December 2022

Volume 13 Issue 2

JOCSE

Journal Of Computational Science Education

Promoting the Use of Computational Science Through Education

ISSN 2153-4136 (online)

Volume 13, Issue 2 **December 2022**

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 13 Issue 2

Steven I. Gordon
Editor
The Ohio State University
Columbus, OH
gordon.1@osu.edu

FOREWORD

In this issue, we combine submissions from the PEARC22 Third Workshop on Strategies for Enhancing HPC Education and Training, one paper from the SC22 Ninth Workshop on Best Practices for HPC Education and Training, and one additional paper. The next issue will feature the remaining articles from the workshop at SC22 and other papers.

Brashear et al. describe an informal, modular course aimed at students in economics with little or no previous programming experience. The modular course uses the R package "learnr" to present topics on programming in the R environment. The course was offered virtually over a two-week period. Student feedback should allow for improvements to the course in the future.

The paper by Feister and Blackwood summarizes the integration of HPC related topics and skills into the undergraduate curriculum at an Hispanic-Serving Institution. They implemented a framework that provides multiple opportunities for students to learn HPC modeling and simulation skills. Those include the integration of HPC related projects into existing courses, student participation in visualization and cluster competitions, participation in faculty research projects, and a capstone project.

Gordon, Lathrop, and Kramer provide a summary of the impacts of the Blue Waters Fellowship Program. They describe the program and the changes that were made to improve the outcomes for the fellows. They also suggest how lessons learned from the program could be implemented in other efforts aimed at strengthening the workforce engaged in computational research.

In their article, Knuth et al. present the structure of a multi-tiered support system that is part of the NSF supported ACCESS project. The first two tiers of support include access to easy-to-use tools and services and a self-help knowledge base. Tiers three and four provide short-term technical assistance to projects using student and staff expertise, respectively.

Kramer et al. describe the range of training and education efforts made as part of the Blue Waters project. The education efforts included the development of curriculum materials, the virtual school for computational science, and education allocations for classroom instruction. The project provided training through a series of petascale computing institutes, an international summer school, hackathons, and webinars. Students participated in an extensive internship program as well as the fellowship program described in a previous article in this issue. The article provides details on each activity, lessons learned, and the overall impacts on the community.

The last article by Young et al. describes the use of novel computer architectures used in several settings at Georgia Tech. They provide an overview of the architectures and the issues that needed to be addressed to integrate them for student use.

Finally, I would like to end with a personal note as this is the last issue of JOCSE where I will be acting as editor. I would like to thank the Shodor Education Foundation for the opportunity to start and nurture this journal. It has provided a critical outlet for faculty engaged in computational science education and students who have undertaken a wide range of research experiences. In particular, I would like to thank Dr. Robert Panoff for his support and Jennifer Houchins and Aaron Weeden for the technical assistance in compiling the journal issues. Dr. David Joiner from Kean University has agreed to take on the editor's role and will continue to provide the leadership necessary to continue the journal.

Bridging Data Science Programming with Advanced Formal Coursework

Wesley A. Brashear
HPRC†
Texas A&M University
College Station, TX US
wbrashear@tamu.edu

Zhenhua He
HPRC†
Texas A&M University
College Station, TX US
happidence1@tamu.edu

Richard Lawrence
HPRC†
Texas A&M University
College Station, TX US
rarensu@tamu.edu

Dhruva K. Chakravorty
HPRC†
Texas A&M University
College Station, TX US
chakravorty@tamu.edu

Tatevik Sekhposyan
Department of Economics
Texas A&M University
College Station, TX US
tsekhposyan@tamu.edu

Margaret L. Carpenter
Department of Economics
Texas A&M University
College Station, TX US
marge@tamu.edu

Honggao Liu
HPRC†
Texas A&M University
College Station, TX US
honggao@tamu.edu

ABSTRACT

In order to fulfill the needs of an evolving job market, formal academic programs are continuously expanding computational training in traditional discipline-specific courses. We developed an informal, twelve contact-hour course tailored for economics students entering a computationally rigorous graduate-level course to help mitigate disparities in computing knowledge between students and prepare them for more advanced instruction within the formal setting. The course was developed to teach the R programming language to students without assuming any prior knowledge or experience in programming or the R environment. In order to allow for ease of implementation across various training approaches, the course was modularized with each section containing distinct topics and learning objectives. These modules can be easily developed as independent lessons so that discipline-specific needs can be addressed through inclusion or exclusion of certain topics. This implementation used the R package 'learnr' to develop the course, which rendered a highly extensible and portable interactive Shiny document that can be deployed on any system on which RStudio is installed. The course is offered as a series of interactive sessions during which students are led through the Shiny notebook by an instructor. Owing to its structure, it can be offered as an asynchronous web-based set of tutorials as well.

DOI: https://doi.org/10.22369/issn.2153-4136/13/2/1

Keywords

R, Economics, Cybertraining, Shiny, Interactive computing

1. INTRODUCTION

The R programming language has continued to gain in popularity over the last several years [1]. It is widely used across numerous disciplines and is a common staple of scientific workflows. One of the reasons for this increase in popularity is the continuous development and publication of packages that extend the utility of the R programming language as a data science tool. While some of these packages are utilized by a wide range of users (e.g., ggplot2), others are restricted to specific disciplines or areas of research and require extensive education within those disciplines for users to understand and utilize the increased functionality. This creates a gap for potential users who have the requisite knowledge and desire to utilize discipline-specific packages but lack the foundational knowledge necessary for working in the R programming environment. Responding to the need to bridge the gap for students who are both new to R programming and require the use of these discipline-specific R packages, we created a twelve-hour non-credit-bearing workshop for students to prepare them for future formal courses that utilize R. Here, we focus on students enrolled in the Graduate program offered by the Department of Economics at Texas A&M University.

An early challenge faced when developing introductory-level curricula is keeping students engaged while learning the foundational concepts required for expansion into discipline-specific computational knowledge [12, 11]. The design of a course plays a large role in student engagement and should include a variety of teaching strategies; including lectures, guided hands-on examples, and independent exercises. Furthermore, teaching in incremental modules with independent learning objectives can foster a sense of productive progress for the students. Another

† High Performance Research Computing

means to increase engagement is to create guided or independent exercises that utilize data relevant to the discipline of the target audience (e.g., economic data) or to shared personal experiences (e.g., enrollment at a particular university).

Texas A&M High Performance Research Computing has abundant experience developing informal courses and pedagogical approaches tailored for specific user groups [3–10]. These courses cover a wide scope of topics including introductions to our HPC systems and Linux basics; programming in Python, R, Matlab, and Pearl; and bioinformatics. After the success of our recent twenty-contact-hour training sessions tailored for teaching Python to economics graduate students [9], we were invited to develop a similar course for the R programming language.

2. METHODS

The topics covered in the class (Table 1) were determined through conversations with faculty members of the Texas A&M Department of Economics, analysis of course syllabi, interviews with students, and surveys of online tutorials for R. Most of the students taking the course were doing so in preparation for an accredited course covering economic forecasting in the following semester. Others did so to demonstrate proficiency in the R programming language to prospective employers. The scope of the course was selected with the students' future coursework in mind. Taking a departure from our previous work that aimed to develop proficiency in Python programming (10), here the learning objectives and outcomes were geared toward ensuring that a student develops proficiency in relevant skills. As such, a guiding principle was to introduce topics with the view of ensuring high adoption among students so that they could use them during the semester to address domain-specific problems. For example, the students would be expected to be ready to learn rmarkdown and statistical methods during the workshop and later rely on this knowledge during their future coursework.

The course's structure builds on the HPRC short course program [4], and leverages the strengths of our Python training program for graduate students [9]. The course was developed for a virtual setting with scaffolding to ensure seamless transition to an in-person learning environment. The course consists of twelve contact hours taught over two days, broken into 50-minute increments. The format of the modules included a brief introduction to each topic (typically through Google Slides) followed by a hands-on, instructor-guided/led example and then independent exercises. Students were given short, one-question quizzes throughout each day to track participation. We used the package 'learnr' [1] to create a Shiny interactive document that allows students to create, edit, and run small code chunks in R. This format allows for instructional guidelines and helpful information to be displayed to students alongside windows for running code. This highly portable format can be deployed across any system with RStudio, including personal computers, on-campus research computing systems, and RStudio Cloud. Students were given the option of running the learnr document for this course on their own computer, in RStudio Cloud, or through Texas A&M's Virtual Open Access Lab (VOAL). VOAL allows Texas A&M students to connect remotely to a Windows machine which features a suite of installed software, including RStudio.

At the end of the course, students were asked to complete a brief survey to provide feedback on the course, including the topics covered, the difficulty of the lessons, course pacing, and RStudio access. Students who took the formal course for which this workshop was tailored were given a survey following completion of the formal course in May.

Table 1a. Topics covered during the R programming for Economics Majors workshop.

