November 2023

Volume 14 Issue 2

JoCsE

Journal Of Computational Science Education

Promoting the Use of Computational Science Through Education

ISSN 2153-4136 (online)

Volume 14, Issue 2 **November 2023**

JOCSE

Journal Of Computational Science Education

The Journal of Computational Science Education (JOCSE), ISSN 2153-4136, published in online form, is a supported publication of the Shodor Education Foundation Incorporated. Materials accepted by JOCSE will be hosted on the JOCSE website and will be catalogued by the Computational Science Education Reference Desk (CSERD) for inclusion in the National Science Digital Library (NSDL).

Subscription: JOCSE is a freely available online peer-reviewed publication which can be accessed at `http://jocse.org`.

CONTENTS

Introduction to Volume 14, Issue 2

David Joiner
Editor
Kean University
Union, NJ
djoiner@kean.edu

FOREWORD

This issue will focus on papers presented at the Sixth Workshop on Strategies for Enhancing HPC Education and Training (SEHET23) at the 2023 Practice & Experience In Advanced Research Computing conference (PEARC23), as well as 2 papers from the 9th Best Practices in HPC Training and Education (BPHTE22) workshop at SC22.

The 9th annual workshop on Best Practices in HPC Training and Education was held on Monday, November 14th, in Dallas TX, and many of the talks for this meeting were featured in Issue 1 of this year. Two additional papers are presented in Issue 2.

Lawrence et al. describe advancements in the Summer Computing Academy (SCA) at High Performance Computing centers and Computer Science departments, focusing on Data Sciences and Cybersecurity for secondary school students. Chakraborty et al. present a series of credentialed short courses designed to provide university students and researchers with vital digital skills in high performance computing, complemented by micro-credentials that integrate with existing academic programs.

The 6th workshop on Strategies for Enhancing HPC Education and Training was held on Monday, July 24, 2023. 5 papers from this conference are presented in this issue.

Bautista and Sukhija present a project addressing the need for diversity in High Performance Computing (HPC) and STEM fields, in which the National Energy Research Scientific Computing Center (NERSC) at Lawrence Berkeley National Lab partnered with a community college to create a pathway for students from disadvantaged communities. Ponce and van Zon discuss the importance of cybersecurity awareness among end-users of remote computing systems, offering training techniques to mitigate cyber threats. Mehringer et al. present the findings of a survey conducted to understand how the High Performance Computing (HPC) community shares and discovers educational and training materials, exploring whether current methods meet their needs and the interest in enhancing these processes. Jezghani et al. present a comprehensive overview of the Lucata Pathfinder system at Georgia Tech's Rogues Gallery and its role in advancing High Performance Computing (HPC) education, particularly the challenges of designing instructional material for new and novel architectures. Speyer et al. describes a joint project between Arizona State University and CR8DL, Inc., which deployed a Jupyter-notebook-based interface to datacenter resources for a semester-long computational biochemistry course.

These 7 papers present exciting and novel work in the field of computational science education, and JOCSE is excited to continue partnering with these organizations to bring these to you. We thank the work of all of the reviewers, workshop chairs, and committee members involved. While we cannot list them all here, I would like to especially thank Nitin Sukhija and Nia Alexandrov for their continued work with these two events.

We hope to see your future papers submitted here to JOCSE, and appreciate your continued support of JOCSE as well as of the SEHET and BPHTE workshop series.

Sincerely,
Dave Joiner

Creating Pathways in Disadvantaged Communities Towards STEM and HPC

Elizabeth Bautista
Lawrence Berkeley National Laboratory
Berkeley, CA
ejbautista@lbl.gov

Nitin Sukhija
Slippery Rock University of Pennsylvania
Slippery Rock, PA
nitin.sukhija@sru.edu

ABSTRACT

Today's job market has its challenges in gaining proficient staff but more so in the High Performance Computing area and within a government lab. Competition from industry in terms of the type of perks they provide, being able to negotiate a higher salary and opportunities of remote work all play a part in losing candidates.

At the National Energy Research Scientific Computing Center (NERSC) at Lawrence Berkeley National Laboratory (LBNL), a site reliability engineer manages the data center onsite 24x7. Further, the facility itself is a unique and complex ecosystem that uses evaporative cooling and recycling of hot air to keep the facility cool. This is in addition to the normal areas to be monitored like the computational systems, the three tier storage, as well as infrastructure and cybersecurity.

To explore creating interest into HPC and STEM within the disadvantaged communities near the Laboratory, NERSC partnered with a community college during the pandemic to support high school seniors and freshmen students to provide an educational foundation. In collaboration with the community college, they created a program of specific classes that students needed to take to prepare them for an HPC and/or STEM internships. In certain demographics, students do not believe they can be successful in science or math and require support from the program such as tutors to help them through. With this type of support, students have successfully completed their classes with passing grades.

As part of their recruitment process for site reliability engineers to continue to support diversity initiatives at the Laboratory, NERSC implemented an apprenticeship program. This paper describes the current work that includes partnering with a community college program and then NERSC provides a summer internship for the student so they can gain hands-on experience. The first cohort of students have graduated into their internship programs this summer. This paper demonstrates early results from this partnership and how it has impacted the diverse pool of candidates at NERSC.

KEYWORDS

Site Reliability Engineer, HPC Education, HPC Training, Diversity, Inclusion, STEM

https://doi.org/10.22369/issn.2153-4136/14/2/1

1 INTRODUCTION

According to a 2014 study of the Public Policy Institute of California, the state is likely to face a shortage of staffing, as high as 1.5 million workers who have college degrees, much more than previous projections depending on the industry [5]. In the post pandemic timeframe, most tech industry workers are seeking amenities like a hybrid or remote working environment that provides the flexibility they require incorporated into their normal life. This shortage is more prominent when recruiting for a government laboratory and in high performance computing (HPC). This industry simply does not have the salaries to compete with the neighboring Silicon Valley tech companies even if they are providing a hybrid or remote environment.

To further complicate matters, the site reliability engineer (SRE) at NERSC, requires staff to be onsite at minimum of two to three eight hour shifts to support the 24x7 requirement of the data center control room. Although the position does not necessarily require a college degree or certifications, the job description does require some system administration, networking, storage and facility management understanding at minimum for a candidate to be successful.

Diversity and inclusion has always been a strong component of the Lab's mission. In fact, the Lab has programs that specifically recruit potential candidates and support internship programs where students or candidates are from the underserved communities in the neighboring areas. In spite of these programs, however, according to a 2022 Lab study of our workforce demographics, we continue to see a small percentage of staff who are underrepresented such as Black/African American, American Indian/Alaska Native, Asian, Hispanic or Latino, etc. We see an even smaller percentage of women in these demographics especially within the Lab Senior Leadership roles [4]. The numbers truly tell a compelling story.

Prior to the pandemic, NERSC had an initiative to expand diversity for site reliability engineers through an innovative apprenticeship program [3]. Though this program was successful in filling all the spots in the control room, the pandemic has created a bigger challenge in hiring and recruiting.

How then can we increase our diversity and inclusion percentages in these areas while still being able to recruit much needed skills into our positions? NERSC continues to gather new ideas to recruit staff and also fulfill the much needed diversity requirements of the Lab. Therefore, by partnering with a community college in the neighborhood, we can potentially create a pipeline by influencing students' education with the much needed support and guidance to pass the required classes to qualify for a technical internship at the Lab.

This paper will document the process for creating this new pipeline and tells the story of early success of the first cohort of students from the program. Section 2 will provide the background of how the pandemic created situations where students were

under prepared for their technical classes and the support the program provided. Section 3 will explain the program itself and the creation of an internship program. Section 4 provides the logistics, such as funding and how the summer program worked. Section 5 will provide case studies of positive outcomes and Section 6 will provide final thoughts and future work.

2 BACKGROUND

In the three years where the world experienced the COVID-19 pandemic, there was a disruption in traditional education and learning. Hundreds of students and faculty in California and globally, transitioned to e-learning. What they did not realize is the impact of that disruption to approximately 91% of the population [8]. The assumption at the time was that since most students were used to a digital format, that learning this way, in the comfort of their home, should be easy. However, the challenge comes from not having the flexibility and capabilities to adapt to the new environment where decision making and problem solving would be tested in addition to the learning curve of dealing with technology. Learning formulas, engineering ideas and coding aren't anywhere close to playing with TikTok.

Enforcing e-learning should have lessened the impact of the school closures but no one expected to depend on the use of high-tech tools and platforms to guarantee success in teaching and learning. According to the November 2022 California test scores survey [7], there was an impact in enrollment by at least 6% across the board but a more disturbing fact is that engagement of students became average. This implies that especially in the STEM areas, the students may not have been engaged enough to learn the complex theories and processes to advance appropriately to the next series of classes.

In discussion with Laney College in Alameda, one of the neighboring schools of Berkeley Lab, they thought that perhaps if students had the support to understand STEM concepts that they would get better grades and be more confident in their learning. Further, if we can provide a cohort of students so they can support each other with a program that will prepare them for a hands-on internship, perhaps we can create a pipeline from the school into industry. As a result, Laney College created a first-year engineering program consisting of these classes:

Semester 1

- Introduction to Engineering
- Introduction to Electrical Engineering
- Statistics
- Programming for Engineers using MATLAB

Semester 2

- Engineering Graphics
- Advanced Statistics
- Mechanics of Materials
- Properties of Materials
- Advanced Programming for Engineers using MATLAB

3 ENGINEERING AT LANEY COLLEGE

Once the classes were determined and teachers were assigned to teach the classes, the program needed to create a support program for the students. This included a tutoring lab that had an engineering trained instructor as a main tutor as well as student tutors and an engineering club that engaged external speakers in the industry to engage students to learn about the type of potential jobs they can have after the program as well as engaging university representatives to encourage students to transfer to a four year university.

However, classes and outreach will do very little if there wasn't a real hands-on goal for students to work toward. The solution was to engage in an internship program with the neighboring organizations. The goal was to have the first year cohort participate in a summer internship program.

The next several months were used to engage the companies and Berkeley Lab was one of them. A job description was needed by each organization who would commit to hire interns with a specific component of hands-on work. This means, they needed to be onsite, 40 hours a week, and they would work with various mentors to learn for the next 10 weeks. Further, they were required to present their work using either a poster session or an oral presentation.

4 LOGISTICS

Growth Sector [6] is an organization that creates pathways for young people toward STEM. They collaborate with many educational institutions across the country and provide support for students to get their foot into the STEM workforce. This organization approached Laney College with a grant to help support their new engineering program cohort. They provided resources such as salary support for the tutoring lab, funding support for the engineering club and agreed to help administer the summer program including paying for the students' salaries during their ten week internship. With this type of financial support, the engineering program was geared toward success.

Employers were motivated to hire a Laney student because the salaries were paid for by Growth Sector. Students had to turn in time cards on a biweekly basis and that is approved by the supervisor of record at the organization. In some instances, they even provided the student housing and travel expenses to ensure they are closer to their employers' locations.

By week eight of the program, the students were well on their way to preparing for their presentations. Depending on the organization, a poster session or an oral presentation was going to be part of the last week of the program where students were coached on how to prepare their poster and/or their presentation speeches with power point.

5 CASE STUDIES

This section will discuss the positive outcomes of three students in the program at NERSC.

Student #1 is a female student who had a prior career in HR but she was laid off from her position at the beginning of the pandemic. Rather than getting into more debt for education, she enrolled in Laney College to get training into STEM. Apparently, this was all new therefore, she needed all the support she could get to get through her classes. Being a single parent as well, she had some challenges maintaining her grades, completing her homework and taking care of her small child. However, she persevered.

When the summer arrived, she applied to various internships at the advice of Growth Sector and Laney advisors. However, by the middle of May, she was quite disappointed that she had not

gotten an offer and she was close to giving up. The thought that perhaps, this was not going to happen this summer.

One of the tutors spoke to a NERSC manager, who already recruited two students from the program. He emphasized that this particular student was hardworking and smart. She just needs a chance. After an interview, it was determined that this student had received good grades in the classes and she participated in extracurricular activities that would enhance her studies. Therefore, this manager decided to take a chance and hire her.

Student #1 was very eager to learn anything hands-on from system administration to laying down networking cables to building servers. Half way through the program, she was invited to learn the job as a site reliability engineer. She wasn't sure about her capabilities but she decided to learn it and by the end of the program, the manager extended her time into the fall. A quote from this student at the end of the program, "I've had the simultaneous challenge of learning to leverage my old soft skills in a new context while learning new technical skills."

Student #2 was a recent graduate from San Francisco State University with a business and marketing degree. After many months of job hunting, she became discouraged and decided to pivot into STEM. She also entered Laney College to minimize taking on more debt for her education. She initially was interested in STEM but was very discouraged when the environment in the four year university did not support her dream. She was told that it was a steep learning curve and she was not confident in her skills to attain it therefore, she changed her major into business.

This particular student graduated recently but learning the classes in the cohort was a challenge. She definitely needed the support of the tutors to get through. She was chosen at NERSC to work on updating the 3D model of the HPC floor and she successfully did so by the end of the program including making suggestions to implement additional features. She was also extended after the summer program.

The last student, student #3, is a young freshman student who originally intended to study civil engineering but changed his major to computing science at Laney. Because he took some of the cohort classes, he also participated in some of the engineering club activities, including hearing one of the NERSC managers speak about careers in HPC and about the NERSC data center. After the talk, he approached her and she encouraged him to join the cohort program to take advantage of the tutoring and support, which he did. He was chosen to be one of the interns for the summer. While he consistently volunteered for much of the hands-on work, his primary goal was to research the replacement of ovirt, an opensource virtualization software management program that managed NERSC's Operations Monitoring and Notification Infrastructure (OMNI) [2]. He too successfully completed his research and summer internship and was extended through the fall to continue testing and implementation of the new virtualization software.

The three students not only presented their work at the Lab but also had the opportunity to attend the Practice and Experience in Advanced Research Computing (PEARC23) Conference in July 2023 in Portland, OR, as part of the Sixth Workshop on Strategies for Enhancing HPC Education and Training (SEHET23) where they presented their experiences as part of the program. Further, the Lab wrote a story about their success, which they proudly took to their school in the fall [1].

As a point of reference, though Growth Sector paid for the students' salaries during the summer, NERSC is currently paying for their salaries for the internship extension.

6 CONCLUSION and FUTURE WORK

We've seen the success of just three students at NERSC, however, there were approximately 45 students in the initial cohort who also had very good summer internships. Laney College will continue this engineering program pending any future funding issues. Early feedback from students at the end of the Laney program shows the following outcomes:

- All 44 out of 45 students completed the summer programs successfully. The one student who did not have family sickness and was required to leave the country.
- 10 out of 45 students were extended by their employers into the fall.
- All students had an opportunity to showcase their work through a presentation or poster session. Three students attended a conference to present their experiences.
- All students reported that they are much more confident coming back to school as a result of the summer program. They are much more motivated with school.
- All students agreed that they would not have been successful without the tutoring or support program provided by Laney.

As an extension of the program, NERSC continues to collaborate with Laney in creating the next step, which is a Data Analytics program. For NERSC, OMNI is the heart of monitoring an HPC data center that collects streaming data from everything in the facility into two streams, one real time to monitor the health of the facility and another for archiving. Visualizing this data is key for the site reliability engineers to quickly "see" what is going on and to quickly diagnose the issues to allow the facility to serve its 7000 global scientists and engineers who use the facility on a 24x7 basis.

Understanding data and visualizing data is important to NERSC now and for future work. Therefore, the next step in their staffing is to create a pipeline for students who can do this work while also being able to fulfill the diversity mission of the Lab.

ACKNOWLEDGMENTS

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DEAC02-05CH11231.

REFERENCES

[1] Elizabeth Ball. *NERSC summer interns feel the thrill of HPC.* Retrieved from https://cs.lbl.gov/news-media/news/2023/nersc-summer-interns-feel-the-thrill-of-hpc/

[2] Elizabeth Bautista, Melissa Romanus, Thomas Davis, Cary Whitney, and Theodore Kubaska. 2019. Collecting, monitoring, and analyzing facility and systems data at the National Energy Research Scientific Computing Center. In *48th International Conference on Parallel Processing:*

Workshops (ICPP 2019), Kyoto, Japan. https://doi.org/10.1145/3339186.3339213

[3] Elizabeth Bautista and Nitin Sukhija. 2021. Employing directed internship and apprenticeship for fostering HPC training and education. *JOCSE 12*, 2. https://doi.org/10.22369/issn.2153-4136/12/2/8

[4] Berkeley Lab. 2022. *Berkely lab workforce demographics*. Retrieved from https://diversity.lbl.gov/berkeley-lab-workforce-demographics-fy2022/

[5] Sarah Bohn. 2014. *California's need for skilled workers.* Retrieved from https://www.ppic.org/publication/californias-need-for-skilled-workers/

[6] Growth Sector.org. n.d. *Growth Sector reimagines the pathway to careers in STEM.* https://www.growthsector.org/

[7] Heather J. Hough and Belen Chavez. 2022. *California test scores show the devastating impact of the pandemic on student learning.* Retrieved from https://edpolicyinca.org/newsroom/california-test-scores-show-devastating-impact-pandemic-student-learning#:~:text=Both%20the%20COVID%2D19%20pandemic,loss%20of%20270%2C000%20students%20statewide

[8] Emad Mushtaha, Saleh Abu Dabous, Imad Alsyouf, Amr Ahmed, and Naglaa Raafat Abdraboh. 2022. The challenges and opportunities of online learning and teaching at engineering and theoretical colleges during the pandemic. *Ain Shams Engineering Journal 13.* Retrieved from https://www.sciencedirect.com/science/article/pii/S2090447922000818

[9] Northwestern Medicine. 2023. *COVID-19 pandemic timeline* Retrieved from https://www.nm.org/healthbeat/medical-advances/new-therapies-and-drug-trials/covid-19-pandemic-timeline#:~:text=By%20March%202020%2C%20the%20World,COVID%2D19%20outbreak%20a%20pandemic

Cybersecurity and Data Science Curriculum for Secondary Student Computing Programs

Richard Lawrence
rarensu@tamu.edu
HPRC[1]

Zhenhua He
happidence1@tamu.edu
HPRC[1]

Dhruva K. Chakravorty
chakravorty@tamu.edu
HPRC[1]

Wesley Brashear
wbrashear@tamu.edu
HPRC[1]

Honggao Liu
honggao@tamu.edu
HPRC[1]

Sandra B. Nite
s-nite@tamu.eduu
HPRC[1]

Lisa M. Perez
perez@tamu.edu
HPRC[1]

Chris P. Francis
chrispeterfrancis@tamu.edu
HPRC[1]

Nikhil Dronamraju
nikhildronam@tamu.edu
HPRC[1]

Xin Yang
karen89@tamu.edu
Medical College of Wisconsin
Milwaukee, WI, USA

Taresh Guleria
taresh@tamu.edu
HPRC[1]

Jeeeun Kim
jeeeunkim@tamu.edu
Texas A&M University
Computer Science & Engineering
College Station, TX, USA

ABSTRACT

Computing programs for secondary school students are rapidly becoming a staple at High Performance Computing (HPC) centers and Computer Science departments around the country. Developing curriculum that targets specific computing subfields with unmet needs remains a challenge. Here, we report on developments in the two week Summer Computing Academy (SCA) to focus on two such subfields. During the first week, 'Computing for a Better Tomorrow: Data Sciences', introduced students to real-life applications of big data processing. A variety of topics were covered, including genomics and bioinformatics, cloud computing, and machine learning. During the second week, 'Camp Secure: Cybersecurity', focused on issues related to principles of cybersecurity. Students were taught online safety, cryptography, and internet structure. The two weeks are unified by a common thread of Python programming. Modules from the SCA program may be implemented at other institutions with relative ease and promote cybertraining efforts nationwide.

