Manager. In *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing* (Tampa, Florida) *(SC '06)*. Association for Computing Machinery, New York, NY, USA, 8–es. https://doi.org/10.1145/1188455.1188464

[26] Andy B Yoo, Morris A Jette, and Mark Grondona. 2003. Slurm: Simple linux utility for resource management. In *Job Scheduling Strategies for Parallel Processing: 9th International Workshop, JSSPP 2003, Seattle, WA, USA, June 24, 2003. Revised Paper 9*. Springer, 44–60.

Received 04 August 2023