Section	Topics Covered
Using the Interactive R Document	Introduction to RStudio IDE
	Introduction to the learnr format
	R as a calculator
Data types	Types of data in R
	Checking data types with class(), str(), is.<type>()
	Conditional operators
	Converting data types
Variables	Assigning variables with <- and =
	print()
Built-in Functions	Introduction to functions
	Getting help with ? and ??
Vectors	Data types and vectors
	Creating vectors with seq() and c()
	Vector arithmetic
	Naming elements
	Subsetting vectors
	Conditional and pattern selection of vectors
Loops	for loops
	Iterating over elements
	while loops
Matrices	Data types and matrices
	Creating a matrix by combining vectors
	Adding row and column names
	Subsetting matrices
	Matrix arithmetic

Table 1b. Topics covered during the R programming for Economics Majors workshop (cont.).

Section	Topics Covered
Data frames	Introduction to R built-in data frame
	head(), tail(), and str() functions with data frames
	Creating data frames with matrices and vectors
	Adding rows and columns
	Subsetting data frames
	Re-ordering data frames
	Read in external files to data frame
Data visualization	Base plotting functions in R
	Scatter plots and histograms with plot()
	Plotting with ggplot2
	Aesthetics
	Plot titles and axis labels
	Working with multiple layers
	Point labels
	Histograms with ggplot2

3. RESULTS

The course was offered in January 2022, a week before the start of the formal class. 75 students registered for the workshop, and tracked participation ranged from 53 to 73 students throughout the duration of the course. Post-course analyses found that out of the 53 students who participated in every session of the workshop, 44 students (83%) had completed 80% of the hands-on exercises. The students had a preference for working through the Texas A&M R installation that was tied to their student accounts. Most students accessed RStudio through VOAL or on their own personal machines, although a sizable percentage (approximately 20%) used RStudio Cloud. Preparatory materials were distributed prior to the day of the course showing students how to use these interfaces. Students using these interfaces faced some "expected" technical difficulties during the first section of the course where they were first introduced to accessing RStudio and running the 'learnr' document. These were mitigated by creating breakout rooms in Zoom where students could get one-on-one help from one of the participating instructors. Most of the initial issues were students having difficulty gaining access to VOAL as well as some difficulties installing RStudio and required packages on personal machines. Such technical issues were common in the formal class taught in the Economics program. As such, in anticipation of initial delays, the first section of the course was designed to be very brief, and the rest of the course ran smoothly once students had managed to run the 'learnr' document.

3.1 Description of the Course Content by Section

3.1.1 Introduction

Students were introduced to the RStudio IDE, gaining familiarity with navigating RStudio, how to create new R scripts, how to install packages, and how to view the file system and environment. Students learned how to launch the interactive Shiny document and were introduced to the learnr format. Students were instructed on mathematical operators in R, and exercises in this section included using R as a calculator for simple equations as well as how to call some higher-level math functions (e.g., atan(), sqrt()).

3.1.2 Data Types

In this section, students were introduced to four common data types in R: logical, numeric, integer, and character. The functions str() and class() were introduced in guided exercises, and students used the two functions to check the data types of several elements from a list that contained numerical and character data. The is.logical(), is.numeric(), is.integer(), and is.character() functions were demonstrated, and students completed an independent exercise that required the use of each. The students also conducted independent exercises to evaluate the result of conducting mathematical operations on differing data types. Students were introduced to conditional operators in R and completed independent exercises using the operators to compare numeric, integer, and logical data. Finally, the students were shown the functions as.integer(), as.numeric(), as.logical(), and as.character() to learn how to convert data types, and they completed independent exercises to reinforce the material.

3.1.3 Variables

Students were introduced to variables in R and learned how to assign values to variables. The print() function was introduced in a guided exercise, and students used this function along with class() and str() to inspect variables. Independent exercises in this section provided code with errors that they were asked to fix. The students also learned how to modify the values of variables through mathematical operations and how to assign the value from one variable to another.

3.1.4 Built-in Functions

In this section, students were introduced to the concept of built-in functions in R. The functions sqrt(), atan(), and round() were used as examples. The use of '?' and '??' to get help with functions was covered, and the students used this information to get help with the function head() with which they were asked to examine the built-in iris dataset.

3.1.5 Vectors

Students were introduced to the vector data structure in R and were shown again the different data types that can comprise a vector, reinforcing information that was covered previously in the course. Guided exercises were used to teach the students how to create vectors in r using the ':' operator and seq() function. The students then completed an independent exercise in which they were asked to create and inspect a new vector. We next instructed the students on creating vectors using the c() function and then asked the students to complete an independent exercise that involved creating two new vectors with this method. We then introduced the concept of vector arithmetic with a guided exercise and asked the students to complete additional independent exercises in which they learned how R handles vector arithmetic when a longer vector is not a multiple of a shorter one. Students were next taught how to name the elements of a vector using the names() function in a guided exercise. We next covered how to select elements of a vector both by slicing and by using the names of the element. The students were then asked to complete an independent exercise in which they created a new vector by combining subsets of two provided vectors. Lastly, the students were instructed on conditional and pattern selection of vectors through several guided exercises and were asked to use both of those in an independent exercise to subset a vector of real estate values based on price and name.

3.1.6 Loops

Students were taught the basic concept of loops and their structure in R. 'For loops' were covered first through several guided exercises by either printing or conducting a simple arithmetic operation on a vector of numbers. The students were then asked to complete an independent exercise wherein they first created a 'for loop' to print only odd numbers from 1 through 21 and then printed the total number of odd numbers their 'for loop' identified. We next asked the students to complete an independent exercise in which an amount of interest accrued within a year with different starting values. We then instructed the students on how to iterate over a vector of characters with a guided exercise and then asked that they complete an independent exercise in which they calculate the property tax for each property within a vector and print out the calculated tax and property name. We next covered the concept of nested 'for loops' in a guided exercise and had the students complete an independent exercise in which they created a nested 'for loop'. Students were then introduced to 'while loops' with a guided exercise and were then asked to complete two independent exercises on the topic.

3.1.7 Matrices

In this section, students were introduced to the matrix data structure in R and were taught how to create, modify, and subset matrices. The students first learned to create a matrix with the function matrix() and by combining vectors with the c() function in a guided exercise. We then covered naming rows and columns using rownames() and colnames(), and to add additional rows and columns with cbind() and rbind(). Students were asked to practice this skill by adding a column to an existing matrix in an independent exercise. Next the students learned how to subset a matrix by supplying indices and were asked to practice this technique in an independent exercise. The students were then introduced to the built-in functions sum(), rowSums(), and colSums(), and were asked to practice using these functions in an independent exercise using a provided matrix. In the last portion of this section the students learned how R conducts matrix multiplication. The students were then asked to complete a final independent exercise in which they had to create two new matrices from data provided in a table and then conduct a series of calculations using each matrix.

3.1.8 Data Frames

Data frames in R are in the format of a table or two-dimensional array. They differ from matrices in that they can store heterogeneous data, including numeric, factor, character, and logical types. The data frame is a fundamental data structure in R and is used by many (if not most) R packages to store required data.

The students will likely work with data sets that have a variety of data types in the future. Thus, we started with teaching them some R built-in data frames with different data types and how to do initial assessment with head(), tail() and str() functions. Then, the students learned how to create their own data frames from a matrix with the as.data.frame() function and by combining vectors with the data.frame() function. They were also taught how to change the sequence of the columns. Next, rbind() and cbind() functions were introduced just like we did previously with matrix; these allow one to append a row or column to a data frame. The students also learned to subset and filter portions of a matrix with different methods and how to order a data frame based on the values of one of the columns with the order() function. Nowadays, most data lives in data files. Therefore, we taught students how to read the data in a file to a data frame at the end.

3.1.9 Data Visualization

One of the reasons R has gained so many users is that it is a powerful platform for generating high quality graphs and figures from analyzed data. The base plotting system in R, however, is somewhat limited. We briefly covered the base functions for plotting in R, and the students learned how to generate a scatter plot and add titles and axis labels in a guided exercise using data from the built-in mpg dataset. This was reinforced by an independent exercise in which the students generated plots using different data from the 'mpg' data frame. Given the somewhat limited nature of the R base plotting functions, we chose to teach plotting with the popular R package 'ggplot2'. The students learned the basic structure of the code for plotting with 'ggplot2' in a guided exercise, wherein we generated another scatter plot using data from the 'mpg' data frame. We then used these results to discuss how 'ggplot2' plots in layers and to discuss the concept of aesthetics in 'ggplot2'. We then asked the students to complete an independent exercise that entailed using 'ggplot2' to create another scatter plot on their own. Following this, the students learned how to change point shapes, fill, size, adding plot titles and axis labels, and adding aesthetics mapped to different variables in a series of guided and independent exercises. We next covered how to add additional layers to a plot and finally how to add point labels using geom_text() and geom_label() through guided exercises. We finished this section by teaching the students how to generate histograms using 'ggplot2'.