KEYWORDS

Outreach, Cybersecurity, Data Science

1 INTRODUCTION

The explosion of computing in everyday life has led to increased interest in teaching computing skills to students of all ages. The projection of information technology and computer science job growth is 13 percent from 2016 to 2026 [11]. Computing education in K12 schools has lagged behind the demand due to lack of resources, including both physical resources and qualified teachers. Previously, the Summer Computing Academy (SCA) at Texas A&M has had success in introducing a variety of computing concepts to secondary school students, especially at the high school level and early college level [4][5]. This program utilizes well reviewed and effective pedagogical techniques to create an effective learning approach. The incorporation of hands-on projects related to the educational topic creates engaging experiences for the students, with the dual goal of internalizing foundational knowledge and driving further interest in the subject.

In addition to the general need for computing education, two computing subfields have been identified as having especially serious unmet workforce needs: Data Science and Cybersecurity. The growth of computing infrastructure has led to an explosion of data volume in multiple scientific and industrial disciplines. In particular, subjects like Genomics in the field of Biology and Machine Learning suffer from a wealth of data. At the same time, the growth of computing infrastructure has led to an explosion of threats, including cyber bullying among child peer groups, cyber crime such as ransomware, and even cyber warfare. It is essential that future efforts to educate students include a focus on these high demand computing subfields.

This paper presents improvements to the Summer Computing Academy curriculum for the purpose of better introducing the topics of Data Science and Cybersecurity to secondary school students. This is beneficial to the students who participate in our summer camps, who develop increased interest in science and computing after participating. In particular, students are made aware of career opportunities in Data Science and Cybersecurity. In addition, the improvements are also of benefit to high school computer science educators who could adopt our methods into their own practice.

[1]Texas A&M University High Performance Research Computing, College Station, TX

https://doi.org/10.22369/issn.2153-4136/14/2/2

In addition, several non-Python-based activities reinforced the cybersecurity concepts: hacking a C program with memory overflow, social engineering to "steal" the instructor's password, and investigating the local structure of the internet. This activity reinforced the importance of protecting personal information with strong passwords and careful consideration of what is posted online.

2 METHODS

2.1 Theoretical Framework

The original basis of the theoretical framework used in the camps came from experiential learning as it was expressed by John Dewey. It applies to formal and informal learning environments and is appropriate for summer camp instruction. It posits that knowledge envelope through experience, practice, and engagement [7]. Many teaching and learning philosophies and frameworks have been built upon this original work. One of those frameworks is the 5-E model of instruction that was originally developed in the field of science education but has been adapted to other STEM education fields. In this model, there is an initial activity to *engage* students in the topic of study. Secondly, students have the opportunity to *explore* ideas and concepts related to the topic at hand. Thirdly, the *explicit* portion involves formalizing the learning with formal definitions of terms, use of formulas, and solution methods. The fourth phase is the *elaborate* phase during which students solve problems and learn more deeply and the topic and applications. Finally, the *evaluation* occurs in order to assess students' learning [3]. In the SCA, students developed knowledge in this way. They engaged in hands-on learning through coding in Python and exploring websites with databases of information related to data science.

2.2 Learning Cycle

To address the need for greater understanding of and interest in careers in data science and cybersecurity, a curriculum consisting of supporting topics and primary topics was interweaved. In addition, it was taken into account that secondary students' ability to focus for longer periods of time is still developing. The first session was a discussion of what data science is all about, with a variety of examples given. Students participated in the discussion as the presenter asked questions to help them discover the use of data science applicable to their lives. This discussion was the initial *engagement* activity for the camp. Students then learned some Python programming that would be the foundational learning as they moved into the next phases. Websites with databases for genomes and DNA strands were provided with the opportunity for students to *explore* various diseases and corresponding DNA strands. Most *explicit* knowledge was gained with additional Python programming and a visit to a DNA processing lab to see the work in action and learn about the research currently underway. The *elaborate* phase flowed directly from the previous phase as students used coding skills and new skills and knowledge of 3D modeling, 3D printing, and visual programming to assemble the electronics from a connection diagram, run the device and observe the functionality. Finally, the *evaluation* of the learning was demonstrated in the team presentation on a topic related to the learning in the camp.

2.3 Curriculum Details

The summer computing academy was a two week program for 7-12th grade students. Each week of five days consisted of four full days and one half day. The two weeks had independent registration, which allowed students to attend one or both weeks. A high fraction of the students attended both weeks. For the purpose of the curriculum, we consider the students who attended both weeks to have attended a single two-week summer camp, because the materials in the two camps were largely unique. Despite the limitation of designing two curricula such that each could stand on its own, the overall two-week summer camp was still highly synergistic. The students who attended both were able to leverage what they learned in the first week and apply it during the second week. This was accomplished by giving the two weeks their own independent themes. The first week had the theme of data science. The second week had a theme of cybersecurity. However, the two weeks were unified by their common use of the Python programming language. Similar but not identical elements of Python syntax were taught in the two weeks so that the students could derive benefit from both, but still be able to participate in the activities even if they only attended one week. Both themes included projects to allow students a hands-on experience. The foundational learning for the summer camp was Python programming language. In order to complete their activities, the students needed the following elements of Python: Variables, Conditionals, Arrays, Dictionaries, and Loops. Other foundational topics were omitted to conserve time.

The first week of the summer camp had a theme of Data Science. For the Python-based activities, the students used the Python libraries NumPy, Matplotlib, Pandas, and Scikit-learn. The exercises were selected to give students a feel for research in data science, including data analysis, visualization, and machine learning. The contents includes how to load different formats of data files, manipulate the data for preprocessing, select best plotting methods for visualization, etc. Students were able to utilize the learned data skills and create linear regression and clustering models for different problems from the Scikit-learn machine learning library. Clustering analysis can be very helpful in cyber attacks detection. For example, Distributed Denial of Service (DDoS) attacks can be proactively detected by clustering analysis of its architecture consisting of the selection of handlers and agents, the communication and compromise [8]. These activities were designed to allow students to internalize the foundational Python knowledge and enhance their learning experience. In addition, several non-Python-based activities reinforced the data science concepts: searching databases for genetic information, 3D modeling and robotics, and a student team presentation on a topic of choice.

The second week of summer camp had a theme of Cybersecurity. For the Python-based activities, the students relied on fundamentals of Python and libraries taught in the first week. The two exercises were selected from the subject of Cryptography: a regular cipher, and the RSA algorithm. The students created their own ciphers in Python from scratch, and used a toy model of RSA also written in Python. Neither used any libraries, but instead reinforced concepts of programming. The cipher is the simpler of the two Cryptography tools. The progression of cipher to RSA provides both a technical and conceptual bridge, since the RSA algorithm is complex and

Table 1: Camper demographics

Gender	M	F				Total
	44	19				63
Ethnicity	A	W	AA	O	NR	Total
	28	23	1	1	10	63
School Type	Pub	Pri	Cha	HS	NR	Total
	42	4	2	5	10	63
Grade Level	8	9	10	11		Total
	2	14	16	12	19	63
Age	13	14	15	16	17	Total
	3	17	13	10	20	63

M = Male; F = Female; A = Asian; W = White; AA = African American/Black; O = Other; NR = Not Reported; Pub - Public; Pri = Private; Cha = Charter; HS = Homeschool

mysterious to younger students. The RSA algorithm is important to introduce to early learners because it is a fundamental technology of the internet, including established protocols such as HTTPS and newer ones such as Blockchain.

Students were also taught how to import Scikit-learn machine learning models to solve image classification problems. Image classification is proven to be very important in cybersecurity. Cybersecurity researchers built machine learning models to detect malware [1] and phishing websites [2] from the images created by applications. In this activity, we used the MNIST (Modified National Institute of Standards and Technology) handwritten digit image dataset [6]. The students learned about image representation, training a machine learning model, evaluating the machine learning model, making predictions, and analyzing the cases when the model makes correct and incorrect predictions on the images. This activity will help the students to be prepared to understand how image classification is applied to detect cyber threats.

3 RESULTS

3.1 Camper Demographics

The students in the camp were diverse in their interests, backgrounds, prior knowledge, and ethnicity. Table 1 provides some demographics of the campers. Some campers attended both camps, but they are represented only once the table. As can be seen in the table, there were only about half as many girls as boys. Most of the campers attend public school, and the vast majority of students were Asian or White.

3.2 Camper Reflections on Learning

Each day students were asked to complete a reflection that asked them to choose just one activity from the day on which to reflect. They were asked why they chose the activity and what they learned from the activity. The three most common activities mentioned were Python programming, visiting the DNA processing lab, and 3D design and printing. It is not surprising that Python programming was mentioned because students knew the camp would be a computing camp, and many high school computing classes use programming languages required by the particular courses. Python is not a prescribed language for advanced placement courses. The DNA processing lab visit tied in very closely with the learning they experienced in the classroom as they learned about data science, and this visit brought it all to life. Some mentioned that they did not expect to do work in this area in the future, but they still chose that activity because of the real-life aspect and connection to the learning activities in the classroom. The 3D design choice was a little surprising in that it was expected that many students would have had this experience previously. However, they would likely not have connected their 3D design object to a microcontroller to control its movement.

4 DISCUSSION

As has been shown in similar camp experiences, the opportunity to learn more about career opportunities in computer science and furthering their knowledge about several topical areas in the field may increase interest in pursuing careers in these fields [9]. In addition to learning more about different career possibilities, students gained introductory knowledge about actually doing some of this work. Gaining knowledge increases confidence, and confidence in an academic area increases the likelihood that a student will decide to pursue a career that involves that knowledge [10]. Students reflected on that knowledge and commented on their enjoyment of learning more about programming in Python and 3D design. Entering college with the confidence that they can be successful in computing or any of the many fields that require some knowledge of computing can increase the interest in a variety of STEM fields of study.

ACKNOWLEDGMENTS

The authors would like to thank staff, student workers and researchers at Texas A&M HPRC, Yang Liu, Michael Dickens, the Laboratory for Molecular Simulation, Dr. Steve Johnson, TAMU Engineering Innovation Center, TAMU IT, TEES IT, TexGen, Division of Research, the Texas Engineering Experiment Station IT, TAMU CPM and TAMU Provost IT for supporting the HPRC SCA program at Texas A&M. We gratefully acknowledge support from the National Science Foundation (NSF) Award OAC #1730695 "Cyber-Training: CIP: CiSE-ProS: Cyberinfrastructure Security Education for Professionals and Students", and NSF Award OAC # 1925764 "CC: Cybterteam: South West Expertise in Training Education and Research", and Texas Workforce Commission award #1622SMP001.

REFERENCES

[1] Irina Baptista, Stavros Shiaeles, and Nicholas Kolokotronis. 2019. A novel malware detection system based on machine learning and binary visualization. In *2019 IEEE International Conference on Communications Workshops, ICC Workshops 2019 - Proceedings (2019 IEEE International Conference on Communications Workshops, ICC Workshops 2019 - Proceedings)*. Institute of Electrical and Electronics Engineers Inc., United States. https://doi.org/10.1109/ICCW.2019.8757060 2019 IEEE International Conference on Communications Workshops ; Conference date: 20-05-2019 Through 24-05-2019.

[2] Luke Barlow, Gueltoum Bendiab, Stavros Shiaeles, and Nick Savage. 2020. A Novel Approach to Detect Phishing Attacks using Binary Visualisation and Machine Learning. In *2020 IEEE World Congress on Services (SERVICES)*. 177–182. https://doi.org/10.1109/SERVICES48979.2020.00046

[3] Rodger W. Bybee. 2014. The BSCS 5E instructional model: Personal reflections and contemporary implications. *The Journal of Computational Science Education* 51 (2014), 10–13. Issue 8. https://doi.org/10.2505/4/SC14_051_08_10

[4] Dhruva K. Chakravorty, Marinus "Maikel" Pennings, Honggao Liu, Xien Thomas, Dylan Rodriguez, and Lisa M. Perez. 2020. Incorporating Complexity in Computing Camps for High School Students - A Report on the Summer Computing Camp at Texas A&M University. *The Journal of Computational Science Education* 11 (Jan. 2020), 12–20. Issue 1. https://doi.org/10.22369/issn.2153-4136/11/1/3

[5] 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

[6] Li Deng. 2012. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine* 29, 6 (2012), 141–142. https://doi.org/10.1109/MSP.2012.2211477

[7] John Dewey. 1938. *Experience and Education.* Macmillan, New York, NY.

[8] Keunsoo Lee, Juhyun Kim, Ki Hoon Kwon, Younggoo Han, and Sehun Kim. 2008. DDoS attack detection method using cluster analysis. *Expert Syst. Appl.* 34 (2008), 1659–1665. https://api.semanticscholar.org/CorpusID:6987397

[9] S. Nite, Ali Bicer, Kimberly Currens, and Rayan Tejani. 2020. Increasing STEM Interest through Coding with Microcontrollers. In *Proceedings of 2020 IEEE Frontiers in Education Conference: Education for a Sustainable Future*. 1–7. https://doi.org/10.1109/FIE44824.2020.9274273

[10] Sandra B. Nite, Devyn Chae Rice, and Rayan Tejani. 2020. Influences for Engineering Majors: Results of a Survey from a Major Research University. In *2020 ASEE Virtual Annual Conference Content Access*. ASEE Conferences, Virtual Online. https://peer.asee.org/34825.

[11] U.S. Bureau of Labor Statistics. 2018. Computer and information technology occupations. Retrieved August 28, 2023 from https://www.bls.gov/ooh/computer-and-information-technology/home.htm

Cybersecurity Training for Users of Remote Computing

Marcelo Ponce
m.ponce@utoronto.ca
Department of Computer and Mathematical Sciences,
University of Toronto Scarborough
Toronto, Ontario, Canada

Ramses van Zon
rzon@scinet.utoronto.ca
SciNet HPC Consortium, University of Toronto
Toronto, Ontario, Canada

ABSTRACT

End users of remote computing systems are frequently not aware of basic ways in which they could enhance protection against cyber-threats and attacks. In this paper, we discuss specific techniques to help and train users to improve cybersecurity when using such systems. To explain the rationale behind these techniques, we go into some depth explaining possible threats in the context of using remote, shared computing resources. Although some of the details of these prescriptions and recommendations apply to specific use cases when connecting to remote servers, such as a supercomputer, cluster, or Linux workstation, the main concepts and ideas can be applied to a wider spectrum of cases.

KEYWORDS

remote computing, cyber-security awareness, training, multi-factor authentication, encryption, secure shell

1 INTRODUCTION

In the last decade, scientific computing, or advanced research computing, has seen a sharp increase in the utilization of computational resources outside of traditional disciplines like the physical sciences and engineering [11, 16]. Nowadays, computational resources are shared with disciplines requiring novel approaches to problems and questions such as digesting and analyzing copious amount of data, simulating models for predicting possible outcomes, and statistically evaluating the support for empirical conjectures. Disciplines such as medical sciences, biological sciences, bioinformatics, machine learning and artificial intelligence have emerged as the heavy users of digital research infrastructures, from computations to storage allocations.

Not so surprisingly, at the genesis of these emerging computational fields, their practitioners were not necessarily savvy or formally trained in technical areas such as programming and high-performance computing. Significant progress and effort in advancing computational knowledge in these fields has been made, although some areas still remain to be improved. In particular, cybersecurity is one area in which not just new scientific computing practitioners but also more experienced ones would benefit from more in-depth awareness.

https://doi.org/10.22369/issn.2153-4136/14/2/3

Furthermore, newer fields such as medicine and biochemistry can bring more sensitive data, such as that collected from individuals, than the more traditional fields. To keep up with a changing environment, such as the increase in working from home, security requirements and best practices keep evolving, and users from all fields will need updated instructions and retraining.

The main goal of this paper is to show what security mechanisms and best practices should be common knowledge for *end-users* (as well as and *the support organization*) when using remote resources such as supercomputers or advanced research computing, based on experiences in training users of the supercomputers at the SciNet HPC Consortium at the University of Toronto [12].

General guidelines on how to remain safe online have been discussed and summarized in multiple publications, e.g. [13]. Similarly, good recommendations on how to strengthen and improve passwords are presented and discussed in [15]. However, it was not until very recently that even specialized organization, such as the National Institute of Standards and Technology (NIST) formally recognized and began a campain to address the issue of cyber-security standardization in High-Performance Computing systems [6].

A second goal for this paper is to be a practical reference for these security techniques. Security is always a moving goal, but we aim to present currently available and appropriate security techniques. We will focus on the following key elements: i) using authentication methods which are more robust and reliable for connecting to remote resources than passwords; ii) concrete practical implementations to be followed when remotely connecting to servers, clusters, supercomputers, or even remote workstations from work, labs, or home; iii) concrete recommendations and tools for users to help protect while working connected to remote systems.

This paper is organized as follows: in Sec. 2.1 we introduce the most relevant and important concepts of cybersecurity, in Sec. 2.2 we explain the characteristics of remote shared resources, in Sec. 2.3 we briefly present the most common type of cyber-attacks currently known. Sec. 3 summarizes SciNet's current training program. To motivate this program and as a reference for future training material, in Sec. 4 we describe guidelines to basic cybersecurity best practices to mitigate some of the main issues presented in the Sec. 2.3, and what role training should play in implementing these practices. Sec. 5 finishes with conclusions.

Additionally, to be able to reflect updates and addendums to best practices, we have created a public accessible repository containing these recommendations, as well as further details and more technical aspects of some of the strategies described in this paper. The location of this repository is https://github.com/cybersec-BestPractices/cybersec-RemoteComputing.

2 SECURITY CONTEXT

2.1 Cybersecurity

The term cybersecurity refers to the different techniques, strategies and methods that can be applied or employed to protect assets and resources against different types of attacks. In particular the "cyber" aspect arises from the fact that the assets are identified as "electronic" or "digital". In many cases, this is data and information stored in digital formats in computer servers and remote machines.

Cyber-attacks, then, are the activities identified as threatening, disrupting, or attempting to gain access to the information illegally, i.e. without authorization.

Many strategies have been developed, and continue to be developed, to protect the confidentiality (i.e. only authorized parties can view the data), integrity (i.e. the data is not unexpectedly modified) and availability (i.e. the data or system is accessible) of digital data or systems. At the same time, cyber-threats continue to grow at a substantial and significant pace [4, 8, 14, 17], both in complexity and number.

Critical to understanding attacks and protections against them, is identifying **vulnerabilities**. Vulnerabilities constitute weaknesses or flaws in systems. Such vulnerabilities can originate from poor designs, oversights of some parts in the systems, uncorrected bugs, or unforeseen use cases.

No system can be guaranteed to be 100% attack-proof, and very stringent approaches could come at the the expense of usability. Because of these reasons, the risks and severity of a breach must be weighed against the cost of protective measures and the impact to usability[1]. For remote shared computing systems meant for academic research, this balance will have a different outcome than for e.g. an online banking site.

2.2 Using a Remote Shared Computing Resource

We need to discuss a few characteristics of Advanced Research Computing (ARC), or High Performance Computing (HPC), facilities[2] before we can introduce what specific threats mean in that environment. We should also note that for the goal of this paper, we will consider ARC and HPC systems as

First of all, access to such systems is usually remote, which means a connection needs to made over the internet. The internet is the paradigmatic example of connectivity. Its confluence of heterogeneous systems also opens up vulnerabilities. At its origin, the internet was a group of mutually trusting entities attached to a transparent network, and not designed with much security in mind. One's default attitude should be to not trust what is on the internet. This particularly applies to websites, where directly or indirectly visiting a fake websites may result in direct exposure and potential attacks to the connecting devices.

The most vulnerable element here is the user and their behaviour. Beginning users of HPC systems may be aware of security concerns using web-based authentication and access, but access methods to remote HPC resources are often unfamiliar. Without training they will not know best practices nor how to remain vigilant in using these systems.

One of the most common methods of connection is via ssh, which stands for "secure shell". In this context, let us call the computer from which the users logs in the "local" computer, and the HPC facility the "remote server". To start the connection from the local computer to the remote server requires the user to authenticate. For a long time, authentication would be based on a username and password, but that is no longer a best practice. While the connection is active, data flows between the user's computer and the remote facility. Ssh connections provide encryption of this data flow.

Additional features of ssh that are common, but have security implications, are support for graphics windows using X forwarding, port forwarding for reaching hosts inside the remote facility, and key-agent forwarding to facilitate authentication to other facilities from the first remote facility.

The remote setup also means that one should be concerned with the security of both the local computer and the remote server. If the local computer is itself a shared computer, for instance, one that is present in a research laboratory, that can pose additional concerns.

The remote server is a shared system, usually running a flavor of GNU/Linux or UNIX as the operating system. Such operating systems make a distinction between privileged users and regular users, sort users in groups, and maintains ownership and group membership of files and running programs, that can and should be used to control access to files and commands, as there are typically many (regular) users logged in.

2.3 Types of Cyberattacks and Cybersecurity Threats

There are many types of cyberattacks, ranging from specially targeted and designed attacks to more generic and opportunistic ones, such as the so-called *zero-day exploits*, in which attackers take advantage of a vulnerability for which a patch has not been developed yet.

In this section, we will review some of the most common types of attacks and the impact they could have on users and services. While they could utilize many different approaches, one could classify them in two basic categories: one where the attack's goal is to impact availability (e.g. by bringing down a particular functionality in a system); another class of attacks where the goal is to impact confidentiality and/or integrity (e.g. attempt to gain access to unauthorized resources, such as systems, privileges or/and users accounts). The techniques, tools and best practices to mitigate these different types of attacks, both by the ARC service providers as well as their user, will be discussed in section 4.

We can distinguish several objectives of cybersecurity attacks:

a. Get past authorization to get access to a system ("Hacking");
b. Disable a service (e.g, by "Denial of Service attack");
c. Steal secure information (e.g., through "Phishing");
d. Install software on a system that can be used later for later attacks (e.g. "Malware infection").
e. Abuse resources (e.g, Cryptocurrency mining when this is against a "Terms-of-Use" agreement)

Cybersecurity attacks usually have several of these objectives. Most of these attacks are crimes in many jurisdictions [7].

[1] This is usually done employing the so-called *cybersecurity matrix* – see our repository for more details about cyber-security.

[2] We should also note that for the scope of this paper, ARC and HPC systems will be considered equivalent.

In a so-called *brute force attack*, an entity will attempt to get access to a system by systematic attempts to guess user credentials to authenticate in the targeted system, e.g. a username and password. Nowadays, brute force attacks often rely on advanced tools to try many different passwords (not just 'guessing' which suggests a manual process, in which the attacker may know something about the victim). Brute force attacks are still quite effective despite the existing controls to prevent them. According to the 2017 Varonis' data breach report [10], '5% of confirmed data breach incidents in 2017 stemmed from brute force attacks'.

Once access to a service has been gained, the consequent risks depends on what authorization the user has whose credentials were obtained. Regular users would only have access to their own data, or to any data shared with them, and the impact of the breach could be limited to just their account. Administrators and staff may have elevated access, and having their accounts hacked would be much more dangerous and impact several users or even the whole system. It should also be noted that unintended security vulnerabilities in the software used in a service or its operating system might make it possible for regular users to gain administrative powers.

Another common type of attack is the so-called *denial of service* (DoS), in which an attacker would attempt to bring down –partially or completely– a system or network. There are different methods in which this can be done. A common method is by flooding with traffic a given system so that it saturates its resources or even the bandwidth of the network.

Since frequent traffic from a single IP could easily raise flags and be stopped, a more elaborate version of this type of attack involves launching DoS attacks from various, distributed servers. This case is referred to as a "distributed"-DoS or DDoS.

Malware is a general term used to refer to any type of malicious computer programs. There are different types of malware, among the most "famous" ones are: viruses, worms, ransomware, Trojan horses, rootkits, etc. Its goal can range from making a system unresponsive, steal information (credentials, documents, etc.), "kidnap" information, espionage, use a connection point to jump to another systems to hide the trace of an attack, etc.

Malware can find their way onto a computer e.g. as part of other software packages, which may have gotten installed as part of a packages, or be installed unintentionally by cleverly disguised website. While they usually target users' personal computing devices because they have administrative permissions on them, servers can also be infected in the form, for example, of so-called root kits.

Sniffing, IP/DNS Spoofing, and Man-in-the-Middle attacks are a collection of techniques aimed at getting information out of the data that is transmitted. Sniffing refers to collecting the packets of information while it is transmitted through the network. When using encryption, the information in the packets themselves can't be read, if the encryption is strong enough; this is why the type of ssh key matters.

IP Spoofing refers to a technique where a malicious party attempts to inject information into the network as if it came from other system, e.g., the actual remote server. The objective of this attack would be to convince the user or other systems that the message comes from a valid source and in this way establish an exchange of information. In this way, credentials or other sensitive information could be obtained. In combination with a DoS or DDoS attack, this attack can be used to disguise and redirect network traffic to malicious and bogus sites.

Man-in-the-middle is a term used to describe a third party that is attempting to eavesdrop or intercept information sent between the user and the remote system. This can happen in different ways, for instance, at a physical level, where a device or connection can be added to the main communication channel; or, at a "software" level, where similarly a program can be employed to intercept and steal the data shared within the communication.

One element which is often essential in cyberattacks is the "human factor" which involves taking advantage of certain characteristics in standardized human behavior by tricking people to divulge sensitive or private information, in order to obtain access to systems or steal information. This type of attack is commonly known as "social engineering". The level of sophistication can vary from very generic to more targeted and specialized.

Phishing techniques are a subcategory of social engineering attacks. They relate to illegitimate emails attempting to acquire *sensitive data* by exploiting the victim's inexperience and trust. They are a very popular mean to obtain credentials, and from there hack into an organization.

Any of these (illegal) cyber attacks can be hard to detect, and hard to fix once the damage is done. The best approach is to try to make such attacks less likely to suceed. The best approach for protection depends on the type of attack.

3 SECURITY TRAINING PROGRAM

The need for training for various forms of user training will be explained in the next section, but for clarity, it is worth to point what security-related training SciNet has offered to its users so far. These courses can be found on https://education.scinet.utoronto.ca by searching for the course codes given in the parentheses below.

Intro to SciNet, Niagara, and Mist (HPC105)
Typically given as a single session of 90 minutes, this presents the details of logging in into the Niagara and Mist clusters (including using ssh and keys), available file storage, and creating and submitting jobs to be the schedule.

Intro to the Linux Shell (SCMP101)
This 3-hour workshop familiarizes new users with the Linux shell, which is the main interface to our systems.

Introduction to Supercomputing (HPC101)
Either given as one session of about 3 hours, or in 3 separate sessions, this workshop shows why (remote) clusters are used, as well as common ways people use it.

Securing File Access Permissions on Linux (SCMP283)
This workshop is aimed to educate users about what permissions are, how to use available tools to control access and sharing, and how to avoid common security pitfalls.

Introduction to Apptainer (SCMP161)
This workshop introduces users to Apptainer, a container solution, which could be used for software that requires a specific OS setup different from what the cluster uses, or to handle workflows with many files, or for enhanced security.

Enable Your Research with Cybersecurity (SCMP183)
A workshop of 4.5 hours given over the span of three days, that covers various aspects of cybersecurity, cyberattack models, and best practices. Also covers cybersecurity in the context of human research data and the Research Ethics Board.

Advanced Linux Command Line (SCMP271)

Week ending on:	Users using keys
Oct 10, 2021	38%
Oct 17, 2021	43%
Oct 24, 2021	47%
Oct 31, 201	50%
Nov 7, 2021	59%
Nov 14, 2021	62%
Nov 21, 2021	67%
Nov 28, 2021	76%
Dec 6, 2021	76%
Dec 12, 2021	86%
Dec 19, 2021	82%
Dec 26, 2021	79%
Jan 2, 2022	80%
Jan 9, 2022	78%
Jan 16, 2022	89%
Jan 22, 2022	100%

Table 1: Week-by-week progress in promoting ssh keys over password authentication to log into SciNet's Niagara cluster. Password access was disabled on January 22, 2022.

Bash command line with common idioms (SCMP281)
Both of these are workshops that allow users to further their Linux skills.

SSH Keys Drop-in Session (SCMP110)
In the ssh keys pilot (Oct 2021-Jan 2022) in which password authentication was replaced by ssh keys on SciNet systems, several drop-in sessions were held to help users set up their ssh keys.

In our cybersecurity training, as in most of our training, the strategy is not to only provide general information, but to offer courses that combine such information with concrete practical exercise either done during the sessions or reviewed and graded with feedback afterwards. In addition, we offer support for users after these sessions. This post-training support makes it much more likely that the security practices are indeed followed, and also makes it possible to enforce some practices.

For security training it is possible to assess its success by aggregating how users use the system. For instance, during the ssh keys pilot, we tracked the number of users using keys, and saw it steadily rise as we gave a training session on the upcoming changes (see Table 1).

While in-person training is often preferable so that issues on users' own laptops can easily be addressed, nonetheless, because the systems hosted and operated by SciNet are used by researchers throughout Canada as well as their collaborators outside of Canada, most of these training sessions were online.

SciNet's courses are open to all users of the Canadian national ARC facilities as well as to members of Canadian academic institutions. Beyond this target group, all materials of SciNet courses are freely available online to anyone on https://education.scinet.utoronto.ca.

4 CYBERSECURITY BEST PRACTICES

There are several lines of defense that the provider of the ARC service should establish proactively.

- Securing authentication: This involves verifying the identity of an entity, user, process, program, server, etc.
- Protecting authorization: Restricting access to certain users, based on their identities and qualifications serves the purposes of preserving the privacy and confidentiality of the data; examples of implementations are role-based access controls.
- Using encryption: Particularly when transferring data, but sometimes also required for data as it is stored in the ARC center.
- Integrity checks: This is quite relevant in order to guarantee that the data that was sent has not been tampered with and is trustworthy. On an operating system level, it is important to check that no system executable are changed and replaced by malware. There are different techniques to implement integrity checks, some of the most common ones include digital signatures or checksum calculations.
- Network filtering: This is to limit traffic coming in and out of a system or security perimeter (e.g. firewalls). Implementations can be done at the software or hardware level.

While the aforementioned approaches are put in place by the ARC center, many users may need to know about them to understand if they are allowed to use that system for their data. In traditional ARC systems, the responsibility for data access control was often left to the users. For some specific types of data, stricter guarantees are needed that require more measure from the ARC center, and various auditable certification levels exist [9].

In addition to measures put in place by the ARC center, users also have a responsibility to protect against cyberattacks, because attacks often do not start on the remote system, but on the end-users local computer. We will present several specific strategies for end-users below.

4.1 Software Updates

The most basic and immediate way to improve the security of a user's local computer is to keep that system's operating system (OS) and programs update to date. Many times attackers will take advantage of systems which are not up-to-date with the latest releases or security patches for the system, and gain access by exploiting *vulnerabilities* that could have been mitigated by a simple OS or application update. Should the end user's workstation be compromised because their machine was not patched in a timely fashion, it could also lead to the compromise of the remote computing system they are connecting to. Therefore, by keeping their systems (desktop, laptop, etc.) up to date, the end user also helps protect remote computing systems.

The best practice for users, as well as for center staff the workstations, is quite simple, keep your systems up to date!

4.2 Antivirus & Malware

Computer viruses and malware are potential high-risk entry vectors to local computers and through those, to our machines. As such, having *antivirus* software installed and running on local computers is critical. Users should be encouraged to check with their university IT department or library, which usually provide licenses for students and staff to get antivirus products.

Antivirus software can detect malware signature's from a database which is updated on a regular basis, but have started to use machine learning techniques to identify unknown or file-less malware which was not previously detected.

4.3 Authentication enhancements with ssh

A very common protocol for connecting to a remote server or system is `ssh`. `ssh` stands for **s**ecure **sh**ell. It creates an encrypted channel between the client (user trying to connect) and the server (system where the user wants to connect). While `ssh` offers a secure way to connect between computers, it can be vulnerable to some of the attacks described in Sec. 2.3, such as man-in-the-middle attacks or brute-force attacks. We will not go into the details of how `ssh` creates this secure communication channel but we will instead focus on the mechanism to authenticate the user in the remote system, as it plays a key role in mitigating the risk of attacks against ssh.

4.3.1 Passwords. At the moment, the most common way of authenticating in ARC systems is by using a username and password. Passwords may give users the illusion of protection, but they are one of the least secure authentication methods. Passwords can be compromised, can be weak, can be stolen, and of course are in most cases chosen by humans – who arguably can be considered the weakest element in the cybersecurity chain.

When there are better alternatives, as the ones mentioned below, they should be used. But many authentication methods still rely on passwords utilization. Whenever this is the case, an additional tool to consider to use is a password manager that stores passwords encrypted. In addition, password managers can help to organize and even validate or check the strength and integrity of passwords. Depending on the OS there are different options available, a couple of open source options are: `KeePassXC` (https://keepassxc.org) and `bitwarden` (https://bitwarden.com).

The main risk with password authentication is the ability for an attacker to obtain these credentials. Since the password needs to be transmitted to the remote site, there is a possibility that it may be intercepted.

4.3.2 `ssh` *keys.* A generally more secure and efficient way to authenticate users with a remote system that does not suffer from the vulnerability that passwords have, is to use *keys*.

The authentication via keys leverages asymmetric encryption. It involves two keys which are part of a key pair: one *private key* which must be kept secure, and one *public key*, which can be distributed. The public key is used to encrypt data, which can be decrypted with the corresponding private key. After the establishment of the ssh connection, the user's authentication occurs. The remote server sends an encrypted challenge request, encrypted with the public key, to the client. The client then decrypts the challenge request with the private key, and sends it back to the remote server. Then, the remote server compares the two pieces of information (the challenge request, versus the challenge response by the client), and if they match, the authentication of the user via ssh keys is successful. It is important to note that these steps are transparent to the user. Also, the private key never leaves the client, making this method of authentication more secure than the authentication with password.

The process of starting to use an asymmetric keys pair for `ssh` can be summarized as follows:

(1) Create an SSH key pair on your **local** machine – on a Linux, Mac OS or even Windows using MobaXterm or Linux-subsystem terminal, this can be done using the following command:

```
ssh-keygen -t ed25519
```

When this command is executed, it will prompt for the *location* where the keys are going to be placed and for a *passphrase* to associate to the keys. The passphrase is like a password local to your computer; its purpose is to encrypt the private key to better protect it against potential theft. After these two pieces of information are entered, the command will create a pair of files to be located at the location specified previously –its default location would be `$HOME/.ssh/`, where `$HOME` represents the user's home-directory–. If no further details are given, the files are by default named as `id_ed25519` and `id_ed25519.pub`, representing the private and public keys respectively. The `-t ed25519` used in the `ssh-keygen` command represents the type of algorithm used to generate the encrypted keys and this one in particular is one of the recommended standards to be used nowadays.

(2) The next step is to transfer the file with the **public** key to the remote server/machine. Different remote facilities will have different mechanisms for this. On some, one can use the following command from the local computer:

```
ssh-copy-id  -i $HOME/.ssh/id_ed25519.pub
    USERNAME@remote.system.ip
```

where the `-i` flag indicates the public key file (in the example the default one located at `$HOME/.ssh/id_ed25519.pub`) to be copied over the remote system –`remote.system.ip`– to which the user `USERNAME` would like connect to. At this point the user will still be asked to authenticate itself, by entering its username/password combination.
On other systems, there may be a web interface that allows users to upload the public keys. That site itself may use passwords, perhaps combined with MFA (see further down).

(3) After these two steps, unless the default location was used for storing the private key, any time one uses the ssh command, it must be told where to find this key with the `-i` flags.

At this point some few observations should be done:

- Having a combination of private/public-keys guarantees that only a machine where the private key can be found can connect to the remote location where the public key resides. Hence why is critical that the private key **never** leaves the machine where the keys were generated, this also includes not copying them to any other machines.
- There should be an unique set of keys per machine. In other words, if a user owns a laptop, a desktop and a workstation, the procedure described above should be repeated independently in each of these devices. One may also want to generate separate keys for each remote system, if one wants each trusted relation to have a unique key pair.

- The private key should always be protected with a passphrase, ie. never leave an empty passphrase! When not specifying a passphrase, the private key remains unencrypted. If someone gains access or takes control of the local device, they will be able to connect to the remote system.

Many more details can be added to the process of keys generation. These details are presented and discussed in our repositor github.com/cybersec-BestPractices/cybersec-RemoteComputing.

We should note that `ssh` keys themselves are also prone to brute force attacks if the length of the key is too short and/or the algorithms used are deprecated. The National Institute of Standards and Technology (NIST) in the US, has developed a series of reports [1, 2] describing the recommended standards to use for keys encryption algorithms along with keys' length (see Table 2). It is advisable to stick to these NIST standards in order to minimize the risk of brute force attacks.

Encryption Algorithm	Key length	key generation command
ECDSA, EdDSA, DH, MQV	224−255 (and above)	`ssh-keygen -t ed25519`
RSA	2048 (or above)	`ssh-keygen -t rsa -b 4096`

Table 2: NIST's standard recommendations for `ssh` keys encryption algorithms [1, 2].

Theoretically, *quantum computers* would be capable of breaking the cryptographic algorithms that are used in ssh. Although it is not clear how soon quantum computers will be powerful enough to do so, in response to the developments in quantum computing, NIST has already began the preparation for the so-called *"Post-Quantum Cryptography Standardization Process"* [5].

As was mentioned above, the `ssh-copy-id` may not work on some systems, particularly on those sites that have further enhanced the ssh-key mechanics with a *centralized SSH-keys* database. The CCDB of the Digital Research Alliance of Canada provides such a capability for the national ARC systems in Canada.[1]. Users can upload their public keys which will then be used in multiple remote systems. This will happen transparently for the users, as the underlying infrastructure will take care of propagating the information across the different systems.

On one of these national systems, the Niagara cluster at SciNet, centralized ssh keys are the only mechanism of authentication, Following the pilot of about four months (see Sec. 3).

Because there is a learning curve to using ssh keys, and because the methods of setting it up depends on the local operating system and ssh clients, a combination of incentives (such as brownout periods –i.e. periods with a reduction or restriction in how users could connect to the system– using password authentication) with training and drop-in sessions has helped deliver a smooth adoption on Niagara.

[1] https://docs.alliancecan.ca/wiki/SSH_Keys

4.4 More secure authentication with MFA

Multi-factor authentication (MFA) is a widely utilized security measure in various technological domains, aimed at ensuring additional layers of authentication. Users may be familiar with implementation of MFA in mobile devices, where biometric factors such as facial recognition, iris scanning, and fingerprint sensors are utilized to authenticate users. But MFA is a more general technique that enhances secure authorization by require multiple separate pieces of authentication, called factors.

The main benefit in security is to go from a single factor to two factors, i.e., to have two-factor authentication, also known as second-factor authentication (2FA). The benefit of adding more authentication factors to the same service tends to be marginal.

There are multiple and diverse MFA mechanisms and implementations, some of which are open-source and free, while others are commercial. Among the more popular choices are time-based approaches which generate a one-time-password (OTP) to use when authenticating. Such a code can only be used once and for a short and specific period of time. This concept is also used in commercial services, like telephone companies or financial institutions, to validate their users credentials by contacting them on their phones as a second way to authenticate their identities. It is also possible to use this form of MFA in combination with ssh using the open-source GoogleAuthenticator (https://github.com/google/google-authenticator), or PrivacyIdea (https://www.privacyidea.org/) which is another free open source initiative. Alternatively, there are commercial solutions, such as Duo. These commercial solutions offer support for pushing an authorization request to a user's cell phone or accepting hardware devices like YubiKeys.

Academic institutions are increasing adopting MFA as well for authenticating. Because many ARC centers like SciNet serve users from several institutions, they require their own MFA implementation, but if it is the same solution, users may be able to reuse the same app. SciNet has used Google Authenticator as an optional MFA method for users (and a mandatory one for its staff) since June 2020. After having taken part in several pilot projects for MFA across the Canadian national ARC systems under the umbrella of the Digital Research Alliance of Canada, has transitioned to using Duo in May 2023.