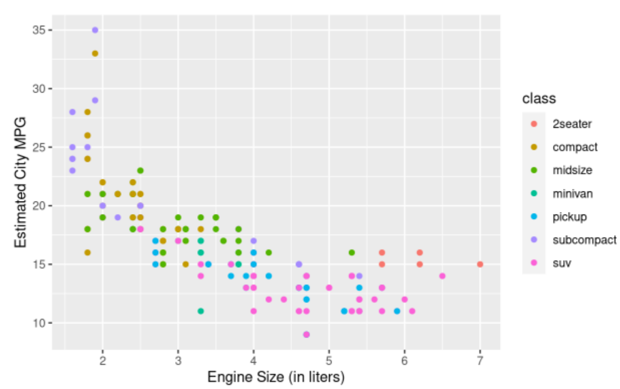

Figure 1. Scatter plot generated in an independent exercise during the Data Visualization section of the R programming for Economics Majors course.

3.2 Feedback from Students

A survey conducted following the end of the course revealed predominantly positive feedback. A vast majority of students (94.3%) felt that the course covered a sufficient range of topics, and most students (87.4%) found that the exercises were at the appropriate level of difficulty. A majority of the students reported that they were engaged for each topic, although many reported needing additional time to complete the exercises as the course progressed through more difficult material (Figure 2). Over half

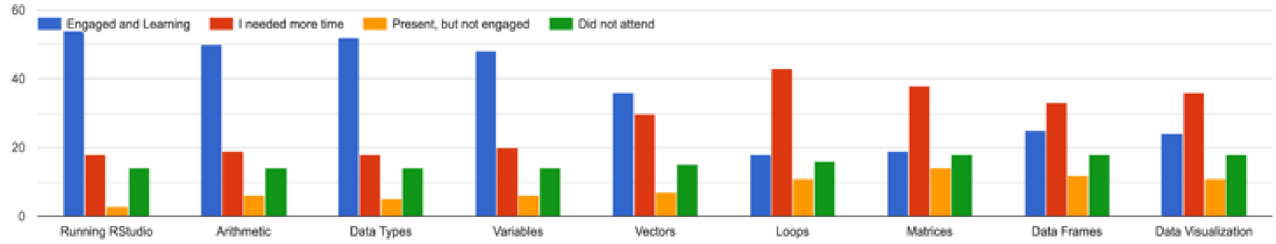

Figure 2. Student self-reported levels of engagement of each topic covered throughout the informal short course.

(51.5%) of the students feel they would have benefited from an additional day of training.

We taught the students different methods of running RStudio (as shown in Figure 3), which gave them more flexibility to choose a favorable platform for the formal economic courses. Most students used either the VOAL option for running RStudio or ran RStudio on their own personal machine. The VOAL option was beneficial, as RStudio was already installed, and installing the packages necessary to run the learnr document was relatively easy for the students, as most of the dependencies for this package were already installed on the VOAL machines. However, interactive feedback during the course as well as survey results revealed that interacting with RStudio through VOAL was frustratingly slow. Given that most students use their personal machines for running RStudio during formal economics courses, it may prove beneficial to direct students to use this approach in future offerings of this workshop. This might be achieved by having a session before the workshop during which students can receive help with installing RStudio and the packages necessary for the workshop.

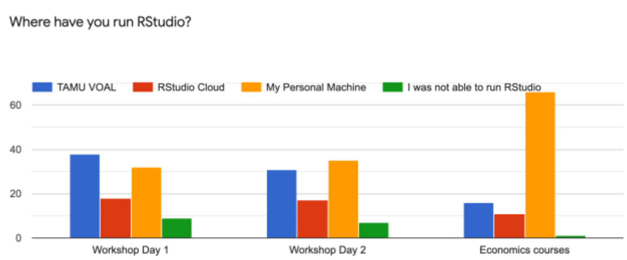

Figure 3. Student responses to where they ran RStudio during the course and where they run RStudio during formal courses.

The survey following completion of the formal economics course offered valuable insight into changes that should be addressed in future iterations of this workshop. Particularly, one open-ended question asked, "In retrospect, what would you change in the pre-course two-day R workshop for it to be more useful in terms of your performance in this course?" Several key points were common in the students' responses. First, many complained about technical difficulties that were encountered during the workshop. These included the issues previously discussed with the connection to VOAL and problems with installation on personal machines. This being one of the most frequent comments following a semester of formal coursework highlights the importance in addressing it in future offerings of this course. Another frequent response on the survey regarded the length of the workshop. Many students commented that staying engaged in a virtual workshop for 12 hours spread across two days was very difficult, and many felt that they would have benefitted from additional contact hours. This problem can easily be solved given the modular nature of the design of the course. The course can easily be adjusted to be taught for fewer hours per day across more days. Also, moving the course from a virtual setting to an in-person one will further improve engagement. Lastly, another common request from the students was to include more instruction on economics-related functions and code. This course was designed to give the students a strong foundation in R prior to a formal course in which R was utilized. In fact, many students commented that they were entering with no knowledge of R. In order to address the needs of students with little or no knowledge of R, the existing modules for the course should be maintained (and perhaps expanded). However, if the course was expanded into more contact hours, it might be worth adding additional modules that include R packages the students might use during the formal course. This might be limited, however, by the need for discipline-specific knowledge for utilizing these packages. For example, some students requested additional training on coding for economic forecasting, which necessitates the need for knowledge on this topic that is outside the scope of our instructors.

4. CONCLUSION

The work is in continuation of our previous effort to build a continuum between cybertraining and domain-specific courses. Working with the faculty of record, we developed a course during which students were instructed in using the R programming language assuming little or no prior knowledge or experience. The topics covered were tailored for these students based on discussions between members of the Texas A&M High Performance Research Computing group and the Department of Economics. The collaborative efforts of our two groups ensured a strong foundational knowledge of R that could be expanded in following discipline-specific graduate level courses. The course was designed using the 'learnr' package in R, which generated a highly portable and easily scalable framework that can be easily adopted and modified for any user group. Students completing the hands-on projects were more likely to complete the course. Feedback from students suggests that this course may best be implemented as several in-person sessions spread across several days. The modular nature of the course makes this implementation easy to achieve and also lends itself to augmentation with additional modules that might cover discipline-specific R packages. In light of the user feedback and growing demand for R training nationwide, the course is being developed as an asynchronous learning resource. Users completing the asynchronous course will be offered the opportunity to earn a micro-credential from TEES-Edge (affiliated with Texas A&M University). The HPRC training program continues to pursue models that leverage campus cyberinfrastructure to positively impact academic curricula.

5. SUPPORTING INFORMATION

The training materials used in this study are available to the community via the Texas A&M HPRC website [13]. The slides used to introduce topics and the R code to render the learnr Shiny interactive document are included in the material. Please send us feedback on your experience adopting this material in an email to help@hprc.tamu.edu.

6. ACKNOWLEDGEMENTS

This work was supported by the Department of Economics at Texas A&M University; the "Accelerating Microcredentials Program" at the Texas Higher Education Coordinating Board; 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; and NSF award number 2019136, "CC* BRICCs: Building Research Innovation at Community Colleges." We thank Dr. Gan Li, Dr. Lisa Perez, and Ridham Patiyola for useful discussions.

7. REFERENCES

[1] Garrick Aden-Buie, Barret Schloerke, and JJ Allaire. 2022. Interactive Tutorials for R • learnr. Retrieved from https://rstudio.github.io/learnr/

[2] Stephen Cass. 2019. The Top Programming Languages of 2019 - IEEE Spectrum. (Sep. 2019). Retrieved from https://spectrum.ieee.org/the-top-programming-languages-2019

[3] 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. *Journal of Computational Science Education* 11, 1 (Jan. 2020), 12–20. DOI: https://doi.org/10.22369/issn.2153-4136/11/1/3

[4] 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. *Journal of Computational Science Education* 10, 1 (Jan. 2019), 40–47. DOI: https://doi.org/10.22369/issn.2153-4136/10/1/7

[5] Dhruva K. Chakravorty, Marinus "Maikel" Pennings, Honggao Liu, Zengyu "Sheldon" Wei, Dylan M. Rodriguez, Levi T. Jordan, Donald "Rick" McMullen, Noushin Ghaffari, Shaina D. Le, Derek Rodriquez, Crystal Buchanan, and Nathan Gober. 2019. Evaluating Active Learning Approaches for Teaching Intermediate Programming at an Early Undergraduate Level. *Journal of Computational Science Education* 10, 1, (Jan. 2019), 61–66. DOI: https://doi.org/10.22369/issn.2153-4136/10/1/10

[6] 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. *Journal of Computational Science Education* 12, 2 (Feb. 2021), 11–17. DOI: https://doi.org/10.22369/issn.2153-4136/12/2/2

[7] 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. *Journal of Computational Science Education* 11, 1 (Jan. 2020), 93–99. DOI: https://doi.org/10.22369/issn.2153-4136/11/1/15

[8] Zhenhua He, Jian Tao, Lisa M. Perez, and Dhruva K. Chakravorty. 2022. Technology Laboratories: Facilitating Instruction for Cyberinfrastructure Infused Data Sciences. *Journal of Computational Science Education* 13, 1 (Apr. 2022), 44–49. DOI: https://doi.org/10.22369/issn.2153-4136/13/1/8

[9] Richard Lawrence, Zhenhua He, Wesley Brashear, Ridham Patoliya, Honggao Liu, and Dhruva K. Chakravorty. 2022. Tailored Computing Instruction for Economics Majors. *Journal of Computational Science Education* 13, 1 (Apr. 2022), 32–37. DOI: 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. *Journal of Computational Science Education* 13, 1 (Apr. 2022), 50–54. DOI: https://doi.org/10.22369/issn.2153-4136/13/1/9

[11] Catherine L. Muller and Chris Kidd. 2014. Debugging geographers: teaching programming to non-computer scientists. *Journal of Geography in Higher Education* 38, 2 (Apr. 2014), 175–192, DOI: https://doi.org/10.1080/03098265.2014.908275

[12] Mahwish Shahid, Afifa Wajid, Kamran Ul Haq, Imran Saleem, and Abdul Haseeb Shujja. 2019. A Review of Gamification for Learning Programming Fundamental. In *2019 International Conference on Innovative Computing (ICIC)*, November 1–2, 2019, Lahore, Pakistan, IEEE, 1–8, DOI: https://doi.org/10.1109/ICIC48496.2019.8966685

[13] Texas A&M High Performance Research Computing. R for Economics Graduate Students. Retrieved from https://hprc.tamu.edu/events/workshops/2022-01-11-REcon.html

HPC Workforce Development of Undergraduates Outside the R1

Scott Feister
California State University Channel Islands
Camarillo, CA
scott.feister@csuci.edu

Elizabeth Blackwood
California State University Channel Islands
Camarillo, CA
elizabeth.blackwood@csuci.edu

ABSTRACT

Many Research-1 (R1) universities create investments in High Performance Computing (HPC) centers to facilitate grant-funded computing projects, leading to student training and outreach on campus. However, creating an HPC workforce pipeline for undergraduates at non-research-intensive universities requires creative, zero-cost education and exposure to HPC. We describe our approach to providing HPC education and opportunities for students at California State University Channel Islands, a four-year university / Hispanic-Serving Institution (HSI) with a primarily first-generation-to-college student population. We describe how we educate our university population in HPC without a dedicated HPC training budget. We achieve this by (1) integrating HPC topics and projects into non-HPC coursework, (2) organizing a campus-wide data analysis and visualization student competition with corporate sponsorship, (3) fielding undergraduate teams in an external, equity-focused supercomputing competition, (4) welcoming undergraduates into faculty HPC research, and (5) integrating research data management principles and practices into coursework. The net effect of this multifaceted approach is that our graduates are equipped with core competencies in HPC and are excited about entering HPC careers.

KEYWORDS

Undergraduate, Workforce, Education, HPC, HSI, Classroom, Student, Scientific computing, Data management

1 INTRODUCTION

When the authors began as new faculty members at California State University Channel Islands (CSU Channel Islands), each had a breadth of experience at multiple universities in bringing High-Performance Computing (HPC) education materials and training opportunities to undergraduates. However, their experience was entirely with Research-1 (R1) universities, and the transition to bringing HPC education to a campus without dedicated HPC on-site training resources was disorienting. In talking with peers who have made a similar transitions from being data researchers at R1 universities into faculty members at non-R1 institutions, our disorientation was not unique. This paper reflects the learning we have undergone in bringing HPC training opportunities to our students. Our goal is to share these experiences in order to help others undertaking similar efforts.

https://doi.org/10.22369/issn.2153-4136/13/2/2

The authors were accustomed to facilitating outreach and exposure for undergraduates by leveraging dedicated university centers. See Table 1 for a few examples of on-site research HPC resources associated with R1 universities, which also provide training support to their undergraduates. University-affiliated HPC resources such as these are created with the resources and research focus of R1 universities.

Undergraduate students at non-R1, more-teaching-focused universities can equally become valuable members of the HPC workforce. However, in the experience of the authors, opportunities for careers in HPC may not be known by some undergraduates at universities where HPC resources are not directly affiliated with the university. This lack of exposure to HPC education may be a barrier to some students who would otherwise pursue it as a career.

In this paper, we provide our approach to bringing undergraduate students into HPC at a university without dedicated, research-program-affiliated HPC resources. Our reflections will be valuable for other science, technology, engineering, and math (STEM) educators working in teaching-intensive universities who are also seeking to expose students to HPC and to support valuable learning experiences in this field.

2 APPROACH

2.1 Integrate HPC into Non-HPC Coursework

We have integrated HPC-relevant activities into everyday coursework for sophomore-level and junior-level undergraduate courses in Mechatronics Engineering and Computer Science. The first course in which HPC concepts are integrated is EMEC 231: Dynamics. This introductory Dynamics course (the physics of motion) is taught with tie-ins to Python numeric programming. Students use numeric programming tools in Python, including the NumPy library, to simulate the motion of a quadcopter in response to forces and torques along multiple axes. In the process, students are given a hands-on introduction to the idea of time-stepping through a simulation and at each step applying known physics equations in order to achieve a numeric solution to a physics problem. This foundation in scientific computing, specifically when it comes to a basic understanding of time-stepping solutions, translates well to many physics HPC applications running on today's supercomputing clusters.

We also have integrated HPC-relevant programming activities into COMP 262: Computer Organization and Architecture and COMP 362: Operating Systems. In both of these courses, students learn to program shared-memory parallel software using OpenMP. In COMP 262, the focus of these exercises is in scientific computing. For example, students in COMP 262 are instructed to parallelize a simple C program that performs a numeric integral to N threads and then create a scaling document to map the performance as a function of threads. Demonstrations of scaling are essential to

Table 1. Examples of On-Campus HPC Resources at R1 Universities. (Deliberately non-comprehensive. We are specifically highlighting a couple R1 institutions in which the co-authors were affiliated prior to joining CSU Channel Islands as faculty and a couple others for context.)

R1 University	Center Providing On-Campus HPC Education
University of Chicago	Research Computing Center
University of Pittsburgh	Office of Research Computing
University of California Los Angeles	Institute for Digital Research & Education
University of Michigan	Advanced Research Computing
Arizona State University	Research Computing

developing software for HPC applications, and this lab gives them their first experience with creating such graphs. We encourage critical thinking about the impact on a future career through a reflection essay on this activity with questions such as:

- Pick one of your C codes and explicitly walk through how your approach to migrating the code from serial to parallel followed each of these three steps: (1) identify parallelism, (2) express parallelism, and (3) express data locality.
- Explain how this lab might affect your own future choices of where to invest your time while developing parallel software in your career.
- From the perspective of a software developer, what did you learn about the difficulties of taking full advantage of the computing power within parallelized CPU architectures?

A second lab day in COMP 262 focuses students' attention on hardware acceleration and General Purpose Graphics Processing Units (GPGPUs). GPGPUs are an essential component of many cutting-edge national HPC resources, and fully utilizing the GPGPUs for scientific computing involves non-trivial programming. We give students their first exposure to NVIDIA CUDA programming through an exercise in which they offload the physics computation of an exploding bag of particles onto a GPGPU. Thousands of independent particles are modeled with kinematic variables stored in large arrays, and this exercise is an analog to physics simulations performed on supercomputers. In the process, students learn about the challenges of massive parallelization of existing computer codes as well as manage the challenges of memory transfer of large arrays of particles between processing units.

2.2 Plot-a-Thon: Campus-Wide Analysis and Visualization Competition

Visualizing large quantitative datasets is a major component of an HPC workflow. The authors co-founded a campus competition in quantitative data visualization that we call the Plot-a-Thon. We have partnered with several local businesses to provide funding for this event on our campus.

The Plot-a-Thon is an overnight hack-a-thon which engages about a hundred students at CSU Channel Islands from a variety of majors. We provide a large quantitative open dataset for students to analyze, drawn from an open online data repository. On the day of the competition, students in teams of four do their best to create compelling stories from the data. The teams are often interdisciplinary, bringing together students from computer science, business, and the humanities to work together and learn from one another. They learn to clean and analyze datasets using tools at various skill levels of expertise: Excel, Tableau, R, and Python. For the Python teams, we teach them quantitative data manipulation using Python packages that are relevant to HPC: NumPy, Pandas, and Matplotlib. Students who are new to the principles and tools have the option to attend faculty-led workshops where they learn using the competition dataset.