While users could add MFA for their computers, e.g, using one of the open source solutions, if their computer is not accessible from the wider internet, the security benefit seems small. But if MFA in connecting to their ARC facility is available, they should be encourage or required to do so.

4.5 VPN

An additional layer of protection that users can add when working or connecting remotely is to use a *Virtual Private Network*. The main objective of this type of technologies is to extend the domain of a private, secure, controlled networks beyond the physical limits that would usually define such a network. VPN offers a secure, encrypted connection over a shared network. A typical example is a VPN offered by an academic institution, which would allow their students and personnel to remotely connect to its network as if they were on campus. This offers multiple advantages, such as having an IP address assigned within the domain or range of

IP addresses within the academic institution, additional protection against undesirable Internet traffic or malicious agents. It is also possible to engage with private providers of VPN services, although we would encourage users to inquire with their IT departments and libraries within their corresponding academic institutions is such a service is available.

VPNs can also be a means to mitigate the security risks of other methods of connecting. For instance, the employment of Virtual Network Connections (VNC) is a common practice when working remotely. It offers the remote user a great deal of flexibility and much more responsiveness in what it refers to graphical interfaces, than other possible counterparts like X-forwarding over `ssh` connections. However there are a couple of elements that are usually considered risky in terms of security: many VNC systems allow for users to connect without the use of a password, which needless to say is a highly discourage practice! Secondly, VNC works by opening connections through a given port in a server, these connections –which by design are resilient– should be tear down when not used to reduce the chance of ports swiping by a malicious party. In many cases, specially in supercomputer centers, where resources may not be directly exposed to the Internet, the best way to reach a service like this is by *tunneling* through the so-called login nodes.

4.6 Further Protection against cyber-attacks

As protection against brute force attacks, ARC centers should have controls in place that detect repeated authentication failure attempts, resulting in the application of a banning policy. For instance, some approaches will ban users from accessing the system for a period of time, or lock their account and request a mechanism to unlock it using another mean to authenticate the user, eg. email or SMS.

Limiting the number of connections per minute mitigates much of the brute force attacks, but there are ways that end users can further mitigate the risk of brute force attacks by choosing longer usernames (the number of possibilities to try for short usernames is small) Similarly, users should avoid having simple, repeated/reused, or short passwords, or even better avoid using passwords at all by substituting them with ssh keys. Additionally, private keys should be protected with a strong passphrase, and never leave the local computer. Note that, ideally, all this should be taught even before the user accesses the remote server for the first time.

To mitigate denial-of-service attacks, frequent subsequent authentications, successful or not, will trigger a banning policy on the originating IP. It is important to inform users of this limitation, and to discourage connecting many times per second or transferring many separate files instead of combining the files into one zip or tar archive files and transfer that. The IP-based banning (even if temporary) can be quite disruptive for research labs where the local computer is shared, or in which the local computers share a single outbound IP.

ARC center can be expected to mitigate the risk of malware with configuration management, restricting root access, a rootkit scanner, etc. End users of remote computing systems have a role to play in protecting themselves against malware by running antivirus software and malware scanners, even on Macs and Linux computers. It is advisable for users to have separate machines for private and research, if they can. On their local computer, encourage them not to blindly click yes on popup windows, and to look at all warnings, errors and messages.

There are different ways for ARC centers to mitigate man-in-the-middle attacks, such as encryption, implementation of integrity checks to verify that the data has not been manipulated, etc. While most controls to protect against such attacks fall under the responsibility of the administrators of the remote computing systems, here again the end user has a role to play.

An example of this, is `ssh` trying to warn users about possible MITM attacks when checking for the "fingerprint" codes of known systems when these change in comparison to previous connections or sessions. The end user should be particularly vigilant if such warnings are displayed in their terminal. They should be aware of these fingerprint codes and their actual values in order to verify the authenticity of the servers one would be connecting to.

Thus, ARC centers should advertise the fingerprints.

One of the best ways to prevent phishing attempts is to educate users! [17]. Check with your university's IT department or library, in general these departments have resources available to instruct and educate users in how to recognize phishing attempts. For example, as many other institutions, the University of Toronto has collected some examples at https://securitymatters.utoronto.ca/category/phish-bowl/.

In many cases, some attacks could happen without the victim being even aware of it. A typical example is users whose accounts have been compromised and are just being used as "trampolines" to jump to other systems. In other cases, attackers may just want to gain access to computational resources in order to have more compute power at their disposition and for instance, mine cryptocurrencies. These examples have been detected multiple times in different supercomputer centers or systems which offer substantial amounts of computational resources [3]. Some simple strategies in order to mitigate this unnoticed-driven and subtle abuse of users' accounts, is for systems to inform the users about details on their connections; e.g. when and from where were their last connections to the system, or even more sophisticated ones such as, keeping track of the usual pattern of connections for users (e.g. IP, geographical location, etc.). The end users should pay attention to these details (which are generally provided at the beginning of the session, in the banner message) and confirm that the activities belong to them. When certain irregularities are detected, they should report the anomaly to the administrators of the remote computing systems.

4.7 Containerized Solutions

Containers have been quickly gaining popularity in the last few years, as their approach offer a simple and robust solution to installing software with multiple or complex dependencies. Containers are also used to isolate resources and services, and in particular can be a great solution to mitigate the escalation of security risks by differentiating and separating into multiple containers. For instance, in the case of an application deploying an attack within a container, this can still shield it from the host; similarly an attack on the host could be shielded from reaching hosted containers; as well as inter-containers attacks. Among the most popular solutions are

apptainer[1] and *docker* containers. Apptainer containers are usually recommended over docker ones due to security concerns – mainly due to the fact that docker images require access to root privileges presenting a potential high-risk liability. Similarly users employing already prepared images should be aware of the risk of utilizing ones coming from untrusted sources.

5 CONCLUSIONS

In this paper, we have presented a basic overview of the most typical forms of cyber-threats in using remote computing facilities. We discussed several useful techniques that end-users can leverage to mitigate some of these attacks. Some of these techniques are well-known and commonly used by professionals in the disciplines of computer science and systems administration. However, many end-users with backgrounds in diverse disciplines may need training in these (for them) novel techniques. Having put them in the context of the risks and impacts, we believe will increase the general awareness and at the end benefit the whole community of remote-systems users.

Users of cloud services should also be concerned about security and privacy risks. Although this paper focused on using traditional ARC clusters, at the very fundamental level, all what is discussed in this paper and the techniques presented and recommended here will still be applicable to this type of systems too. Remote connectivity using ssh and its improved forms, such as keys and MFA, should be used, as well as any other form of enhanced connectivity. Nevertheless there are a few elements that may be different from our previous discussion. For instance, if the user is responsible for deploying, installing, configuring, administrating and maintaining its own machines and environments in the cloud infrastructure, then special considerations should be given to the OS installation, permissions and privileges, allowed services running in the remote machine, open ports, etc. If the cloud service will be used as a sort of web-portal or gateway, additional attention should be paid to web services running on the machine, tight all possible access points and methods, as well as, being compliant with certificates and protocols standards. It is always recommendable that if the end-user is not familiar with this type of configurations, to request support from specialized personnel such as system administrators or technical support to check on all of these.

It is critical when using shared resources and accessing them remotely, to realize that the system as a whole is as weak as its weakest element. Hence why considering the implementation of the combined techniques and strategies presented here is highly recommended to improve the overall cyber-security posture of the remote system and local users connecting to it.

As a final remark, we have created a repositor github.com/cybersec-BestPractices/cybersec-RemoteComputing where we aggregate and present most of the best practices, concepts and implementation details presented in this work.

We decided to present this information in this way, so that it can be updated as technology, trends and threats change and advance.

At the same time, we allow users to use this as a consolidated reference, contribute, keep track of changes. We additionally enable issues requests for users and readers to ask questions or make comments. Similarly we have enabled the wiki feature to allow for users' contributions – which of course will be curated by the authors and collaborators.

[1]Previously known as *singularity*.

ACKNOWLEDGMENTS

We would like to thank the SciNet team for their involvement in the cybersecurity program, and in particular Raphaëlle Gauriau for promoting cybersecurity and cybersecurity awareness at SciNet. We would also like to thank Michael Nolta for helping to collect the numbers in Table 1.

MP acknowledge financial support from UTSC's PPG Grant and UTSC/CTL's Professional Development Fund.

The SciNet HPC Consortium is funded by Innovation, Science and Economic Development Canada; the Digital Research Alliance of Canada; the Ontario Research Fund: Research Excellence; and the University of Toronto.

REFERENCES

[1] Elaine Barker. 2020. *Recommendation for key management: Part 1 - General.* National Institute of Standards and Technology, U.S. Department of Commerce. https://doi.org/10.6028/NIST.SP.800-57pt1r5

[2] Elaine Barker, Allen Roginsky, and Richard Davis. 2020. Recommendation for Cryptographic Key Generation. (June 2020). https://doi.org/10.6028/NIST.SP.800-133r2

[3] bbc. 2020. Europe's supercomputers hijacked by attackers for crypto mining. (2020). https://www.bbc.com/news/technology-52709660

[4] C Beek, T Dunton, J Fokker, S Grobman, T Hux, T Polzer, M Rivero, T Roccia, J Saavedra-Morales, R Samani, et al. 2019. Mcafee labs threats report: August 2019. *McAfee Labs* (2019).

[5] G. Alagic et al. 2022. Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. (September 2022). https://doi.org/doi.org/10.6028/NIST.IR.8413-upd1

[6] Y. et al. Guo. 2023. High-Performance Computing (HPC) Security: Architecture, Threat Analysis, and Security Posture. (February 2023). https://doi.org/doi.org/10.6028/NIST.SP.800-223.ipd

[7] ICLG. 2023. Cybersecurity Laws and Regulations 2023. (2023). https://iclg.com/practice-areas/cybersecurity-laws-and-regulations

[8] Martin Jartelius. 2020. The 2020 Data Breach Investigations Report – a CSO's perspective. *Network Security* 2020, 7 (2020), 9–12. https://doi.org/10.1016/S1353-4858(20)30079-9

[9] U.S. Department of Defense. 2023. Cybersecurity Maturity Model Certification. (2023). https://dodcio.defense.gov/CMMC

[10] Jeff Petters. 2021. What is a Brute Force Attack? https://www.varonis.com/blog/brute-force-attack Accessed Feb. 15, 2022.

[11] Marcelo Ponce, Erik Spence, Ramses van Zon, and Daniel Gruner. 2019. Bridging the Educational Gap between Emerging and Established Scientific Computing Disciplines. *The Journal of Computational Science Education* 10, 1 (Jan 2019), 4–11. https://doi.org/10.22369/issn.2153-4136/10/1/1

[12] Marcelo Ponce, Ramses van Zon, Scott Northrup, Daniel Gruner, Joseph Chen, Fatih Ertinaz, Alexey Fedoseev, Leslie Groer, Fei Mao, Bruno C. Mundim, Mike Nolta, Jaime Pinto, Marco Saldarriaga, Vladimir Slavnic, Erik Spence, Ching-Hsing Yu, and W. Richard Peltier. 2019. Deploying a Top-100 Supercomputer for Large Parallel Workloads: the Niagara Supercomputer. In *PEARC '19: Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning).* 1–8. https://doi.org/10.22369/issn.2153-4136/10/1/1

[13] Danielle Smalls and Greg Wilson. 2021. Ten quick tips for staying safe online. *PLOS Computational Biology* 17, 3 (03 2021), 1–6. https://doi.org/10.1371/journal.pcbi.1008563

14 Internet Society. 2018. Cyber incident & breach trends report. (2018).

15 Joshua Tan, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2020. *Practical Recommendations for Stronger, More Usable Passwords Combining Minimum-Strength, Minimum-Length, and Blocklist Requirements.* Association for Computing Machinery, New York, NY, USA, 1407–1426. https://doi.org/10.1145/3372297.3417882

[16] Ramses van Zon, Marcelo Ponce, Erik Spence, and Daniel Gruner. 2019. Trends in Demand, Growth, and Breadth in Scientific Computing Training Delivered by a High-Performance Computing Center. *The Journal of Computational Science Education* 10, 1 (Jan 2019), 53–60. https://doi.org/10.22369/issn.2153-4136/10/1/9

[17] Verizon. 2021. Verizon 2021 Data Breach Investigations Report. (2021). https://www.verizon.com/business/resources/reports/dbir/2021/masters-guide/

Assessing Shared Material Usage in the High Performance Computing (HPC) Education and Training Community

Susan Mehringer
Center for Advanced Computing
Cornell University
Ithaca, New York
shm7@cornell.edu

Kate Cahill
Ohio Supercomputer Center
Columbus, Ohio
kcahill@osc.edu

John-Paul Navarro
University of Chicago
Argonne National Lab
Naperville, Illinois
navarro@anl.gov

Scott Lathrop
University of Illinois
Shodor Education Foundation, Inc.
Urbana-Champaign, Illinois
lathrop@illinois.edu

Charlie Dey
Texas Advanced Computing Center
Austin, Texas
charlie@tacc.utexas.edu

Mary Thomas
San Diego Supercomputing Center
University of California San Diego
La Jolla, California
mpthomas@ucsd.edu

Jeaime H. Powell
Texas Advanced Computing Center
Austin, Texas
jpowell@tacc.utexas.edu

ABSTRACT
This paper shares the results of a survey conducted October-November 2022. The survey's intent was to learn how the community both shares and discovers training and education materials, whether those needs were being met, and if there were interest in improving how materials are shared. The survey resulted in 112 responses primarily from content authors who are, or support, academics. While the majority of respondents considered themselves successful in finding materials, most also encountered barriers, such as finding materials, but not at the needed depth or level. Most respondents were both interested in, and able to, work toward community efforts to improve finding materials, with most citing lack of staff time as a barrier to doing so. Proposed efforts in community engagement to work toward these efforts are discussed.

Keywords
Survey, education, training, community engagement.

1 INTRODUCTION
The use of computing technologies is rapidly expanding in many sectors, necessitating access to high-quality education and training materials to facilitate research computing. The demand for instructional materials, encompassing a wide range of topics related to the development and application of research computing technologies across disciplines, is crucial for both formal classroom settings, informal training, and self-paced learning.

One way to meet this need and keep up with the ever-evolving landscape of HPC educational and training material development is to improve how the community shares and finds materials. In order to gauge the needs of the HPC Education and Training community with regard to sharing training materials we sought input from stakeholders. To this end, we conducted a survey to explore interest and key factors related to sharing and discovering training and education materials. The results of this survey highlighted the barriers to finding relevant materials and the barriers to sharing materials developed. Overall, we learned that there is a great deal of interest in sharing materials developed more widely and making access to these materials easier for learners.

https://doi.org/10.22369/issn.2153-4136/14/2/4

2 SURVEY BACKGROUND & MOTIVATION
The survey aimed to assess if individuals or organizations have training materials they wish to share or make more widely accessible, and if local communities require more efficient means of locating relevant materials. We were interested in learning how the research computing community shares and finds these materials and if they thought there should be more or better options for doing this. The term "repositories" was used broadly to encompass portals, collections, libraries, and lists of training and education materials and resources for the purposes of this survey; this statement was included in the survey preamble.

3 SURVEY METHODOLOGY
The survey [2] authors developed the survey questions which then were reviewed by a focus group from the HPC education and training community. To ensure accessibility, the survey was conducted using Google Forms and remained open for several weeks in October and November 2022. Invitations to participate were sent out through well-known mailing lists in the HPC support community, such as CaRCC People Network, Campus Champions, Virtual Residents, Coalition for Academic Scientific Computation, and the EDU Special Interest Group on High Performance Computing. This effort resulted in a total of up to 112 responses received for each question. We were targeting professionals in the HPC support community at research computing centers. We shared interim results at the Ninth SC Workshop on Best Practices for HPC Training and Education (BPHTE22) at SC22.

4 SURVEY RESULTS

The survey results based on 112 responses to the questions listed in Appendix A are described in this section. The survey data is available online [2].

4.1 About the Respondents

The survey begins with two questions pertaining to the respondent's role and the communities they support. Both questions allowed multiple selections and were answered by all 112 respondents. Tabulated results, summarized in Table 1 and Table 2, show that 84% are, or support, academics, closely followed by 71% for both the Grad/Post doc and Undergrad communities. 37% of the respondents specifically selected these three options only. 84% are content authors, while 61% curate appropriate materials for their community. 25% see themselves as filling all four roles.

Table 1. Which communities do you support or participate in?

Community	Responses
Academia	94
Grad/Post doc	80
Undergrad	80
Government	25
Pre-college	18
Industry	16
Other	4
Total	**112**

Table 2. When it comes to training and education materials, I consider myself to be:

Respondent's role	Responses
Content author	94
Curator collecting appropriate materials for my community	68
Consumer of materials hosted by other organizations	57
Consumer of materials hosted by my organization	35
Other	6
Total	**112**

4.2 Finding Materials

Question 3, with 111 responses, show only 20% found it difficult to find appropriate training and education materials on specific topics, for themselves. Question 4, "How easy is it for you to find appropriate training and education materials on specific topics, for group(s) you support?", resulted in only 32% of 111 respondents saying that it was difficult or very difficult.

In question 5, we asked a multiple selection question to learn where respondents look for material, with 111 responses. 94% use search engines, 64% use portals or repositories hosted by other organizations, 45% use a portal or repository hosted locally, and 17% selected Other.

Question 6, "Which portals, repositories, search engines or other resources do you use or find helpful?" resulted in a broad array of both general and specific responses by 93 respondents. In question 7 we asked "How important is, or would be, having easy access to repositories of training and education materials from multiple organizations to your community?"; on a scale of 1 for "not important" to 5 for "very important," the average for 112 responses was 4.3. Figure 1 shows the results of questions 3, 4, and 7, displaying ease of finding materials along with the importance to material access to their community.

Question 8, regarding barriers encountered when searching for materials, answered by 109, asked for all barriers encountered, with results shown in Table 3. 66% said they can find materials, but not at the right depth or level needed. Question 9, answered by 108, asked for the single barrier that it would be most helpful to remove; The top answer was again that they can find materials, but not at the right depth or level needed, with 43%, as shown in Table 4.

4.3 Working Toward Solutions

Question 10 asked if the respondent's organization wants to make it easier to find their materials, question 11 asked for the biggest challenge to sharing, and question 12 asked whether the organization would be willing and able to provide metadata in a standard format. Figure 2, which combines results from questions 10 and 12, shows that most of the respondents fall into the top right area, both interested in, and able to, share materials. Question 11, which called for free response, was answered by 99 people with a broad variety of responses, including time, cost, copyright, and issues raised by the organization.

87 people responded to question 13, a multiple selection question seeking to identify which roadblocks prevent an organization from sharing content information in a standard format. Results in Table 5 show that 75% cited lack of staff time. 7% selected all options, while 30% selected a single reason, lack of staff time.

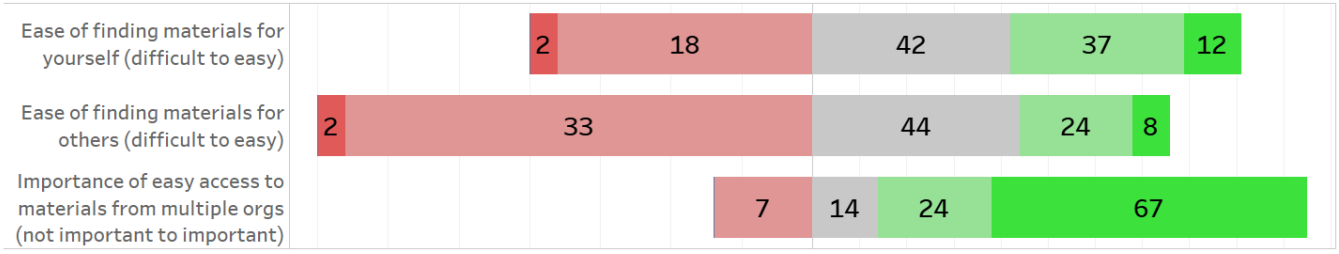


Figure 1. Top: Question 3, finding material for self. Middle: Question 4, finding material for others. Bottom: Question 7, importance of access to materials from multiple organizations.

In question 14, 47 people indicated they are interested in joining a group working on this project by providing contact information. In response to question 15, 56 respondents indicated they are willing to be contacted regarding survey responses. 36 people responded to the closing comments question.

Table 3. What barriers have you encountered when searching for materials?

Barriers encountered	Responses
I can't find materials on the topic I need	35
I can find materials on the topic, but not at the depth or level I need	72
I find too many materials, and I can't effectively sort through them all	44
I am aware of specific appropriate materials, but search engines don't list them in the top results	26
Other	28
Total	**93**

Table 4. Which barrier, if removed, would be most helpful for finding appropriate materials?

Barrier to remove	Responses
I can't find materials on the topic I need	12
I can find materials on the topic, but not at the depth or level I need	46
I find too many materials, and I can't effectively sort through them all	29
I am aware of specific appropriate materials, but search engines don't list them in the top results	10
Other	11
Total	**109**

Table 5. If your organization is not willing and able to provide metadata about your materials in a standard format, what are your roadblocks?

Roadblocks to providing metadata	Responses
Lack of staff time	65
Lack of funding	38
Inadequate staff expertise	28
Our materials aren't in a catalog	37
Other	12
Total	**87**

5 SURVEY ANALYSIS

The 112 survey responses and results show a strong interest and importance (question 7) in the topic of finding and sharing (question 10) education and training materials in the cybertraining community. While we saw strong interest, the results also showed many barriers (questions 8 & 9).

In questions 3, 4, and 7, we found that most of the respondents considered themselves successful in finding appropriate materials for themselves and others, while question 8 shows 109 responses listing barriers encountered when searching for materials, shown in Table 3. Perhaps this indicates that finding materials, while possible by dedicated professionals, could be significantly improved.

Figure 2 displayed two sparse quadrants. In the upper left quadrant, it is unsurprising that only one respondent is both able to provide metadata but uninterested. It is more interesting to see that the bottom right quadrant is also sparse; there are only 4 respondents saying that they want to make finding data easier, but don't have the ability. This shows great potential in the community moving forward with solutions.

Altogether, the results imply that the community sees the potential for improving discovery of materials and many have the interest and ability to contribute to a solution.

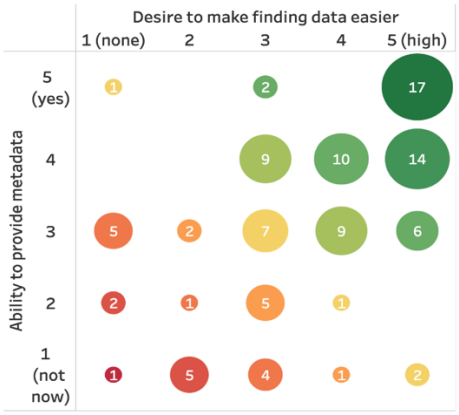


Figure 2. Sharing materials: interest and ability

6 RECOMMENDATIONS & FUTURE WORK

6.1 Community Engagement

6.1.1 Collaborating with HPC Education and Training Communities

We plan to collaborate with educational and training material organizations that focus on high-performance computing (HPC). This will involve our participation in various organizations such as the ACM SIGHPC and the NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing, working with institutions such as the NSF ACCESS MATCH program, Cornell Center for Advanced Computing, Kean University, San Diego Supercomputer Center, Texas Advanced Computing Center, Ohio Supercomputer Center, Pittsburgh Supercomputing Center, and others. In addition, we intend to connect with organizations that have received NSF CyberTraining awards to explore opportunities for sharing their

training products and increasing usage within and across disciplines.

6.1.2 *Build an HPC Professional Trainer Community*

Building a diverse community for HPC education and training begins with building a diverse trainer community. A diverse training community is an important goal because it confronts preconceived stereotypes in learning and education, allowing enhancements to both workplace and community cohesion. Diversity encourages critical thinking while helping students learn to communicate effectively with people of varied backgrounds [3]. The scientific research community is very diverse; therefore, a diverse group of educators is important. The Train-the-Trainer model is very effective in this pursuit [1]. By directly engaging and encouraging the underrepresented minority community in HPC at symposiums such as ADMI (The Association of Computer Science Departments at Minority Institutions), we can support their training and understanding of HPC systems and technologies. Those participants can then take their newly enhanced knowledge to their education and research institutions and train their fellow colleagues. The primary issue to overcome is finding high quality material that has been properly curated, which a federated and decentralized catalog of HPC training material can solve.

6.1.3 *Organize Birds-of-a-Feather (BoFs) Meetings*

We intend to hold Birds-of-a-Feather (BOFs) to share our findings and to gather more input from relevant communities. We will target key technical meetings such as PEARC23 [5], Supercomputing (SC23) [6], and ISC [4] where large community gatherings occur, allowing us to discuss and work toward solutions to the opportunities for improvement in finding and sharing materials.

6.2 Organize Community Hosted Training Material Services

Our survey showed that training materials across institutions are currently isolated and distributed, and the community recognizes the need for improving discovery and sharing of materials. Our goals include extending the reach of our training materials into underserved communities and identifying gaps in training. The lack of a central platform for sharing training and event services in HPC is a key factor in hindering discovery and advertising of training opportunities.

We plan to work with the HPC Training and Education Communities to identify best practices for sharing training resources. This includes using metadata tagging, adopting publishing mechanisms like GitHub or ReadTheDocs, open sharing of training materials, and collecting and disseminating educational material reviews and ratings. Training gaps can be filled through regular communication between contributors and the community, enhancing local portals by adding training materials shared by others. Shared material contributions would come from organizations that have a history of creating, developing, collecting, and displaying computational science education and training materials, as well as individual developers.

ACKNOWLEDGMENTS

We want to acknowledge the use of several NSF funded resources and services including: the SDSC Expanse project (#1928224); TACC Stampede System (# 1663578); the NSF Track 3 Award: COre National Ecosystem for CyberinfrasTructure (CONECT) (#2138307); and the Extreme Science and Engineering Discovery Environment (XSEDE) (NSF award #ACI-1548562). We also want to acknowledge Ben Trumbore for creating the two figures.

REFERENCES

[1] Elizabeth Bautista and Nitin Sukhija, 2021. Employing directed internship and apprenticeship for fostering HPC training and education. *JOCSE, 12,* 2. *https://doi.org/10.22369/issn.2153-4136/12/2/8*

[2] Katherine Cahill, David Joiner, Scott Lathrop, Susan Mehringer, JP Navarro, and Aaron Weeden, *Final results: National survey on educational and training materials repositories.* Retrieved from https://www.cac.cornell.edu/about/pubs/Survey2022.pdf

[3] Patricia Gurin, Eric Dey, Sylvia Hurtado, and Gerald Gurin. 2002. Diversity and higher education: Theory and impact on educational outcomes. *Harvard Educational Review, 72*, 3. https://doi.org/10.17763/haer.72.3.01151786u134n051

[4] ISC High Performance. n.d. Retrieved from https://www.isc-hpc.com/

[5] PEARC. n.d. Retrieved from https://pearc.acm.org/pearc23/

[6] SC23. 2023. 2023. Retrieved from https://sc23.supercomputing.org/

APPENDIX A: SURVEY INSTRUMENT

The following is the survey as it appeared while it was active in 2022; this section shows both the introductory text and lists the questions in the order they appeared.

A working group has been formed within the ACM SIGHPC Education Chapter to discuss metadata standards for sharing materials across all interested organizations. We are asking the community to complete a short survey to understand the challenges and opportunities for the ACM SIGHPC Education Chapter to consider in promoting metadata standards. We will share what we learn from the survey at the Ninth SC Workshop on Best Practices for HPC Training and Education (BPHTE22) at SC22 and during the ACM SIGHPC Education Chapter working group.

We hope you will join us for the SC22 workshop presentations and discussions, in person or virtually, and we welcome you to join the ACM SIGHPC Education Chapter working group.

For the purposes of this survey, we use the term repositories to broadly include portals, collections, libraries, and lists of training and education materials and resources.

SURVEY PROCEDURES & CONFIDENTIALITY

If you agree to participate in the survey, you will not be required to provide any identifying information, and you will not be required to complete all questions. You will have the option of providing your name and contact information if future contact is desired. Efforts will be made to keep confidential any self-identifying information that you intentionally or inadvertently disclose. Your identity will be held in confidence in reports in which the survey results may be published and/or databases in which results may be stored. We may use aggregated data or anonymous comments from the survey in reports.

1. Which communities do you support or participate in? (check all that apply): *Multiple selections: (a) Pre-College, (b) Undergrad, (c) Grad/Post Doc, (d) Academia, (e) Government, (f) Industry, (g) Other*

2. When it comes to training and education materials, I consider myself to be (check all that apply): *Multiple selections: (a) Consumer of materials hosted by my organization, (b) Consumer of materials hosted by other organizations, (c) Content author, (d) Curator collecting appropriate materials for my community, (e) Other*
3. How easy is it for you to find appropriate training and education materials on specific topics, for yourself? *Likert scale: 1 (very difficult) - 5 (very easy)*
4. How easy is it for you to find appropriate training and education materials on specific topics, for group(s) you support? *Likert scale: 1 (very difficult) - 5 (very easy)*
5. Where do you look for material? *Multiple selections: (a) Portal or repository hosted by my organization, (b) Portals or repositories hosted by other organizations, (c) Search engine, (d) Other*
6. Which portals, repositories, search engines or other resources do you use or find helpful? *Free response text*
7. How important is, or would be, having easy access to repositories of training and education materials from multiple organizations to your community? *Likert scale: 1 (not important) - 5 (very important)*
8. What barriers have you encountered when searching for materials? *Multiple selections: (a) I can't find materials on the topic I need, (b) I can find materials on the topic, but not at the depth or level I need, (c) I find too many materials, and I can't effectively sort through them all, (d) I am aware of specific appropriate materials, but search engines don't list them in the top results, (e) Other*
9. Which barrier, if removed, would be most helpful for finding appropriate materials? *Single selection: (a) I can't find materials on the topic I need, (b) I can find materials on the topic, but not at the depth or level I need, (c) I find too many materials, and I can't effectively sort through them all, (d) I am aware of specific appropriate materials, but search engines don't list them in the top results, (e) Other*
10. Does your organization want to make it easier for the public to find your training and education materials? *Likert scale: 1 (not at all) - 5 (very much)*
11. What do you consider to be the biggest challenge(s) in sharing your materials? *Free response text*
12. Would your organization be willing and able to share your training and education materials in a public catalog by providing metadata about your materials in a standard format? *Likert scale: 1 (not at this time) - 5 (yes, even if it takes a few weeks)*
13. If your organization is not willing and able to provide metadata about your materials in a standard format, what are your roadblocks? *Multiple selections: (a) Lack of staff time, (b) Lack of funding, (c) Inadequate staff expertise, (d) Our materials aren't in a catalog, (e) Other*
14. Are you interested in joining a group working on this project? If so, please provide your contact information or write to hpc.edu.train@gmail.com. *Free response text*
15. Are you willing to be contacted by the survey organizers for follow-up regarding your responses? If so, please provide your contact information. *Free response text*
16. We would be happy to hear any additional comments you have on this topic. *Free response text*

Access to Computing Education Using Micro-credentials for Cyberinfrastructure

Dhruva K. Chakravorty
chakravorty@tamu.edu
HPRC[1]

Richard Lawrence
rarensu@tamu.edu
HPRC[1]

Zhenhua He
happidence1@tamu.edu
HPRC[1]

Wesley Brashear
wbrashear@tamu.edu
HPRC[1]

Honggao Liu
honggao@tamu.edu
HPRC[1]

Andrew J. Palughi
ajp2795@tamu.edu
HPRC[1]

Lisa M. Perez
perez@tamu.edu
HPRC[1]

Xin Yang
karen89@tamu.edu
Medical College of Wisconsin
Milwaukee, WI, USA

Jacob Pavelka
jacobpavelka98@tamu.edu
Texas A&M University
College Station, TX, USA

Ritika Mendjoge
ritika.mendjoge@tamu.edu
HPRC[1]

Marinus Pennings
pennings@tamu.edu
HPRC[1]

Randy McDonald
randymcdonald@tamu.edu
HPRC[2]

Gerry Pedraza
gpedraza@tamu.edu
HPRC[2]

Sunay V. Palsole
sunay.palsole@tamu.edu
HPRC[2]

ABSTRACT

In response to an increasing demand for digital skills in industry and academia, a series of credentialed short courses that cover a variety of topics related to high performance computing were designed and implemented to enable university students and researchers to effectively utilize research computing resources and bridge the gap for users with educational backgrounds that do not include computational training. The courses cover a diverse array of topics, including subjects in programming, cybersecurity, artificial intelligence/machine learning, bioinformatics, and cloud computing. The courses are designed to enable the students to apply the skills they learn to their own research that incorporates use of large-scale computing systems. These courses offer advantages to generic online courses in that they teach computing skills relevant to academic research programs. Finally, the micro-credentials obtained from these courses are transcriptable, may be stacked with existing degree programs and credit-bearing courses to create a larger degree plan, and offer a meaningful mechanism of adding to a student's resume.

KEYWORDS

Micro-credentials, Computing Education, Python, R, Artificial Intelligence, Machine Learning, Bioinformatics, Cybersecurity, Linux

https://doi.org/10.22369/issn.2153-4136/14/2/5

1 INTRODUCTION

Growing federal and state-funded programs have contributed to increase the availability of cyberinfrastructure (CI) resources to researchers at institutions of all sizes [1][2][3]. Job-skills data collected by National Science Foundation CORD (for Texas) and similar studies by other Federal (Department of Energy COVID-19 preparedness report etc.) and private agencies (Deloitte etc.) show an increasing demand in industry and academia for digital skills in the areas of coding, artificial intelligence (AI/ML), bioinformatics, and cybersecurity. Now, more than ever, the efficacy of a research team is limited by their ability to effectively use CI resources. Simultaneously, research workflows integrate interdisciplinary approaches and rely on researchers having expertise in several domains of science. As such, researchers find them simultaneously requiring assistance in (i) learning how to use technologies on CI resources, and (ii) finding support in CI-enabled fields of science that go beyond their local fields of expertise. Micro-credentials offer an interesting opportunity toward alleviating this urgent need to train researchers in computing technologies. They could provide an accelerated introduction for new students and faculty who are interested in computing for research. By design, these courses should be of short durations, ensuring that researchers can readily take them without committing to an entire semester of study, and should teach computing skills relevant for pursuing further academic coursework.

The High Performance Research Computing group has previously developed and offered short courses using a model of continuous improvement. Materials for courses presently in use can

[1]Texas A&M University High Performance Research Computing, College Station, TX
[2]Texas A&M University Engineering Studio for Advances Instruction & Learning, College Station, TX

be dated to early iterations in 2017 [4]. Methods of teaching and delivery have been explored and reported in previous publications [10][7][5][6][9]. It was shown that users are interested in remote learning options and that these can be effective [7]. It was argued that researchers prefer to learn code using interactive graphical interfaces and therefore computing educators should leverage appropriate platforms when teaching computing skills [10]. The platforms that were identified for this purpose were OnDemand, which is a browser interface for HPC systems, and Google Colab, which is a freely available graphical IDE for coding practice.

2 METHODS

Funded by the Texas Higher Education Coordinating Board (THECB) program for accelerated credentials, these credentialed courses serve multiple audiences. Our surveys and interviews of industry employers and academic programs, coupled with participation trends in workforce programs indicate a strong need for a program offering quick stackable credentials in digital skills to graduate students, postdoctoral associates and professionals. Students emphasize that they should have opportunities to develop these skills at their own pace. To ensure that learners with different learning needs are accommodated, the credential courses are offered through both in-person workshops, and online asynchronous options. Each course incorporates a combination of brief lectures with hands-on exercises to optimize student engagement with the material. The course offerings cover a variety of topics, grouped categorically as follows: Coding, Cybersecurity, Artificial Intelligence/Machine Learning, Bioinformatics, and Cloud Computing. Next, our credentialed short courses provide a bridge to computing for senior undergraduate students and junior graduate students in non-traditionally computational programs; e.g. Economics.

For asynchronous courses, the presentation material was generated in Microsoft PowerPoint, Google Colab, and the learnR package. The PowerPoint slides were useful in explaining broad concepts, such as the block structure of control flow in Python programs, and Colab was useful in explaining the mechanics of the broad concepts, such as illustrating the required arguments for a Python package function. Additionally, Colab could be used to facilitate demonstrations and practice assignments for the students. Google Colab gave students access to the Python programming language as well as a Linux kernel for practicing system administration.

As part of our design, we utilized best-practices in curricular design such as building modules to ensure easier adoption in credit bearing courses, matching assignments to learning objectives and outcomes, and adopting principles of interactive lecturing and adult learning principles to ensure student engagement. Each course was developed in a modular manner. Courses were organized into modules and topics. Modules served as distinct groupings of materials and topics were used to break modules into 10-15 minute digestible lectures in which students could take breaks after and assess their previous course in its entirety. The presentation material followed this organizational structure.

Upon preparing the presentation material, recordings were made using the facilities in the Engineering Studio for Advanced Instruction and Learning (eSAIL). An MP4 format was used for these recordings to allow additional data to be associated with them, such as closed captioning. The Descript software suite was used to implement closed captioning and edit the video recordings. This suite included an AI to predict the captioning from the audio file. This accelerated the process of closed captioning significantly.

Multiple-choice quizzes were chosen for these courses' assessment. While not ideal for assessing programming comprehension, this type of quiz allowed for automatic assessment, where a short answer quiz would require manual assessment. Quiz questions covered broad programming concepts, course specific concepts (Machine learning courses required understanding of gradient descent), and programming syntax and usage concepts.

Finally, Canvas was chosen as the delivery medium for the asynchronous courses. Canvas is a learning management system that is commonly used by universities for official coursework. On Canvas, video files and coding assignments can be given to the students together, and quizzes can be given to the students with randomized quiz questions such that students will be less likely to take the same quiz.

The courses, and credentials are delivered with the assistance of Texas A&M Experiment Station Educating Generations (TEES EDGE), a group with a long history of working with industry and State agencies to deliver workforce development training and education at the post baccalaureate, graduate and continuing education levels. The TEES EDGE website provides a full set of services that allows registration, tracking and awarding of credentials to include record maintenance for participants of the program. The courses utilize a variety of different resources to train students. Several courses require students to run various software programs with example data directly on our campus computing clusters. Other courses use platforms such as Google Colab and the learnR package in R to engage students.

Several parameters are identified for each course. These included the CIP CODE, the designated major fields of study (Data Science, CyberSecurity etc.), intended audience (undergraduate students, graduate students, postdoctoral associates, professionals, teachers etc.), the duration of the course (days/ hours / weeks), equivalent professional development units, the number of contact hours that a student will spend with the instructor or learning materials, the mode of delivery (digital, face-to-face, or hybrid), and the possible linking programs. The Classification of Instructional Programs (CIP) code is the federal government classification system that standardizes fields of study across the U. S. The codes in Table 1 whose first two digits are 11 fall under the category of "Computer and Information Sciences and Support Services." The codes beginning with 16 are "Biological and Biomedical Sciences," and the codes under 30 are "Multi/Interdisciplinary Studies." [8]. Additionally, for each course, quizzes and feed-back surveys were developed. To ensure rapid deployment of credential courses, we leveraged existing curricular materials and worked with experienced instructors. Many of the credentialed courses were built on previously offered workshops or other courses [9]. This has allowed us to leverage the feedback we have received from previous students to make improvements and ensure that the credentialed courses are beneficial to the learners. Instructional support and course curricula were delivered by experienced instructors at the High Performance Research Computing (HPRC) group at Texas A&M. HPRC's decade-long informal training programs have taught students computing digital

and analytics skills using methods compatible with best practices for computing, so the students who complete our programs will be ready to use computing resources responsibly and effectively. The Engineering Studio for Advanced Instruction & Learning (eSAIL) at Texas A&M led the instructional design of the project. The eSAIL team worked with the subject-matter experts at HPRC to build interactive and flexible online, blended, and technology enabled face-to-face courses. The group has state of the art facilities and includes instructional and learning designers, multimedia specialists, and a learning architect. eSAIL's instructional designers ensure that learning outcomes of the course are measured, and all feedback is folded into a continuous improvement cycle.