Most importantly, students are taught how to tell a story from their data rather than simply speaking about the numbers. The teams must submit a short video describing their process and the story that they intend to tell. This "soft skill" of communication about quantitative data analysis is a valuable one for a career in HPC and its adjacent STEM fields. The event also features a variety of guest speakers on topics of careers in data science from both local and national companies as well as networking opportunities for students who are seeking internships and jobs. The Plot-a-Thon has not only fostered students' skills with real-world HPC concepts but also created meaningful relationships between faculty and local businesses seeking to hire graduates.

2.3 2022 Winter Classic Invitational Student Cluster Competition

Providing students exposure to national-scale HPC resources and professional HPC workflows, as well as helping them develop a relevant professional networking in HPC contexts, will ultimately carry them into the HPC workforce after graduation. Our university was invited to a supercomputing competition targeted at HSI institutions called the 2022 Winter Classic Invitational Student Cluster Competition. According to the competition website, the competition in 2022 was comprised of "twelve teams of students from Historically Black Colleges and Universities (HBCUs) and Hispanic Serving Institutions (HSIs)." Ten of our undergraduate students competed, forming two of the twelve teams.

In the competition, our students learned about HPC workflows and were given access to HPC training crash-courses led by experts. Then, in a series of several weeks, they were given access to supercomputing facilities throughout the country to participate in bi-weekly competition events with other university teams. Every other week, students competed to compile and optimize computer codes for HPC applications. For example, in one event, the teams optimized a simulation of motorcycle air drag using the computational fluid dynamics code OpenFOAM. The winning team was able to coarsen the simulation mesh sufficiently to provide speedup

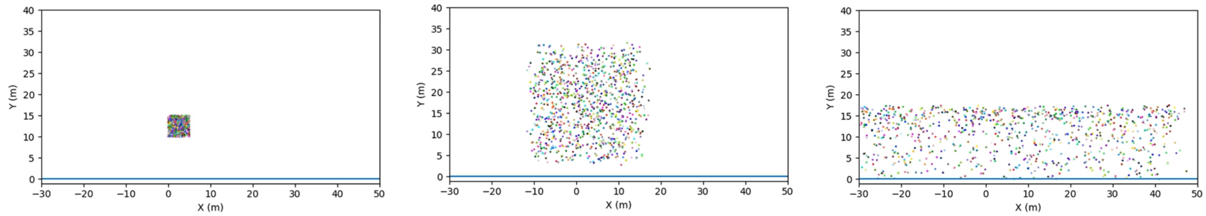

Figure 1. Three time frames from an assignment created by the author (Feister) simulated in C and visualized in Python: "An Exploding Brick of Bouncy Particles." In COMP 262, students offload the core physics time-stepping computation of this simulation onto a GPGPU.

without crossing a threshold in fidelity of the simulation. In addition to learning through doing, students at CSU Channel Islands learned from their competitors afterwards and analyzed each event's outcomes.

The Winter Classic Competition also provided students valuable networking with HPC experts. For example, they set up credentials on NASA computers and Oak Ridge National Laboratory computers. Several CSU Channel Islands students have reported that they have been contacted since the competition with opportunities in HPC and are now considering joining the HPC workforce. This was a high-impact event for our students and we are thankful to have been invited to participate.

2.4 Student Involvement in Faculty HPC Research Projects

Students at CSU Channel Islands can work with faculty members who have ties to the HPC community to develop their skills in undergraduate research projects. These one-on-one mentorship opportunities give students hands-on experience with national supercomputer resources and give them experience compiling and solving problems with advanced scientific codes. One student project worked in a collaboration between Computer Science and Physics to model the trajectory of particles through a proton spectrometer using the Geant4 software. Geant4 is the same software used in medical radiation applications and for high energy physics applications on HPC clusters. Another student project involved a collaboration with faculty in Environmental Sciences modeling sand erosion at beaches along the California coastline. Yet another involved a collaboration with Mathematics to model the cracks that form under rusty conditions in metals. The common link between these various student research projects is that each involved access to faculty member allocations at national-scale supercomputing resources. These projects are the first time any of these students have worked with an HPC resource allocation. Furthermore, each of the codes discussed are massively parallel Message-Passing Interface (MPI) codes written in C or Fortran.

2.5 Integrate Research Data Management Curriculum into Capstone Coursework

We have integrated Research Data Management Curriculum into required capstone coursework for a variety of HPC-adjacent projects. Completion of a capstone project is required for all students at CSU Channel Islands and is performed over one or two semesters during a student's senior year. These projects are mentored by a faculty member and are often based within a subset of the faculty member's research. Many of the aforementioned faculty HPC research projects become students' capstone work. At the beginning of the project, we require students to partake in a Research Data Management Workshop, led by a data librarian, which includes an introduction to the terminology, best practices, and implications of data management.

To solidify these learning objectives from the workshop, students are required to complete a data management plan using the DMPTool (an open online application) for their capstone project at the beginning of the term. Additionally, students are required to submit a final "repository" at the end of the project that includes the data that they plan to archive, relevant documentation, and a "readme" summary. These graded assignments build vital data management skills that students will need for HPC-related fields, especially those that rely heavily on grant funding.

3 ASSESSMENT

One way to assess the impact of these interventions is by examining these in the framework of skills and concepts learned by students that would be needed to enter the HPC workforce. Several were touched on and reinforced for our participating undergraduates through the various interventions.

We created a list of student learning outcomes related to HPC that were met by at least one of these interventions. Then, we compared the relative effectiveness at meeting these outcomes across our various interventions.

The student learning outcomes we assessed in Table 2 are:

(1) Professional Networking in HPC
(2) Remote SSH Access / File Transfer
(3) Multithreaded Programming
(4) GPU Programming
(5) Differentiating HPC Clusters
(6) Compiler Optimizations
(7) Implementing Math/Physics Equations
(8) Domain Decomposition
(9) Troubleshooting with HPC Staff
(10) Managing Compute Resource Allocations
(11) Teamwork in HPC
(12) Data Visualization

(13) Communication of Computational Analysis
(14) Data Management
(15) Writing Technical Documentation

Table 2. Various interventions had different effectiveness in meeting student learning outcomes relevant to HPC workforce development. PT = Plot-a-Thon, FR = Faculty Research, CI1 = Classroom Integration 1 (COMP262), CI2 = Classroom Integration 2 (EMEC 231), WC = 2022 Winter Classic Competition, DMW = Data Management Workshops.

	PT	FR	CI1	CI2	WC	DMW
SLO1		x			x	
SLO2		x	x		x	
SLO3		x	x		x	
SLO4			x			
SLO5			x		x	
SLO6					x	
SLO7		x	x	x		
SLO8		x			x	
SLO9		x			x	
SLO10		x	x		x	
SLO11	x	x			x	
SLO12	x	x	x	x	x	
SLO13	x	x	x		x	
SLO14	x	x			x	x
SLO15		x				x

4 REPRODUCIBILITY

The details of each course integration, campus event, student competition, and faculty project may not be identically reproduced. However, we argue that the overall structure can be reproduced at other universities. The following can be reproduced by faculty at non-R1 institutions using our work here, and the work of others, as example foundations:

- Assigning undergraduates to create computer simulations in non-HPC introductory science courses
- Assigning scientific computing projects, and GPU programming, in non-HPC computer science courses
- Creating campus events promoting communication and visualization of quantitative data
- Seeking out opportunities to engage students in external training and professional networking opportunities like the Winter Classic competition
- Partnering with librarians or other subject experts to build HPC-related training and skills into existing curricula

5 CONCLUSIONS

The approach described in this paper offers a model for exposing students in non-R1 universities to valuable HPC learning experiences. Despite a lack of on-site research-oriented HPC resources and infrastructure, we got students engaged with HPC and equipped them to pursue HPC careers after graduation. Some of the key successes of this effort were student understandings of the challenges and benefits of massively parallel computer programming, students' first access to national research computing facilities, and engagement of students with HPC professionals. The sustainability of such a model is key. We believe that by integrating HPC-focused assignments into existing courses and by organizing extracurricular HPC activities, universities without traditional dedicated computing centers can and will offer valuable HPC learning experiences to their students. The authors plan to further integrate HPC into non-HPC courses, advertise HPC student activities, provide professional networking for students, and seek external collaborations with the national HPC education community in the future to continue to implement these experiences. We also plan to create student feedback surveys on the effectiveness of these interventions and to begin to conduct meaningful assessments that can help us align our practices with the needs of our unique student body. With creativity, willingness, and persistence, educators at universities of all kinds can create a variety of HPC workforce development opportunities for their undergraduate students.

ACKNOWLEDGMENTS

The authors would like to thank Dan Olds for creating and inviting our university to the amazing HPC competition from which our students benefited: the 2022 Winter Classic Invitational Student Cluster Competition. The authors also thank Dr. Megan Kenny Feister for engineering education perspectives and writing support.