3 RESULTS

3.1 Overview of Courses

The coding courses include incremental levels of Python (Fundamentals, Intermediate, and Advanced) and R (Introductory and Intermediate). These courses stack with each other and build a foundation for courses in the other categories. In these courses, the students learn foundational knowledge of programming, such as flow control, data structures, and object-oriented programming. They also learn skills in data science, such as analysis and visualization. In addition to the CI-based courses, we have also created a 3-contact hour asynchronous course titled "Fundamentals of Cybersecurity" that introduces professionals to basic topics in Cybersecurity. This introductory course teaches students the important concepts and terminology of cybersecurity. This covers fundamental cybersecurity principles such as the types of cyberattacks and how to defend against them.

The Artificial Intelligence/Machine Learning (AI/ML) section includes "Fundamentals of AI/ML", which introduces some fundamentals of AI and ML including their relationship, different types of data, training and testing, common types of learning techniques (supervised and unsupervised learning) and applications (regression, classification, and clustering). The short course "Introduction to Deep Learning with TensorFlow" gives a brief introduction to deep learning with TensorFlow (an open-source software library for machine intelligence) and covers basic concepts of deep learning methods. "Introduction to Deep Learning with PyTorch" covers the basic concepts of deep learning and PyTorch with examples. PyTorch is based on the Torch library and can be used for various applications such as computer vision and it provides tensor computing that could be accelerated with GPUs. Finally, the short course "Using SciKit-Learn for AI and ML" introduces some fundamentals of AI and ML and machine learning algorithms including linear regression, logistic regression, Support Vector Machine (SVM), K Nearest Neighbors (KNN), and K-Means clustering with guided practices.

A number of topics are covered in the Bioinformatics section, including "RNA-seq and Differential Expression". In this course, students learn the basic steps that need to be completed for differential expression analyses, including library quality control and trimming, read alignment to a reference genome, generating count files, and differential expression analysis and data visualization in R. Another short course in this section is "Introduction to Metagenomics". This course covers fundamental concepts of conducting metagenomic experiments with next-generation sequencing data, including working with whole genome sequencing data, targeted amplicon sequencing, metagenomic assembly methods, and using the Qiime2 software suite. The course "Introduction to ChIP-seq" teaches students the basic workflow to analyze data generated when combining chromatin-immunoprecipitation with massively parallel sequencing. The Bioinformatics section also includes "Short Variant Discovery", where students use example data to work through a typical short variant discovery pipeline, from library QC and trimming, to mapping reads to a reference genome, and calling variants with the GATK software suite.

The final section on Cloud Computing contains several in-person courses, including "Fundamentals of Linux", "Linux for Administrators", "Job Scheduling SLURM", "Containers and Orchestration", and "Introduction to Cloud and Cluster Computing". The asynchronous offerings include "Parallel Computing Using OpenMP", which introduces students to parallelizing their code using OpenMP and covers topics including OpenMP concepts, OpenMP program layout, worksharing constructs, synchronization pragmas, and OpenMP tasks. Lastly, the course "Parallel Computing Using MPI" covers the Message Passage Interface (MPI), a standard library to create parallel codes for distributed systems. Topics covered in this course include MPI terminology, Communicators, Point to Point communications, and collective communications.

3.2 Course Details

Learning Objectives by course

- Fundamentals of Cybersecurity
  - Define computer security and approaches to implementing it.
  - Provide general definitions for computer security concepts
  - Describe methods of Social Engineering.
  - Describe Malware and the software vulnerabilities it preys on.
- Fundamentals of Artificial Intelligence and Machine Learning
  - Understand the fundamentals of AI/ML including AI and ML relationship, Training and Testing, Supervised and unsupervised learning, Regression, classification, and clustering.
  - Solve some simple regression, classification, and clustering problems with a machine learning library.
- Introduction to Deep Learning with TensorFlow
  - Understand the fundamentals of deep learning, including what deep learning is and why we need it, its learning principle, convolution, pooling operations and neural networks.
  - Use TensorFlow Keras API to build and train an image classification neural network
- Introduction to Deep Learning with PyTorch
  - Understand the fundamentals of deep learning
  - Use PyTorch framework to build and train an image classification neural network
- Using SciKit-learn for Artificial Intelligence and Machine Learning
  - Understand the fundamentals of AI/ML

Table 1: The credential courses provided by the program

Theme	Course Title	CIP CODE(s)	Duration	Delivery Mode
Cybersecurity	Fundamentals of Cybersecurity	11.1003	0.3 PDU	Asynchronous
Coding	Fundamentals in Python Programming	11.0201	1 PDU	Both
	Intermediate Python Programming for Data Science	11.0202, 30.71	1 PDU	Both
	Advanced Python Programming with Xarray and Dask	11.0202, 30.71	0.5 PDU	Both
	Fundamentals R Programming	11.0201	1 PDU	Both
	Intermediate R Programming	11.0202	1 PDU	Both
	GPU Programming with CUDA	30.3001	0.3 PDU	Live
AI/ML	Fundamentals of Artificial Intelligence and Machine Learning	11.0102	0.3 PDU	Both
	Introduction to Deep Learning with TensorFlow	11.0804, 11.0202	0.3 PDU	Both
	Introduction to Deep Learning with PyTorch	11.0804, 11.0202	0.3 PDU	Both
	Using Scikit-Learn for Artificial Intelligence and Machine Learning	11.0804, 11.0104, 11.0202	0.3 PDU	Both
Bioinformatics	RNA-seq and Differential Expression	11.0104, 11.0401, 26.1103	0.3 PDU	Both
	Short Variant Discovery	11.0104, 11.0401, 26.1103	0.3 PDU	Both
	Introduction to Metagenomics	11.0104, 11.0401, 26.1103	0.3 PDU	Both
	Introduction to ChIP-seq	11.0104, 11.0401, 26.1103	0.3 PDU	Both
Linux	Fundamentals of Linux	11.0201	0.3 PDU	Both
	Linux for Administrators	11.1006, 11.1001	0.3 PDU	Live
Cloud Computing	Job Scheduling with SLURM	11.0103	0.3 PDU	Live
	Containers and Orchestration	11.0103	0.3 PDU	Live
	Introduction to Cloud and Cluster Computing	11.0103	0.3 PDU	Live
	Parallel Computing Using OpenMP	11.0201	0.3 PDU	Asynchronous
	Parallel Computing Using MPI	11.0201	0.3 PDU	Asynchronous

  - Understand some commonly used machine learning algorithms including Linear Regression,
  - Logistic Regression, Support Vector Machine (SVM) and K-Means Clustering.
  - Use Scikit-learn machine learning library to solve regression, classification and clustering problems.
- RNA-seq and Differential Expression
  - Learn about the different techniques used to generate RNA-seq libraries
  - Learn how to properly design a differential expression study
  - Understand the basic steps that need to be completed for differential expression analyses
- Short Variant Discovery
  - Learn how to work through a typical short variant discovery pipeline from library QC and trimming, to mapping reads to a reference genome, and calling variants with the GATK software suite.
- Metagenomics
  - Learn how to use the Qimme2 bioinformatics platform with example data.
- ChIP-seq
  - Learn about next generation sequencing library QC and trimming
  - Align reads to a reference genome
  - Filtering and sorting alignment files
  - Calling ChIP-seq peaks
- Fundamentals of Linux
  - Utilize some commonly used Linux commands for management of files and directories, I/O redirection, customizing environment, and text processing.
- Linux for Administrators
  - Master some Linux administration skills including account management, packages installation, monitoring disk usage, and process control.

Assessment

The course quizzes are in the format of multiple choice questions as shown in Figure 1, which are used to evaluate how well the students understand what they learned and how well they mastered the material.

3.3 Promotion and Initial Reception

The short courses offering micro-credentials were announced to a broad, Texas A&M affiliated audience during a presentation open

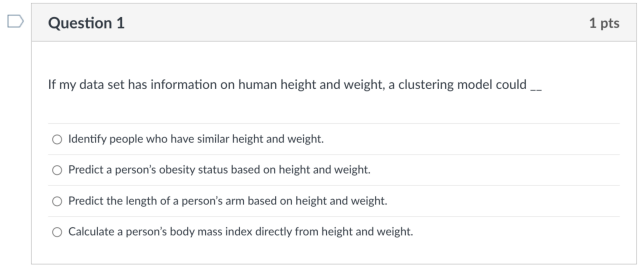


Figure 1: An example of multiple choice quiz questions.

to the university. During the presentation, we discussed the micro-credentialing program, the basic structure of the courses (i.e. modular, stackable design), and detailed the courses that were being offered. The first in-person micro-credentialing course scheduled following this presentation (an 8-week series covering Python programming) resulted in 79 individual registrations before being closed due to physical space constraints. Subsequent requests for this in-person short course were directed to the online asynchronous offering.

4 CONCLUSIONS

The micro-credential-bearing short-courses offer hands-on and project-based learning experiences to students in a choice of live-training and asynchronous educational scenarios. These courses will be developed and iteratively refined in consultation with our academic, workforce, and industry partners to ensure they meet the current and expected needs of academia and the labor market. We ensured rapid deployment of these courses by building on top of successful training models that have been used in the past for other upskilling efforts. The program is offered to students residing in Texas, via a web-based credential system that tracks student participation and offers easy integration for providing upskilling, enabling students state-wide to display their digital prowess, and stacking formal, informal, professional education programs.

The presented credentialed short courses have two major benefits that make them a superior choice to a generic online programming course. First, exercises and topics are chosen that teach skills relevant for scientific research. Second, the credentials provided offer a meaningful reward for completion that can be recognized by academic programs and industry professionals. To promote sustainability, these training materials will transition from informal efforts into curricular products. The collaboration leverages expertise in facets of computational sciences and large scale computing to address a number of long-standing issues encountered in accessing computing resources. From this work, a new, shareable model is provided that increases the accessibility of available resources in areas of multi-disciplinary appeal for researchers. The goal is that CI facilitators at institutions can utilize these resources in training and research workflow support activities.

ACKNOWLEDGMENTS

This work was supported by the Texas Higher Education Coordinating Board (THECB), the National Science Foundation (NSF) award number 1925764, "CC Cyberteam SWEETER", NSF award number 2019129, "MRI:FASTER", NSF award number 1730695, "CyberTraining: CIP: CiSE-ProS: Cyberinfrastructure Security Education for Professionals and Students", NSF award number 1818253, "Frontera: Computing for the Endless Frontier", NSF award number 2019136, "CC BRICCs: Building Research Innovation at Community Colleges", and NSF award number 2112356 "ACES, Accelerating Computing for Emerging Sciences".

REFERENCES

[1] 2020. NSF Major Research Instrumentation Program (MRI). Retrieved August 28, 2023 from https://beta.nsf.gov/funding/opportunities/major-research-instrumentation-program-mri
[2] 2021. NSF Campus Cyberinfrastructure (CC*). Retrieved August 28, 2023 from https://beta.nsf.gov/funding/opportunities/campus-cyberinfrastructure-cc
[3] 2022. Advanced Cyberinfrastructure Coordination Ecosystem: Services and Support (ACCESS). Retrieved August 28, 2023 from https://access-ci.org/
[4] 2023. High Performance Research Computing Past Short Courses. Retrieved August 28, 2023 from https://hprc.tamu.edu/training/previous.html
[5] Dhruva Chakravorty and Minh Tri Pham. 2020. Evaluating the Effectiveness of an Online Learning Platform in Transitioning Users from a High Performance Computing to a Commercial Cloud Computing Environment. *The Journal of Computational Science Education* 11 (Jan. 2020), 93–99. Issue 1. https://doi.org/10.22369/issn.2153-4136/11/1/15
[6] 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
[7] Dhruva K. Chakravorty, Lisa M. Perez, Honggao Liu, Braden Yosko, Keith Jackson, Dylan Rodriguez, Stuti H. Trivedi, Levi Jordan, and Shaina Le. 2021. Exploring Remote Learning Methods for User Training in Research Computing. *The Journal of Computational Science Education* 12 (Feb. 2021), 11–17. Issue 2. https://doi.org/10.22369/issn.2153-4136/12/2/2
[8] Institute for Educational Sciences. 2020. The Classification of Instructions Programs. Retrieved September 9, 2022 from https://nces.ed.gov/ipeds/cipcode
[9] Richard Lawrence, Zhenhua He, Wesley Brashear, Ridham Patoliya, Honggao Liu, and Dhruva K. Chakravorty. 2022. Tailored Computing Instruction for Economics Majors. *The Journal of Computational Science Education* 13 (April 2022), 32–37. Issue 1. https://doi.org/10.22369/issn.2153-4136/13/1/6
[10] Richard Lawrence, Tri M. Pham, Phi T. Au, Xin Yang, Kyle Hsu, Stuti H. Trivedi, Lisa M. Perez, and Dhruva K. Chakravorty. 2022. Expanding Interactive Computing to Facilitate Informal Instruction in Research Computing. *The Journal of Computational Science Education* 13 (April 2022), 50–54. Issue 1. https://doi.org/10.22369/issn.2153-4136/13/1/9

Multifaceted Approaches for Introducing a Hardware-thread Migratory Architecture

Aaron Jezghani
ajezghani3@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia

Jeffrey Young
jyoung9@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia

Vedavyas Mallela
vmallela6@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia

Will Powell
will.powell@cc.gatech.edu
Georgia Institute of Technology
Atlanta, Georgia

ABSTRACT

The challenges of HPC education span a wide array of targeted applications, ranging from developing a new generation of administrators and facilitators to maintain and support cluster resources and their respective user communities, to broadening the impact of HPC workflows by reaching non-traditional disciplines and training researchers in the best-practice tools and approaches when using such systems. Furthermore, standard x86 and GPU architectures are becoming untenable to scale to the needs of computational research, necessitating software and hardware co-development on less-familiar processors. While platforms such as Cerebras and SambaNova have matured to include common frameworks such as TensorFlow and PyTorch as well as robust APIs, and thus are amenable to production use cases and instructional material, other systems may lack such infrastructure maturity, impeding all but the most technically inclined developers from being able to leverage the system.

We present here our efforts and outcomes of providing a co-development and instructional platform for the Lucata Pathfinder thread-migratory system in the Rogues Gallery at Georgia Tech. Through a collection of user workflow management, co-development with the platform's engineers, community tutorials, undergraduate coursework, and student hires, we have been able to explore multiple facets of HPC education in a unique way that can serve as a viable template for others seeking to develop similar efforts.

KEYWORDS

HPC education, novel architecture workflows, CS curriculum, workforce development, community education

1 INTRODUCTION

Despite overlaps with traditional computing, HPC education and training requires specialized skills, especially in terms of code scaling and target hardware dispatch. Target audiences for this type of training can include students [2, 4, 16], researchers [6, 21], and future administrators/facilitators [1, 22]. Approaches for training range from using traditional classroom instruction to condensed workshops or tutorials to unplugged activities to overcome the challenges of HPC training across the array of demographics and science domains, with a number of focus groups running annual workshops to aggregate best practices and pave the path forward; for example, see [10–12, 24].

Further complicating the state of HPC education is the ever-changing landscape of cluster infrastructure. Notably, the broad recognition of scalability challenges in the movement of data for modern computational research has led to the introduction of a new standard for a high-bandwidth, disaggregated architecture with shared memory access, Compute Express Link, or CXL [7]. Although systems capable of supporting the first generation of the CXL standard are just now deploying, major vendors have changed product offerings [8], and it is largely expected that broad adoption will follow the adoption of the 2.0 standard by 2025 [15]. Given the major shift in infrastructure, the need for a well-developed user community and management workforce is readily apparent.

Figure 1: An internal view of the Lucata Pathfinder chassis.

The Rogues Gallery is an NSF-funded novel architecture testbed designed to provide early experiences for a variety of resources [28], with training being one of its core tenets. Approaches based on near-memory computation (of which CXL is a subset), reconfigurable

https://doi.org/10.22369/issn.2153-4136/14/2/6

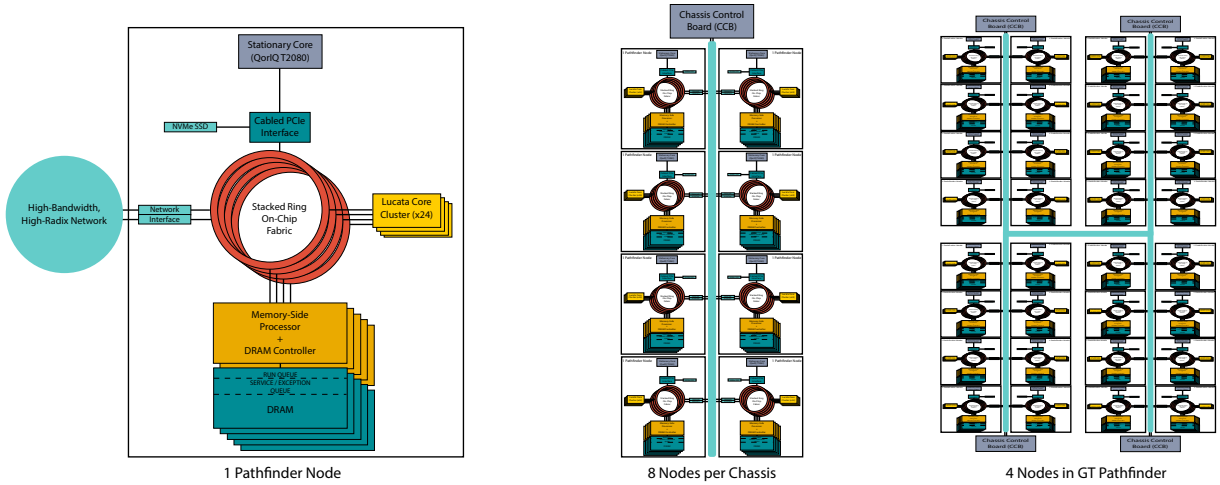


Figure 2: The architecture of the Pathfinder. Each node consists of 24 Lucata cores, managed by a stationary core, with access to shared memory and a reconfigurable network. Each chassis contains 8 nodes, with 4 chassis total in the GT Pathfinder system.

hardware, advanced networking, and neuromorphic processing are hosted, allowing students and researchers to consider how these architectures can be leveraged, either using current software capabilities or through novel algorithm design, to tackle many of the problems in HPC. In this paper, we will discuess one particular platform, the Lucata Pathfinder, and how we leverage it to educate the community, students, and future administrators, as covered in Sections 4, 5, and 6, respectively.

2 THE LUCATA PATHFINDER SYSTEM

The Pathfinder advances the capabilities of the prototype Emu Chick architecture [9], an earlier prototype of migratory thread hardware.The Pathfinder-S is based on the concept of migrating threads rather than a deep cache hierarchy to improve locality and efficiency. The GT Pathfinder system is comprised of 4 chassis, shown in Figure 4, each of which contains 8 nodes to run compute. The primary compute elements, dubbed Lucata cores, execute these migratory threads and forego a traditional memory controller in favor of eight memory side processors (MSPs) that assist in accelerating specific in-memory operations like remote writes, addition, and atomics. There are 24 Lucata cores per node, and thus 192 Lucata cores per chassis and 768 for the GT Pathfinder system.

Figure 2 provides a high-level overview of the Pathfinder architecture as a single node, chassis, and multi-chassis view. The Lucata cores are managed by "stationary cores", which use a QorIQ T2080 CPU [17], with 4 multi-threaded PowerPC e6500 cores, and 8 GB of memory; the 8 boards in each chassis are managed by a chassis control board, which uses the same processor model. The nodes and chassis are connected via a reconfigurable, high-bandwidth, high-radix network for efficient scaling in problems such as sparse matrix operations and graph analysis algorithms.

The chassis control boards and stationary cores run Yocto Linux [18], with a custom API for Lucata core execution and thread management across the system. The custom kernel provides a subset of the standard utilities found for more common platforms, such as x86 with RHEL or Ubuntu OSes, but this has yet to prove to be a barrier in delivery of an operational system.

In addition to the Pathfinder, RG also hosts x86 machines to develop, validate, and compare code against dispatch to the target hardware. These servers range from a Intel Westmere Jupyter front-end with 64 cores and 1 TB of memory for lightweight simulation by many concurrent users to a set of four Intel Ice Lake nodes with 64 cores and 512 GB of memory for benchmark performance comparison. Other available architectures such as GPUs, FPGAs, and Arm CPUs can be used for such efforts, but have not yet been explored for these purposes.

3 SCHEDULER AND SOFTWARE INFRASTRUCTURE

Pathfinder access is handled via the Slurm cluster management and job scheduling system [25]. We chose to implement the Pathfinder as a federated cluster in the Rogues Gallery for several reasons, including the dynamic private network for the compute nodes, unsupported plugins leveraged on other hardware, and highly variable configurations thoughout the cluster. The federation allows for the Pathfinder Slurm instance to utilize a unique configuration while readily accepting jobs from users on the Rogues Gallery login nodes for a smoother integration. Additional system SSH configurations allow users to directly login to the compute nodes using the global system names and going through the gateway server.

In order to program code for execution on the Pathfinder, the Cilk parallel-programming model is utilized with the Lucata API [23]. The successor to Intel's cilk/cilk++ frameworks, OpenCilk provides a simple means for users to write deterministic parallel code as an extension to the LLVM compiler. Users are encouraged to run Jupyter notebooks on Hawksbill, from which they can explore the Lucata API and Cilk programming framework, followed by launching code for scaled execution on systems like the Frozones or the actual Pathfinder system.

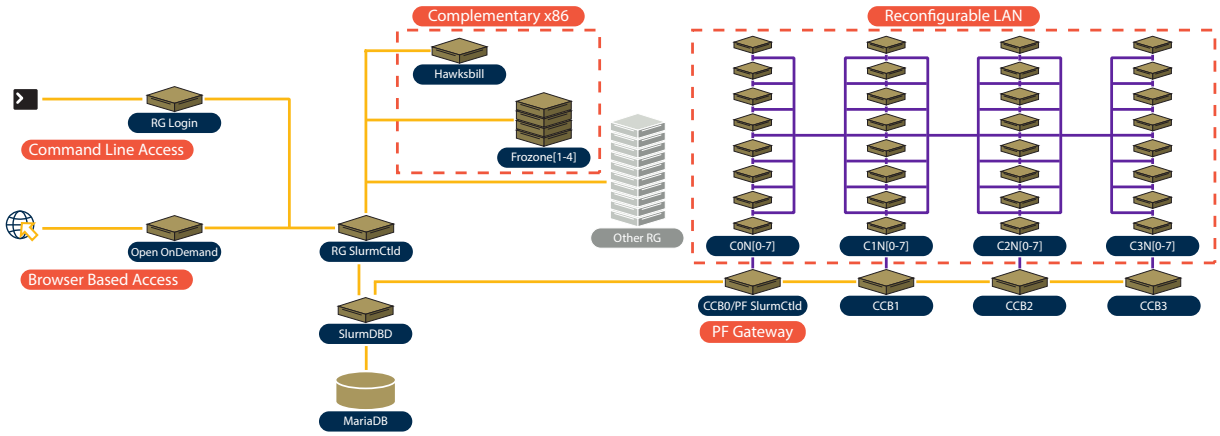


Figure 3: The Rogues Gallery is comprised of federated clusters, with a primary Slurm controller serving the majority of the hardware, including x86 servers that provide compute capabilities complementary to the Pathfinder system, and a secondary Slurm controller atop the Pathfinder and its reconfigurable LAN. Additional system-wide SSH configurations allow users to submit to either controller from common cluster access points.

4 COMMUNITY TUTORIAL EXPERIENCES

Tutorials focusing on the GT Emu Chick and Pathfinder systems have been presented at several conferences, either as components of a broader RG tutorial for the ASPLOS19 and PEARC19 conferences [19, 20], or as a dedicated presentation for the PEARC21 and HPEC22 conferences [26, 27] in an online-only format. Additionally, proposals were submitted to present tutorials at PEARC22 and SC22 but were not selected for the final program.

In all cases, the target audience was users who had some familiarity with high-performance or parallel computing but who might not be experts in computing with a system like the Lucata platform. As shown in Figure 5, the most recent HPEC22 tutorial debuted the usage of Jupyter notebooks as a mechanism for tutorial attendees to learn the basics of the Cilk-based Pathfinder workflow and to compile and run code for the Pathfinder simulator and the hardware via integrated Slurm commands. This new interface dramatically increased the number of attendees who interacted with the tutorial code samples to approximately 50% from earlier tutorials where just a few attendees downloaded and ran code samples on their laptops. This interface was created through an iterative process, which took several tutorial offerings to fully create and deploy from initial C-based code samples and slides that were used at the first tutorial back in 2019.

In general, each of these tutorials had anywhere from 10 to 25 attendees, with some outliers like PEARC22, which limited attendees to pre-registered sessions, which seemed to artifically limit the number of attendees who might join for part of the tutorial. Surveys indicated that most attendees were already working in some computing-related fields, and the more dedicated online and half-day tutorials like HPEC22 provided a more focused group of attendees.

The key lessons learned from these tutorials were as follows: 1) The user interface and required background knowledge can dramatically increase/decrease engagement by attendees. 2) Tutorial venues with limited numbers of cross-scheduled tutorials and an option for a hybrid attendance mode seemed to increase audience attendance and possibly participation. 3) Most attendees had some familiarity with HPC or parallel computing concepts but required background knowledge to fully understand the tutorial's material.

Figure 4: An earlier Rogues Gallery tutorial at Georgia Tech.

5 FUTURE COMPUTING VIP'S NEAR-MEMORY SUBTEAM

The Future Computing with the Rogues Gallery VIP course introduces students to the various architectures in the Rogues Gallery and their target applications through a vertically-integrated, project-based approach [14]. In particular, the near-memory subteam is tasked with exploring the Pathfinder architecture and its use in HPC through a series of targeted benchmarks, including

- HPCG for sparse-matrix operations [13],
- pChase for exploring memory access latency [5], and
- breadth-first search for use in graph applications [3].

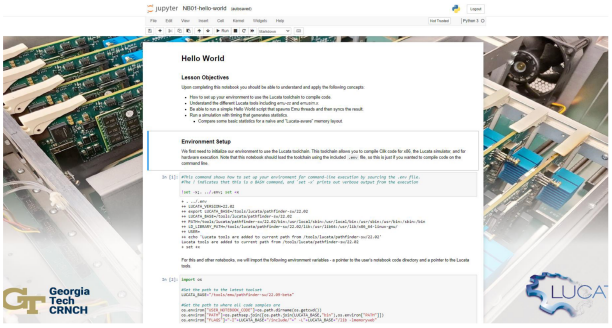


Figure 5: Screenshot from one of the tutorial demo Jupyter notebooks. To help reinforce the focus and attention of attendees, customized Jupyter styling was developed for the hands-on components.

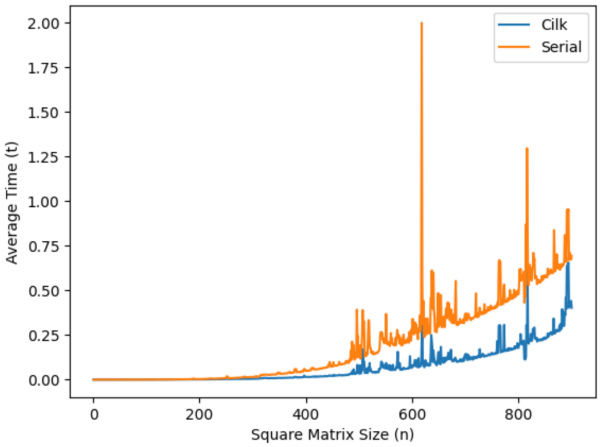


Figure 6: Benchmark comparing serial and Cilk-parallelized algorithms to perform square matrix multiplication. As with other parallel computing frameworks, the performance improvement requires sufficient problem size.

The students in the subteam focus on one of the given applications in smaller project subteams, using the general class meeting each week to address common issues with Slurm, Cilk, or the Lucata API.

To help onboard new team members, the students have employed the OpenCilk documentation to understand the parallel-programming framework, and notebooks from the RG/Pathfinder tutorials to see practical uses for executing code on the system. Additionally, they develop a rich knowledgebase through their weekly notebook entries, which collectively provide a robust documentation of technical implementation details, references, and experiences that enable new students to contribute quickly to the team. As an example, student success in comparing serial code against Cilk-programmed multiplication of square matrices can be seen in Figure 6. As the students moved from toy models to benchmarking x86 and the Pathfinder systems, they engaged the Lucata engineers for code and algorithm development, and turned to the company repositories for components like

As an added benefit to the nature of the VIP program, students are able to use the opportunity to develop projects that can apply to other courses in their computer science curriculum. In particular, two students enrolled in CS3210: Design of Operating Systems chose the Pathfinder system to explore kernel functionality by attempting to write their own DMA driver to provide better performance via lower latency memory access for the Lucata cores, as depicted in Figure 7. As with the benchmark efforts, the students worked heavily alongside Lucata developers to understand the workflow to build the kernel from source. Although the semester ended with an incomplete kernel, as the networking components built incorrectly, the students gained considerable knowledge with regards to low-level activities to advance HPC capabilities on novel architectures.

Perhaps one of the biggest unexpected challenges encountered by students from this subteam was contention with Lucata development cycles. A combination of scheduling conflicts, in which the engineers had reserved time on the system and rendered it unavailable for student use, and a lack of understanding of the local network configuration, which caused problems if student code was built for a different setup, created disruptions throughout the semesters. Fortunately, both the Lucata personnel and students were patient, and both groups graciously considered the opportunity as a learning experience from which workflows and system tools could be improved.

6 POTENTIAL PIPELINE FOR NEXT-GENERATION ADMINISTRATION AND FACILITATION

A recent initiative within the Rogues Gallery has been the hiring of a student worker to develop administrative utilities such as a production solution to publish the Pathfinder system configuration and facilitate access among the Lucata developers and research users. Unlike student research employees, whose work may utilize only familiar academic tools, or student administrative employees, whose work may only include a very targeted activity within the broader infrastructure, administrative employment on a near-production system such as the Pathfinder provides broad exposure to both the research workflows and the standard tools required to manage the systems. In this way,

The first project addressed by the student employee focuses on clarifying the complexity of the Lucata Pathfinder scheduling and reconfiguration mechanisms. Notably, the Pathfinder 4 chassis system can be reconfigured in up to eight valid configurations that include combinations of single-node, multi-node but single-chassiss, and multi-chassis.

To work around this complexity, the initial scheduling mechanism for the Pathfinder used a Google Calendar to block off time with specific configuration details. In 2022, Slurm was built for the Pathfinder system and was set up as a federated instance with the rest of the Rogues Gallery testbed. However, it was noted that the calendar interface seemed somewhat more intuitive to some of the engineers and developers working with the system.

Figure 8 shows a new approach that attempts to combine the best of both worlds by using Slurm to control and reconfigure

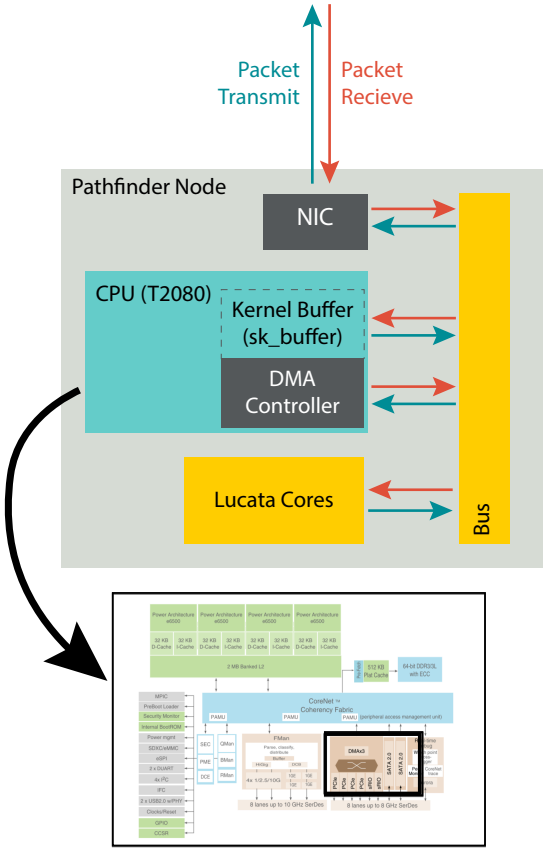


Figure 7: Schematic showing how a DMA driver would improve data throughput for the Lucata cores by bypassing processing in the stationary cores entirely. Despite the DMA controller being present on the T2080 CPU, the kernel currently provides no explicit support to move data directly between Lucata cores and DRAM.

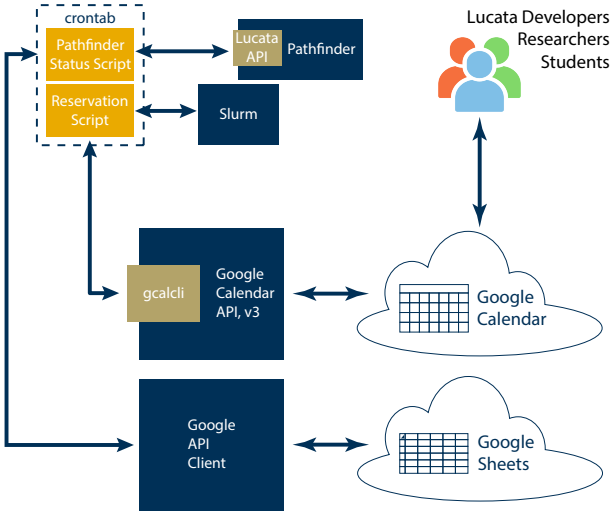


Figure 8: Automated status update.

Chassis 0	Logical Nodes	Chassis 1		Chassis 2		Chassis 3	
n0	sn0	n0	sn8	n0	sn0	n0	sn0
n1	sn1	n1	sn9	n1	sn1	n1	sn0
n2	sn2	n2	sn10	n2	sn2	n2	sn0
n3	sn3	n3	sn11	n3	sn3	n3	sn0
n4	sn4	n4	sn12	n4	sn4	n4	sn0
n5	sn5	n5	sn13	n5	sn5	n5	sn0
n6	sn6	n6	sn14	n6	sn6	n6	sn0
n7	sn7	n7	sn15	n7	sn7	n7	sn0
C0 Status	Multinode C0-C1	C1 Status	Multinode C0-C1	C2 Status	Multinode C2	C3 Status	Single node

Figure 9: Cluster Configuration Example with Google Sheet.

the system and Google Calendar and Google Sheets to help share and update the state of the system. The Google Calendar API is used to read the state of the Pathfinder system as configured via Slurm and report it to a calendar that can be checked by Lucata engineers, researchers, and students. Likewise, Figure 9 shows the current status of the Pathfinder system with this example showing two chassis as combined "multi-chassis" instances in blue, a single-chassis instance in green, and single nodes in grey.

One potential enhancement to this setup would be to actually interpret and allow inputs from the Google Calendar to drive Slurm jobs for the Pathfinder or even eventually for other less complicated platforms. A key note here is that while the Lucata system might seem like a unique and one-off type system with a very bespoke scheduler, we are starting to see similar setups in systems with NVIDIA Multi-Instance GPUs (MIGs) which currently are not compatible with current Slurm installations, due to their reconfigurability. It is likely that other cutting edge systems could benefit from thinking about user interfaces in similar fashions to improve both utilization and user experiences.

Although we recognize that asking student employees to "fix" interface issues like this for a unique system has some shortcomings, we feel that near-production system administration fills a specific need in the training process. Broader-reach workforce development initiatives churn out larger numbers of trained individuals, but with less depth of knowledge due to the nature of the training exercises and limited time for the training. At the other end of the spectrum, student researchers managing isolated systems may become experts in the systems they manage, but are less likely to use industry-standard tools in accordance with best practices. By working on a near-production system, students are exposed to real-world problems and the tools used to monitor, mitigate, and manage them, providing a nice balance between richness and exposure. Furthermore, as multiple near-production systems may exist on a given campus, such a workforce model can broaden its impact.

7 CONCLUSIONS

The deployment of extremely novel and unique near-memory migratory thread systems like the Lucata Chick and Pathfinder has provided both new opportunities for training as well as challenges from an educational and maintenance standpoint.

From an educational perspective, the Lucata systems have provided an extremely compelling high-performance platform that has been used to support local coursework at Georgia Tech, tutorial development at architecture and HPC conferences, and proving

grounds for testing of new tools and techniques to support next-generation HPC ecosystems. Figuring out the best way to engage students and non-experts has required iterative and collaborative work with GT researchers as well as strong support from the vendor.

Some takeaways that could improve this process would be an earlier focus on smarter tooling and interfaces to support new users including Slurm scheduling instead of ad-hoc scheduling, automated scripts to reconfigure the system, and Jupyter notebooks for teaching and tutorials. Furthermore, matching tutorial materials and scope with the correct conference audience has proven to be important to get good turn out and engagement with such a unique platform.

Overall, the engagement with the community and collaboration with local researchers and the vendor has provided Georgia Tech with an incomparable resource and experience for teaching and researching high-performance computing applications. Future efforts will focus on extending the scope of applications that can be run using tutorial-style notebooks and in supporting added tools and features to improve the user experience when using the Pathfinder system as part of the testbed.

ACKNOWLEDGMENTS

This research was supported by the NSF MRI award #1828187: “MRI: Acquisition of an HPC System for Data-Driven Discovery in Computational Astrophysics, Biology, Chemistry, and Materials Science." Additionally, this research was supported in part through research infrastructure and services provided by the Rogues Gallery testbed hosted by the Center for Research into Novel Computing Hierarchies (CRNCH) at Georgia Tech. The Rogues Gallery testbed is primarily supported by the National Science Foundation (NSF) under NSF Award Number #2016701. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the author(s), and do not necessarily reflect those of the NSF.