Impact of the Blue Waters Fellowship Program

Steven I. Gordon
The Ohio State University
Columbus, Ohio
gordon.1@osu.edu

Scott Lathrop
University of Illinois
Urbana, IL
lathrop@illinois.edu

William Kramer
University of Illinois
Urbana, IL
wtkramer@illinois.edu

ABSTRACT
The Blue Waters Fellowship program supported by the National Science Foundation focused on supporting PhD candidates requiring access to high performance computing resources to advance their computational and data-enabled research. The program was designed to strengthen the workforce engaged in computational research. As the program developed, a number of modifications were made to improve the experience of the fellows and promote their success. We review the program, its evolution, and the impacts it had on the participants. We then discuss how the lessons learned from those efforts can be applied to future educational efforts.

Keywords
Computational science education

1. INTRODUCTION
The Blue Waters Graduate Fellowship program was launched in 2013 and continued through 2021 to provide graduate students across the nation with financial support, access to leading-edge petascale computing resources, and technical and scientific support to accelerate their computational and data analytics research. The program was supported with funding for the Blue Waters project [4] by the National Science Foundation (NSF) [13]. The Blue Waters Fellowship program was modeled upon the NSF graduate fellowship program [16] but with an emphasis on supporting PhD candidates needing access to high performance computing (HPC) resources to support their computational and data-enabled research.

The fellowship was designed to educate scientists and engineers in the use of computational modeling and data analytics applied to critical problems across multiple disciplines. The need for researchers and scholars skilled in the use of computational methods has been recognized in a number of national studies as well as in the ongoing Department of Energy Computational Science Graduate Fellowship program [20, 15].

The fellowship provided funding of up to $50,000 for one year to provide tuition and a stipend for each fellow. Each fellow received an allocation of 50,000 node hours (1.6 million core hours) on the Blue Waters supercomputer in support of sustained petascale computations. As the fellowship program matured, it became clear that a number of changes and additions to the original concept needed to be made in order to ensure the success of the fellows and their research projects.

The fellowship program has continued since 2021 and is now named the New Frontiers Fellowship Program [19], with a focus on national security research and funding from NSF. The program continues to evolve and improve based on our experiences.

In this paper, we describe the evolution of the program to provide the support the fellows needed. We then summarize the impacts of the fellowship program on the participating students and discuss the implications of the program for future efforts that aim to improve the pipeline of researchers that utilize high performance computing to advance their research.

2. GETTING STARTED
With the start of each new cadre of fellows, the fellows were invited to the National Center for Supercomputing Applications (NCSA) at the University of Illinois [6] to initiate their year-long fellowship. Their advisors were also invited to attend. This introductory meeting was beneficial for establishing a good working relationship among the fellows and the staff. The staff provided an overview of the NCSA organization, expectations of the fellows, and a technical overview of the computing environment. The fellows provided a summary of their research goals, computational plans, training needs, and anticipated challenges using the Blue Waters system.

The fellows were invited to the annual Blue Waters Symposium at the start of their fellowship to provide posters describing their proposed research and to meet the fellows from the previous cadre and undergraduate interns from the Blue Waters Student Internship program. Towards the end of their fellowship, the fellows were invited back to the next annual Symposium to present their research findings along with the other Blue Waters Investigators. Each fellow was also encouraged to submit a presentation to a conference within their own research domain, and, if needed, travel funding was provided in addition to their fellowship.

Prior to arriving at NCSA, the fellows completed a short survey about their HPC knowledge so that the staff could tailor start-up training on the use of the Blue Waters system. A similar approach was conducted at the annual Blue Waters Symposium in which fellows and other Blue Waters researchers were able to attend short workshops tailored to teaching participants about using the system effectively.

3. TECHNICAL SUPPORT
To make effective use of the computational resources, the fellows needed some training as well as technical and scientific support. The Blue Waters project assigned a primary point of contact (POC) to each fellow, similar to the support provided to the major NSF Petascale Computing Resource Allocations (PRAC) teams [14]. The POCs were professional staff with significant expertise using HPC systems and a diverse mix of science and engineering backgrounds. Many of the POCs had master's degrees or PhDs. We

DOI: https://doi.org/10.22369/issn.2153-4136/13/2/3

worked to match the technical and domain knowledge of the POCs with the research conducted by each fellow.

The POCs were tasked with organizing monthly virtual meetings with each fellow to obtain updates on their progress and to address barriers to progress. The POC level of effort varied across the fellows depending on the students' background and goals. The interactions among the fellows and their POCs included getting software tools working, compiling and setting up job scripts, porting and compiling codes on the Blue Waters system, and optimizing their workflow to get the best throughput on the system. As a result of the selection process, the fellows already had a strong computational background and did not require significant introductory training or technical support. When the POC was unable to address a problem or challenge, the POCs were able to refer the issue to other University staff with the relevant expertise on algorithms, tools, and effective strategies for conducting computations on the Blue Waters system. Even when referring the request to other staff, the fellow's POC was responsible for managing the request to satisfactory conclusion.

Several of the fellows cited the support and training as key to their success. "One of my favorite components of the fellowship, and something that sets it apart from others, is the access to training seminars and workshops led by experts on topics across the high performance computing spectrum," said Lucas Ford, 2021–2022 Fellow. As Salme Cook, 2017–2018 Fellow, put it, "Having a designated person to provide mentoring and guidance about the Blue Waters system broke down social barriers that made the science progress faster than it otherwise would have. That to me was the "X" factor for the Blue Waters experience."

On average, the staff reported a low level of effort to support the fellows. Most of the fellows could be supported with about 1–2 hours of effort per month. There were a few cases in which four to 20 hours of assistance were provided in the first couple weeks during the initial port of their codes. Usually that initial support was the majority of staff time that was required.

The rationale behind this model of support is that the POC provides continuity of support and a holistic awareness of the research goals and challenges from the outset. The POCs viewed no question as inappropriate or too simple. As one staff member said, "the advantage isn't that the fellow is attached to someone that has an enormous amount of time to spend on them, it's that they're attached to someone who already knows their project, knows their background, and doesn't have to be brought up to speed if they have a quick question. The advantage to me is the already loaded background in the POC person, not the totality of their available time to spend."

4. EXTENDING THE EFFORTS

After the meeting with the first cadre of fellows, it became apparent that the initial meeting needed to be extended from one day to two days. Adding a second day allowed the fellows to spend more time with their POC and receive more in-depth advice for getting started on the Blue Waters System. With more time, the fellows were able to talk with other professional staff and make contact with University of Illinois researchers within their field of study.

It became apparent that 50,000 node hours were insufficient for many of the projects, and that one year of access to the system and POCs was not adequate for the fellows to bring their research projects to fruition. Fellows could propose to receive additional allocations on the Blue Waters system based on progress reports and ongoing plans. After the one-year fellowship, fellows could request additional allocations provided they were enrolled in an academic program and reporting significant research progress. For many of the fellows, the follow-on years were used to fine tune their codes and make production runs aimed at completing their PhD research.

5. BUILDING COMMUNITY

The fellows considered forming a community of practice to be very important. The fellows appreciated getting to know their fellow colleagues, the Blue Waters staff, and meeting other researchers attending the Blue Waters Symposium and other conferences. The quarterly meetings among all of the fellows and support staff added to the sense of community along with the welcoming attitude of the technical and research staff ready and willing to assist as needed. The fellows valued the ability to learn from one another and mentioned that some of these friendships lasted far beyond the timeframe of their fellowship.

Several fellows cited the community as critical to their success. "The Fellowship was instrumental in my career both because it gave me hands-on experience on a large-scale HPC system […] but also because the associated workshops gave me the courage and self-understanding to begin to perform significantly more independent research," said Rachael Mansbach, 2017–2018 Fellow. Micheline Soley, 2019–2020, Fellow, said, "Blue Waters was more than a supercomputer; it was a community and a family. […] Blue Waters was a transformative experience that I hope others will have in the future."

The annual Blue Waters Symposium [5] brought together principal investigators, researchers, students, and leaders in computational science and engineering. The participants shared challenges, solutions, and successes in large-scale heterogeneous computing. Along with presentations from the Blue Waters science teams, the symposium featured keynotes from innovative thinkers in science and provided opportunities to share and discuss specific topics of interest. The Blue Waters external evaluators highlighted the interdisciplinary nature of the Symposium as a key benefit to participating students, faculty, developers, and principal investigators.

The Symposium exposed the fellows to common challenges and technological solutions that span multiple disciplines. They were able to network with other researchers through formal sessions and informal gatherings. The ability to display their own posters and make presentations on their research gave them greater confidence in their activities as well as suggestions for improving their methods.

6. IMPACT

A total of 55 graduate students were selected through competitive national application processes. The fellows were selected from among 38 academic institutions across 27 states. Eleven of the institutions are in EPSCoR[1] jurisdictions and one institution is a Minority Serving Institution. Twenty-two of the fellows (40%) identified as female and/or a historically excluded race or ethnicity.

[1] EPSCoR — Established Program to Stimulate Competitive Research, https://beta.nsf.gov/funding/initiatives/epscor

We maintained communication with 96% of the fellows through ongoing communications and surveys to learn of their career progress and to capture their advice on how the fellowship program could be improved. We conducted a final survey of all fellows in March 2022. Eighteen of the fellows reported being in a postdoc position, twelve in a professional position, and eight with a faculty appointment. One of the faculty members reported pursuing tenure approval during 2022. One of the fellows in the 2021–2022 cadre accepted a postdoc position to begin upon completion of their fellowship. One fellow reported ending their academic studies before receiving their PhD. There were two fellows for whom we were not able to ascertain their status. The remainder of the fellows indicated they were working to complete their PhD within the next couple years.

The fellowship had major impacts on the pace and subject of the fellows' research. As George Slota, 2014–2015 Fellow, indicated, "The fellowship allowed me to focus on my research, which allowed me to finish my PhD much sooner and with a greater number of publications than expected; this enabled me to find a professor position immediately after graduating." The work of the fellows enabled their advisors to publish more papers and increase their HPC knowledge.

A total of 4,161,078 node-hours[2] were used by the 55 fellows. Twenty-four (48%) of the fellows used more than 50,000 node-hours either via supplemental allocations or by making use of reduced charge rates; twelve (24%) fellows used more than 100,000 node-hours. Fifteen of the fellows made little use of their Blue Waters allocations, primarily due to having access to other computing systems on their campus and national laboratories. For a petascale system like Blue Waters, this is a very small fraction of the system and could easily be scaled up to support many more fellows.

A number of the fellows mentioned pursuing different research goals and careers than they had originally intended. Based on the feedback from the fellows, they have been tremendously successful in leveraging their fellowships to advance their research, education, and careers. It would be a very rewarding experience to be able to continue to observe their endeavors over time, and to learn how they in turn impact their colleagues, students, and mentees. "The BW fellowship provided me with the opportunity to conduct independent research early on in my PhD. This helped prepare me for life as an independent researcher and greatly contributed to my ability to get my current position without doing a postdoc. I am very confident in saying that the BW fellowship was central to my development as an independent researcher and as a tenure-track faculty," said Jon Cameron Calhoun, 2014–2015 Fellow.

7. DISCUSSION

Given the success of the program, we believe there are opportunities to leverage the lessons learned to benefit the success of other graduate students pursuing research using computational science. We discuss the issues and possible approaches in the subsections below.

7.1 Scalability and Sustainability

The level of funding available for fellowship positions varied considerably each year. The program was very competitive, selecting between 4% and 10% of the applicants. It is clear that there are many more graduate students looking for fellowships than there are available positions. Due to the pent-up demand, the thematic focus of a fellowship can be as broad or narrow as needed to address the host institution's mission and goals. In the final years of the Blue Waters program, an emphasis was placed on applications from students pursuing geospatial intelligence research projects, and the selection process remained equally competitive.

The need and the value of supporting more fellowships is well documented [20, 15]. The nation would benefit greatly from supporting more fellowships. The Texas Advanced Computing Center (TACC) Frontera Computational Science Fellowships and the SC George Michael Fellowships support HPC fellowships [18, 2]. However, without significant funding, it is unclear how any single institution can realistically support a large number of fellowships. A rare exception is the Department of Energy Computational Science Graduate Fellowship (DOE CSGF) managed by the Krell Institute which focuses on Department of Energy mission oriented research [12].

7.2 Extending HPC Support

Replicating the Blue Waters Fellowship will be difficult given the level of resources needed. However, it may be possible to use parts of the fellowship model to provide assistance to graduate students pursuing research in science and engineering. The personnel time to support the fellows, as previously described, was mostly a light load on the staff. The staff generally supported one or two fellows at a time, along with supporting many other researchers conducting petascale class computational projects.

We believe this model of support is a viable option for other organizations. One possible option is to experiment with the POC idea through other means. Technical and scientific support is needed to complement access to leading-edge and emerging technologies and tools. Allocations of computing time at HPC and University computing centers could be accompanied by assignment of a POC for the initial year of a project allocation to ramp up the applications and codes to make effective use of the HPC system. This could be done via a special, additional application by the graduate students working on a project to receive this extra start-up assistance. It could be accompanied by either some mandatory training or confirmation of basic HPC and programming competencies to ensure the students are well prepared to initiate their computational research and have knowledge of whom to contact when they encounter barriers or need extra assistance.

For those students who are selected for such a program, the assignment of a POC who becomes familiar with the project could help advance the progress of the research projects at minimal cost to the organization providing the HPC allocations. A side benefit might be that the codes being used would be more likely to be optimized to the host system, thus preserving scarce and expensive computing resources.

Providing ongoing access to leading-edge HPC resources is critical to application driven research and the ability to pursue complexity at scale. The level of computational resources used by the fellows was found to have negligible impact on the large-scale research projects on a system like Blue Waters. However, if there are limited resources on a campus computing system, education allocations are readily available on the NSF funded systems (e.g., ACCESS [1]).

[2] 133,154,496 core hour equivalents

7.3 Continuing to Build the Community

In our surveys of the fellows, there was a strong sentiment that even more community building should be pursued. We have seen the fellows transform from student to faculty/research member to major contributor within their field of science. It is useful to pair/group graduate fellows so they feel like they're part of a team and can support one another and answer each other's questions. The use of social media can bolster the connections among the fellows and their extended community of peers, advisors, POCs, etc. Regular scheduled meetings provide a useful forum for sharing progress, challenges, and potential solutions across diverse projects and further the process of building community.

Through the fellows program, we have seen that peer mentoring is very beneficial to all of the participants. We have seen that graduate students relate very well to undergraduate students as near-peer mentors, and these collaborations enhance their own confidence and leadership skills.

The knowledge and experience gained from a fellowship places the fellows in a unique position to advise faculty and their research teams about new techniques, tools, and methods. Their home institutions can leverage this knowledge to enhance local practices. A number of fellows shared their fellowship experiences with others within their research team, including students, advisors and researchers. This helped to raise the awareness of computational methods and improved the research teams' methods and approaches. Some of the fellows also conducted outreach sessions on their campuses through local presentations and informal discussions. They created communities of practice within their local community. The fellows were more attuned to sharing news of computational resources and opportunities that could benefit their colleagues.

7.4 Student Preparation

The fellows chosen for the Blue Waters fellowships had a strong research proposal and strong computational skills and experience. To ensure that more students are equally competitive, academic institutions need to place a greater emphasis on student preparation. However, the concepts and skills relevant to HPC research are still not a part of many scientific curricula even as there is a growing need for computational expertise in most disciplines to apply HPC within their research. The evolution of course curricula does not keep pace with the exponential evolution of technology. Workforce preparation remains a persistent challenge with each new generation of students and a rapidly evolving technological infrastructure.

There are many other activities academic institutions can pursue to prepare their faculty, staff, and students to improve workforce preparation. This begins with raising the basic awareness of computational thinking, parallel programming, and quantitative reasoning. Many campuses benefit from local Software Carpentry [17] and Data Carpentry [9] training sessions. The federally funded HPC centers offer numerous training sessions, workshops, hackathons, etc. that can be accessed virtually, at little or no cost to the students. NSF ACCESS [1] provides easy access to education allocations and start-up research projects on leading-edge HPC systems.

Course sharing between institutions [10] has been shown to be an effective strategy to bring courses to campuses that could not otherwise be offered. In addition to expanding access to more advanced topics, the faculty who are involved can gain confidence to teach the content on their own. HPC centers and professional societies [3, 11] can provide advice and guidance to assist campuses in bringing needed formal and informal courses to campuses.

Campus activities may also include working to host curriculum development workshops with their faculty to incorporate existing computational and data analytics modules into the mainstream educational system. Campuses can introduce informal computational science and data analytics training events offered by Software Carpentry and HPC centers at minimal cost. Establishing research and education partnerships with other institutions and industry will open new opportunities such as fostering internships and fellowships locally and through other organizations.

Every effort should be made to provide students in all disciplines with access to HPC resources with minimal barriers to access, learning, and achievement. Faculty and staff need to infuse new tools and methods for conducting computational methods. Workshops, internships, fellowships, and mentoring are essential to complementing formal coursework. Collectively, these efforts impact student retention and motivation to continue learning when they address authentic real-world problems and challenges.

Finally, a regular series of seminars and workshops will expand informal student learning opportunities. Upon returning from a conference or other similar event, students should be encouraged to share what they learned with fellow students and advisors to potentially incorporate new tools and methods into local practices.

8. CONCLUSIONS

There are many programs that provide in-depth preparation of talented individuals for advanced careers. However, the demand for such programs far exceeds the available resources and programs to give all students this level of engagement.

Academic institutions need to pursue strategies to prepare more researchers able to effectively use HPC technologies and advanced algorithms to advance discovery and scholarship. There are HPC centers, professional societies, consortia, and other organizations committed to working with institutions [11, 7, 8] to address these needs with practical solutions, while also developing scalable and sustainable approaches that work for very diverse institutions and organizations. Persistent attention is needed to ensure that all students are provided with opportunities for engagement in these programs and activities.