REFERENCES

[1] David Akin, Mehmet Belgin, Timothy A. Bouvet, Neil C. Bright, Stephen Harrell, Brian Haymore, Michael Jennings, Rich Knepper, Daniel LaPine, Fang Cherry Liu, Amiya Maji, Henry Neeman, Resa Reynolds, Andrew H. Sherman, Michael Showerman, Jenett Tillotson, John Towns, George Turner, and Brett Zimmerman. 2017. Linux Clusters Institute Workshops: Building the HPC and Research Computing Systems Professionals Workforce. In *Proceedings of the HPC Systems Professionals Workshop (HPCSYSPROS'17)*. Association for Computing Machinery, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3155105.3155108

[2] A. Antonov, N. Popova, and Vl. Voevodin. 2018. Computational science and HPC education for graduate students: Paving the way to exascale. *J. Parallel and Distrib. Comput.* 118 (2018), 157–165. https://doi.org/10.1016/j.jpdc.2018.02.023

[3] David A. Bader, John Feo, John Gilbert, Jeremy Kepner, David Koester, Eugene Loh, Kamesh Madduri, Bill Mann, and Theresa Meuse. 2007. HPCS Scalable Synthetic Compact Applications #2 Graph Analysis (SSCA#2 v2.2 Specification). online. http://www.graphanalysis.org/benchmark/HPCS-SSCA2_Graph-Theory_v2.2.pdf

[4] Fabio Banchelli and Filippo Mantovani. 2018. Filling the gap between education and industry: evidence-based methods for introducing undergraduate students to HPC. In *2018 IEEE/ACM Workshop on Education for High-Performance Computing (EduHPC)*. 41–50. https://doi.org/10.1109/EduHPC.2018.00008

[5] Tim Besard and Steven Noonan. 2013. pChase. github. https://github.com/maleadt/pChase

[6] Caughlin Bohn. 2022. HPC Outreach and Education at Nebraska. In *Practice and Experience in Advanced Research Computing (PEARC '22)*. Association for Computing Machinery, New York, NY, USA, Article 63, 4 pages. https://doi.org/10.1145/3491418.3535128

[7] The Compute Express Link Consortium. 2022. About CXL. https://www.computeexpresslink.org/about-cxl

[8] Intel Corporation. 2022. Migration from Direct-Attached Intel Optane Persistent Memory to CXL-Attached Memory. Technical Brief. https://www.intel.com/content/dam/www/central-libraries/us/en/documents/2022-11/optane-pmem-to-cxl-tech-brief.pdf

[9] Timothy Dysart, Peter Kogge, Martin Deneroff, Eric Bovell, Preston Briggs, Jay Brockman, Kenneth Jacobsen, Yujen Juan, Shannon Kuntz, Richard Lethin, Janice McMahon, Chandra Pawar, Martin Perrigo, Sarah Rucker, John Ruttenberg, Max Ruttenberg, and Steve Stein. 2016. Highly Scalable Near Memory Processing with Migrating Threads on the Emu System Architecture. In *2016 6th Workshop on Irregular Applications: Architecture and Algorithms (IA3)*. 2–9. https://doi.org/10.1109/IA3.2016.007

[10] SIGHPC Education. 2023. Sixth Workshop on Strategies for Enhancing HPC Education and Training (SEHET23). https://sighpceducation.acm.org/events/sehet23_cfp/

[11] EduHPC. 2022. EduHPC-22: Workshop on Education for High-Performance Computing. https://tcpp.cs.gsu.edu/curriculum/?q=eduhpc22

[12] EduPAR. 2023. EduPar-23: 13th NSF/TCPP Workshop on Parallel and Distributed Computing Education. https://tcpp.cs.gsu.edu/curriculum/?q=edupar23

[13] Michael Allen Heroux and Jack. Dongarra. 2013. Toward a new metric for ranking high performance computing systems. (6 2013). https://doi.org/10.2172/1089988

[14] Aaron Jezghani, Jeffrey Young, Will Powell, Eric J. Coulter, and Ronald Rahaman. 2023. Future Computing with the Rogues Gallery. [online] EduPar23 preprint. https://tcpp.cs.gsu.edu/curriculum/?q=system/files/Jezghani%20paper.pdf

[15] Astera Labs. 2022. Compute Express Link CXL 2.0 Specification Released the Big One. News Brief. https://www.asteralabs.com/news/compute-express-link-cxl-2-0-specification-released-the-big-one/

[16] Julia Mullen, Lauren Milechin, and Dennis Milechin. 2021. Teaching and learning HPC through serious games. *J. Parallel and Distrib. Comput.* 158 (2021), 115–125. https://doi.org/10.1016/j.jpdc.2021.07.014

[17] NXP. 2023. QorIQ T2080 and T2081 Multicore Communications Processors. [online] Product Page. https://www.nxp.com/products/processors-and-microcontrollers/power-architecture/qoriq-communication-processors/t-series/qoriq-t2080-and-t2081-multicore-communications-processors:T2080

[18] The Yocto Project. 2023. Yocto Project (webpage). https://www.yoctoproject.org/

[19] Jason Riedy, Will Powell, Jeffrey Young, Tom Conte, and Vivek Sarkar. 2019. Rogues Gallery: Addressing Post-Moore Computing. https://crnch-rg.gitlab.io/pearc-2019/

[20] Jason Riedy, Jeffrey Young, Tom Conte, and Vivek Sarkar. 2019. Programming Novel Architectures in the Post-Moore Era with the Rogues Gallery. https://crnch-rg.gitlab.io/asplos-2019/

[21] Semir Sarajlic, Neranjan Edirisinghe, Yuriy Lukinov, Michael Walters, Brock Davis, and Gregori Faroux. 2016. Orion: Discovery Environment for HPC Research and Bridging XSEDE Resources. In *Proceedings of the XSEDE16 Conference on Diversity, Big Data, and Science at Scale (XSEDE16)*. Association for Computing Machinery, New York, NY, USA, Article 54, 5 pages. https://doi.org/10.1145/2949550.2952770

[22] Semir Sarajlic, Neranjan Edirisinghe, Yubao Wu, Yi Jiang, and Gregori Faroux. 2017. Training-Based Workforce Development in Advanced Computing for Research and Education (ACoRE). In *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact (PEARC17)*. Association for Computing Machinery, New York, NY, USA, Article 71, 4 pages. https://doi.org/10.1145/3093338.3104178

[23] Tao B. Schardl and I-Ting Angelina Lee. 2023. OpenCilk: A Modular and Extensible Software Infrastructure for Fast Task-Parallel Code. In *Proceedings of the 28th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming (PPoPP '23)*. Association for Computing Machinery, New York, NY, USA, 189–203. https://doi.org/10.1145/3572848.3577509

[24] MIT Supercloud. 2022. Scaling HPC Education. https://supercloud.mit.edu/scaling-hpc-education

[25] 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.

[26] Jeffrey Young, Patrick Lavin, Jason Riedy, Srinivas Eswar, Janice McMahon, Aaron Jezghani, Will Powell, and Darryl Bailey. 2022. Exploring Graph Analysis for HPC with Near-Memory Accelerators. https://doi.org/10.5281/zenodo.7117251

[27] Jeffrey Young, Semir Sarajlic, Will Powell, Janice McMahon, and Jason Riedy. 2021. Lucata Pathfinder-S Tutorial: Next-generation Computationwith the Rogues Gallery. https://crnch-rg.gitlab.io/pearc-2019/

[28] Jeffrey S. Young, Jason Riedy, Thomas M. Conte, Vivek Sarkar, Prasanth Chatarasi, and Sriseshan Srikanth. 2019. Experimental Insights from the Rogues Gallery. In *2019 IEEE International Conference on Rebooting Computing (ICRC)*. 1–8. https://doi.org/10.1109/ICRC.2019.8914707

Orchestrating Cloud-supported Workspaces for a Computational Biochemistry Course at Large Scale

Gil Speyer
Arizona State University
Tempe, AZ
speyer@asu.edu

Neal Woodbury
Arizona State University
Tempe, AZ
laserweb@asu.edu

Arun Neelicattu
CR8DL, Inc.
Goodyear, AZ
arun.neelicattu@cr8dl.ai

Aaron Peterson
CR8DL, Inc.
Goodyear, AZ
aaron.peterson@cr8dl.ai

Greg Schwimer
CR8DL, Inc.
Goodyear, AZ
schwim@cr8dl.ai

George Slessman
CR8DL, Inc.
Goodyear, AZ
g@cr8dl.ai

ABSTRACT

A joint proof-of-concept project between Arizona State University and CR8DL, Inc., deployed a Jupyter-notebook based interface to datacenter resources for a computationally intensive, semester-length biochemistry course project. Facilitated for undergraduate biochemistry students with limited high-performance computing experience, the straightforward interface allowed for large scale computations. As the project progressed, various enhancements were identified and implemented.

KEYWORDS

Cloud computing, AlphaFold, Computing education

1 INTRODUCTION

In early 2022, CR8DL, Inc. launched a new concept in datacenter services, offering easily accessible sandbox services intended to empower large-scale computation. In order to promote this concept with real-world success stories, Arizona State University (ASU) was approached to solicit research or educational computational challenges. A compelling combination of both of these was readily identified: a senior level computational biochemistry course, "Modern Approaches to Biochemical Data Analysis."

The CR8DL resource addressed two critical challenges in the course. First, students would be assigned a semester-long project involving repeated prediction of protein structures from mutated sequences using the Alphafold software [4]. Depending on the length of the sequence, the resulting structure inference could be accelerated employing graphical processing units (GPUs). Second, undergraduate biochemistry students in the course would not have exposure to campus cluster computational resources. To provide services like this for a class would have been expensive for the University in both in terms of time as well as financially (set up, student training, operations staff, time-on-system availability, and so on). Further, the use of a traditional command-line interfaces and use of a job scheduler on University systems pose a steep barrier to entry for a large percentage of the students.

In contrast, CR8DL's services remove these barriers by providing on-demand compute resources that easily fit the purposes of the course. This is provided via a straightforward interface that is easy for students naïve to computer programming to navigate. The Jupyter notebook interface, with its file upload/download capability and persistent pages accessible through a browser, allowed for the students to access these resources with minimal knowledge of the underlying computer system architecture or operating system.

This work emphasizes solely the integrated technologies made available to the students in the course in order to enable an enhanced activity, namely a semester project. No course redesign occurred to accommodate this project. Student information was not used for this study, nor was student feedback solicited.

2 ENABLING HPC TO BIOCHEMISTRY UNDERGRADUATES

For each student, the project entailed starting with a library of one thousand randomly generated sequences of the same size as a target protein domain, ARR10, a plant transcription regulator (Protein Data Base entry 1IRZ) [2]. The goal of the project was to see if they could start from a random library of sequences and create a protein domain with the same alpha carbon backbone structure as the ARR10 domain. They did this via an iterative process. In each cycle they used an algorithm that compared the structure of each generated sequence to the target backbone structure, selected the best structure, mutated it, generated 100 variants of this amino acid sequence, uploaded these to a compute cluster, ran AlphaFold to predict their resulting structure, performed post-processing of these structures involving downloading the results for visual and algorithmic assessment. The cycle was repeated about 10 times. A workflow, showing the iterated steps, including preprocessing and postprocessing steps run on the ASU supercomputer Agave is shown in Figure 1.

Throughout this process the student had to make a number of decisions. Not only were they looking for the best fit of the backbone to the target, they were considering the quality of prediction of alphafold as determined by the predicted local distance difference test (pLDDT) score. They also had to decide how many mutations

https://doi.org/10.22369/issn.2153-4136/14/2/

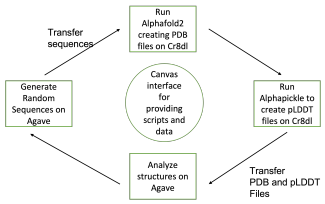


Figure 1: Workflow diagram, showing the iterated steps.

to make each round and then visually decide if the key structural elements were coming into place on the best performers.

By the end of the semester, substantial progress was made by many of the students in using this random mutation followed by selection approach to generating a backbone structure similar to the target. Figure 2 shows the overlap of the target structure (red) and the structure generated after twelve rounds of random mutation followed by selection. The pLDDT was 67.5 (borderline believable) and the root mean squared error (RMSE) between structures was 3.47 based on an angstrom distance scale and focused on the central region ignoring the unstructured portions of the target.

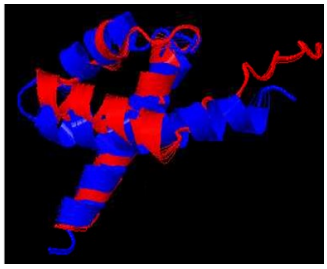
Alphfold Overlap.jpg 

Figure 2: Overlap between the target alphacarbon backbone structure and the generated structure after twelve rounds of random mutation and selection using Aphafold2 to generate structure data and pLDDT scores. A separate algorithm (in Matlab) was used to generate the RMSE scores. Students used the pLDDT, RMSE and visual inspection of overlapped structures to select which of the 100 mutations in each structure would advance to the next round.

2.1 Compute and Software Resources

The cloud environment provided for this project allowed all students in the class to concurrently share a high-performance compute infrastructure. Resources such as high-capacity storage, CPU, and the latest GPU compute resources were provided. The software components "over the top" of these resources meant that at no time was a user expected to know how to administer or otherwise program on it. The underlying compute infrastructure used for this project consisted of a number of physical "servers" in a cluster configuration. Each user was provided guaranteed, full time access to "slices" of these resources for their own use. Each slice was composed of:

- CPU: 8 cores
- RAM: 32 GB
- GPU: Nvidia A100-SXM4-80G - 1 "MiG" instances, with a "3g.40gb" profile (3 slices 40GB RAM)
- Storage: Unlimited. No user used more that 75GB during the project

These resources were provided on-demand, such that when a user's interaction was completed, they were reclaimed to a pool and made available to other users on the system. At no point was a user denied these resources as there was ample capacity on the entire system for all.

2.2 User access and accounts

Student and instructor access was provided via a web-based interface. Authentication was enabled by way of single sign on (SSO) federation with ASU systems. This allowed users to use their own existing accounts, controlled by ASU's own infrastructure services.

2.3 Input data and executables

The entire AlphaFold database set from the public AlphaFold repository was replicated to a shared storage cluster. These databases were made available to all users via a read-only directory structure represented in their main home directory.

The primary path of job execution was provided as a single Jupyter notebook. Within this notebook, code was placed to automate the job processing. The goal of this code was to abstract much of the complexity from the user, while also leveraging the high-performance compute capabilities of the overall system.

2.4 Workflow

Upon login with a web browser, the users were presented a standard Jupyter Lab interface. Using this interface, users would implement a simple, standardized workflow to launch AlphaFold and calculate the results for further analysis:

(1) Upload fasta files

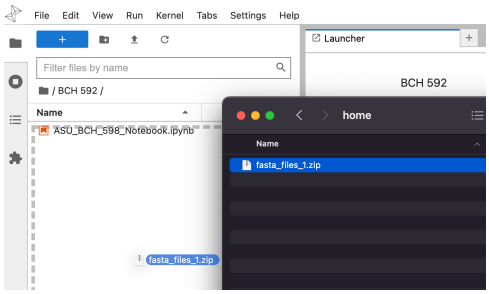


Figure 3: User uploads their fasta files via a drag and drop interface provided by Jupyter Lab.

(2) Launch the provided Jupyter Notebook and run it
(3) Input the directory of fasta files in the provided notebook interface widget
(4) Select "Run Alphafold"
(5) User collects resulting output for visual analysis.

Calculated results were preserved within the user's personal directory for the entire duration of the project, only removed by the

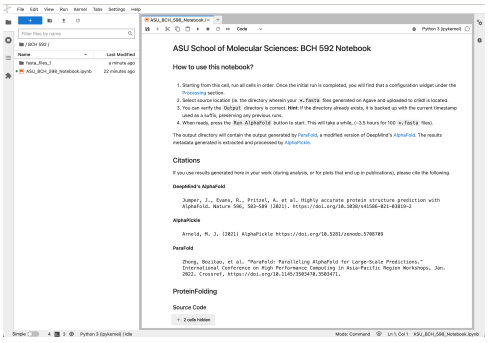


Figure 4: Notebook interface.

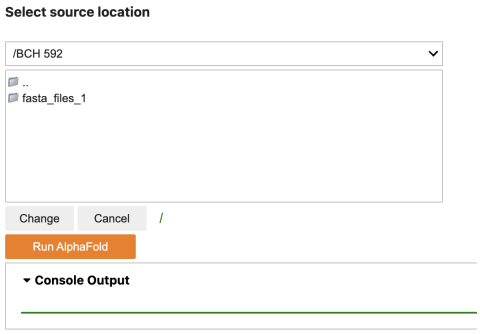


Figure 5: User selects the source fasta file and is then presented with a Run AlphaFold button interface. An output "console" below provides the user with feedback as the job is run.

user's own actions. The purpose of this was to encourage iterative engagement with the results.

2.5 Output data post-processing and iteration

Alphafold generates five predicted structures for the input sequence in a pdb format, easily rendered in a viewer. In this way, students could overlay these structures and inspect them visually. Using a Matlab program, RMSE between the atomic positions could be calculated [3]. Finally, the Alphapickle python script provided students with a quantitative and visual means, via pLDDT scores, to assess confidence in output structures [1].

3 REFINEMENTS

Through the semester, the students continued to iterate with the AlphaFold pipeline, uploading new input sets of fasta files representing one hundred new mutations based on the previous runs. As these jobs were run, the CR8DL team tracked performance and investigated opportunities for improvement. Among such refinements were the abstraction of interaction with Python code to a more graphically driven interface as shown in various figures in this document, as well as a restructuring of the way in which the user was expected to interact with the filesystem. This led to a general improvement in workflow, enabling users to be more efficient in their efforts.

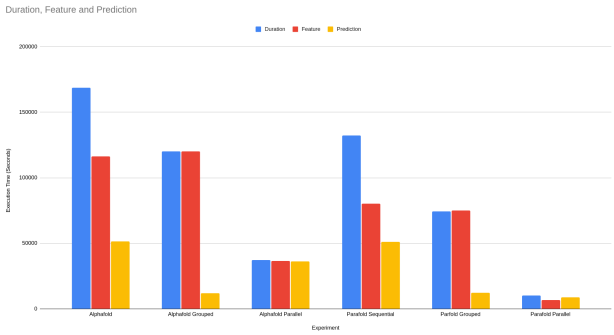


Figure 6: Alphafold and Parafold job performance comparisons for 100 fasta files across three different processing strategies: "Base", "Grouped", and "Parallel". "Base" is the as-provided, out-of-the-box implementation. The "Grouped" strategy is enhanced by avoiding repeated data loads. "Parallel" execution orchestrates separate, concurrent CPU ("features") and GPU ("prediction") steps. Parafold further exploits maximum parallelism in the feature generation, resulting in improved overall performance. Blue = Overall Duration, Red = Feature Generation Duration, Yellow = Prediction Duration for each job type.

3.1 AlphaFold to ParaFold

Alphafold incurs a time intensive workload, even when presented with a high-performance infrastructure similar to what was provided by the CR8DL team for this project. It was realized early on that by splitting CPU and GPU components of the AlphaFold pipeline, available parallel resources –which are abundant in a datacenter– can be employed to accelerate the workflow. Aptly named, the Parafold utility further parallelizes these computations by exploiting the similarity across the multiple sequences to optimize resources both for the CPU-based feature determination and the GPU-based inference steps [5]. Figure 6 illustrates the enhancements observed for a specific data set, exceeding an order of magnitude acceleration.

4 CONCLUSION AND FUTURE WORK

Continuing efforts are under way to further streamline the user's experience when engaging with complex coursework that leverages high-performance computing resources. User interface improvements, pre-determined visualization outputs, and the ability to run even larger workloads without a need for software development experience are all part of our next iterations. The employment of multiple platforms, despite the vast majority of computation on the CR8DL resource, could be alleviated. Future implementations of the course look to porting tools to remove dependence on the academic Matlab license and co-locating a common folder for project materials for the students.

ACKNOWLEDGEMENTS

The authors acknowledge CR8DL, Inc., and Research Computing at Arizona State University for providing resources that have contributed to the results reported within this paper.

REFERENCES

[1] M. J. Arnold. 2021. mattarnoldbio/alphapickle: v1.4.1. https://doi.org/10.5281/zenodo.5752375

[2] Kazuo Hosoda, Aya Imamura, Etsuko Katoh, Tomohisa Hatta, Mari Tachiki, Hisami Yamada, Takeshi Mizuno, and Toshimasa Yamazaki. 2002. Molecular Structure of the GARP Family of Plant Myb-Related DNA Binding Motifs of the Arabidopsis Response Regulators. *Plant Cell* 14, 9 (Sept 2002), 2015–2029. https://doi.org/10.1105/tpc.002733

[3] The MathWorks Inc. 2022. MATLAB version: 9.13.0 (R2022b). https://www.mathworks.com

[4] John M. Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Zídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andy Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David A. Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature* 596 (2021), 583 – 589.

[5] Bozitao Zhong, Xiaoming Su, Minhua Wen, Sichen Zuo, Liang Hong, and James Lin. 2021. ParaFold: Paralleling AlphaFold for Large-Scale Predictions. arXiv:q-bio.BM/2111.06340

November 2023

Volume 14 Issue 2

ISSN 2153-4136 (online)