Using some of the lessons learned from the Blue Waters Fellowship program can help to encourage both new formal teaching and learning efforts as well as informal programs that help build a cadre of STEM students pursuing computational science related research projects. The ongoing challenge is to raise the awareness of successful workforce development methods so that all academic institutions, and thus all students, have the potential to advance discovery using current and emerging technologies.

9. ACKNOWLEDGEMENTS

The Blue Waters Graduate Fellowship program is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) the State of Illinois, and as of December, 2019, the National Geospatial-Intelligence Agency. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.

10. REFERENCES

[1] ACCESS. 2022. Home - Access. Retrieved November 2022 from https://access-ci.org/

[2] ACM, Inc. 2022. ACM-IEEE CS George Michael Memorial HPC Fellowships. Retrieved November 2022 from https://awards.acm.org/hpc-fellows

[3] ACM SIGHPC Education Chapter. Retrieved November 2022 from https://sighpceducation.acm.org/

[4] Board of Trustees of the University of Illinois. 2022. Blue Waters User Portal. Retrieved November 2022 from https://bluewaters.ncsa.illinois.edu/

[5] Board of Trustees of the University of Illinois. 2022. Blue Waters User Portal | Blue Waters Symposium. Retrieved November 2022 from https://bluewaters.ncsa.illinois.edu/blue-waters-symposium

[6] Board of Trustees of the University of Illinois. 2021. Home - NCSA. Retrieved November 2022 from https://www.ncsa.illinois.edu/

[7] Campus Champions. 2022. Home | Campus Champions. Retrieved November 2022 from https://campuschampions.cyberinfrastructure.org/

[8] Campus Research Computing Consortium (CaRCC). About – carcc.org. Retrieved November 2022 from https://carcc.org/about/

[9] Data Carpentry. Retrieved November 2022 from https://datacarpentry.org/

[10] Steven I. Gordon, James Demmel, Lizanne Destefano, and Lorna Rivera. 2015. Extending Access to HPC Skills Through a Blended Online Course. In *XSEDE '15: Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*. July 26, 2015, St. Louis, Missouri. Association for Computing Machinery, New York, NY, USA, 1–5. DOI: https://doi.org/10.1145/2792745.2792760

[11] IEEE. 2018. Technical Community on High Performance Computing - IEEE Computer Society Technical Consortium on High Performance Computing. Retrieved November 2022 from https://tc.computer.org/tchpc/

[12] Krell Institute. 2022. Front Page | Krell Institute. Retrieved November 2022 from https://www.krellinst.org/

[13] NSF - National Science Foundation. Retrieved November 2022 from https://www.nsf.gov/

[14] The National Science Foundation. Petascale Computing Resource Allocations (PRAC) (nsf17542) | NSF - National Science Foundation. Retrieved November 2022 from https://www.nsf.gov/pubs/2017/nsf17542/nsf17542.htm

[15] National Science Foundation Blue Ribbon Panel on Simulation-Based Engineering Science. 2006. Simulation – Based Engineering Science: Revolutionizing Engineering Science through Simulation. (May 2006). Retrieved November 2022 from https://www.nsf.gov/pubs/reports/sbes_final_report.pdf

[16] NSF Graduate Research Fellowships Program (GRFP). Retrieved November 2022 from https://www.nsfgrfp.org/

[17] Software Carpentry. Retrieved November 2022 from https://software-carpentry.org/

[18] Texas Advanced Computing Center, The University of Texas at Austin, Office of the Vice President for Research. 2020. Frontera Computational Science Fellowships. Retreived November 2022 from https://frontera-portal.tacc.utexas.edu/fellowship/

[19] University of Illinois Board of Trustees. 2022. New Frontiers Fellowships – New Frontiers Initiative. Retrieved November 2022 from https://newfrontiers.illinois.edu/opportunities/new-frontiers-fellowships-2023-2024/

[20] John Wells, Holly Bozeman, Brad Keller, Atsushi Miyaoka, Vasiliy Sergueev, Adrienne von Glatz, and Kerri Wills. 2021. U.S. Department of Energy: Computational Science Graduate Fellowship 1991–2021: A follow-up study of recipients and programmatic outcomes. (Dec. 2021). Retrieved November 2022 from https://www.krellinst.org/csgf/study

The Multi-Tier Assistance, Training, and Computational Help (MATCH) Project, a Track 2 NSF ACCESS Initiative

Shelley L. Knuth
University of Colorado Boulder
Boulder, CO
shelley.knuth@colorado.edu

Julie Ma
Massachusetts Green High Performance Computing Center (MGHPCC)
Holyoke, MA
jma@mghpcc.org

Joel C. Adams
University of Kentucky
Lexington, KY
joel.adams@uky.edu

Alan Chalker
Ohio Supercomputer Center
Columbus, OH
alanc@osc.edu

Ewa Deelman
University of Southern California
Marina del Rey, CA
deelman@isi.edu

Layla Freeborn
University of Colorado Boulder
Boulder, CO
Layla.Freeborn@colorado.edu

Vikram Gazula
University of Kentucky
Lexington, KY
gazula@uky.edu

John Goodhue
Massachusetts Green High Performance Computing Center (MGHPCC)
Holyoke, MA
jtgoodhue@mghpcc.org

James Griffioen
University of Kentucky
Lexington, KY
griff@netlab.uky.edu

David Hudak
Ohio Supercomputer Center
Columbus, OH
dhudak@osc.edu

Andrew Pasquale
Massachusetts Green High Performance Computing Center (MGHPCC)
Holyoke, MA
andrew@elytra.net

Dylan Perkins
University of Colorado Boulder
Boulder, CO
dylan.perkins@colorado.edu

Alana Romanella
University of Colorado Boulder
Boulder, CO
alana.romanella@colorado.edu

Mats Rynge
University of Southern California
Marina del Rey, CA
rynge@isi.edu

ABSTRACT

NSF-supported cyberinfrastructure (CI) has been highly successful in advancing science and engineering over the last few decades. During that time, there have been significant changes in the size and composition of the participating community, the architecture and capacity of compute, storage, and networking platforms, and the methods by which researchers and CI professionals communicate. These changes require rethinking the role of research support services and how they are delivered. To address these changes and support an expanding community, MATCH is implementing a model for research support services in ACCESS that comprises three major themes: 1) leverage modern information delivery systems and simplify user interfaces to provide cost-effective, scalable support to a broader community of researchers, 2) engage experts from the community to develop training materials and instructions that can dramatically reduce the learning curve, and 3) employ a matchmaking service that will maintain a database of specialist mentors and student mentees that can be matched with projects to provide the domain-specific expertise needed to leverage ACCESS resources. A new ACCESS Support Portal (ASP) will serve as the single front door for researchers to obtain guided support and assistance. The ASP will leverage emerging, curated tag taxonomies to identify and match inquiries with knowledge base content and expertise. Expert-monitored question and answer platforms will be created to ensure researcher questions are accurately answered and addressed in a timely fashion, and easy-to-use interfaces such as Open OnDemand and Pegasus will be enhanced to simplify CI use and provide context-aware directed help. The result will be a multi-level support infrastructure capable of scaling to serve a

https://doi.org/10.22369/issn.2153-4136/13/2/4

growing research community with increasingly specialized support needs, resulting in research discoveries previously hindered by researchers' inability to effectively utilize NSF CI resources. This paper will cover the components of the MATCH project and discuss how MATCH will engage and work with the ACCESS community.

KEYWORDS

ACCESS, Research support, Knowledge base, Science gateways, Scientific workflows

1 INTRODUCTION

As science and engineering researchers continue to push boundaries in their respective fields, consumption of cyberinfrastructure (CI) in the form of compute, storage, networking, and software becomes increasingly important. Ensuring the availability of state-of-the-art CI to continue these successful research endeavors, the National Science Foundation (NSF) announced a new vision for access to NSF funded CI through the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) project. The goal of this project is to provide democratized utilization of CI, allowing for equitable use of these powerful systems. This vision builds from the success of the eXtreme Science and Engineering Discovery Environment (XSEDE) [6], which provided a robust ecosystem of resources and support to enable research discovery. The ACCESS project as outlined consists of five tracks focused on allocations, end user support services, operations and integration services, monitoring and measurement services, and technology translation services, tied together through an ACCESS Coordination Office.

Available hardware to enable research, while important, is only one component of a successful CI ecosystem. Without adequate support to understand how to use CI efficiently and appropriately, researchers can face challenges in successful research discovery. The Multi-Tier Assistance, Training, and Computational Help (MATCH) project is designed to support the end users of these systems, providing equitable, scalable support to best enable research on NSF funded CI. The goal of the MATCH project is to decrease the time to successful solution for computational research problems, enabling researchers to focus less on complex workflows and more on discovery, while also providing confidence in their results.

2 TIERED SUPPORT MODEL

To accomplish this goal, the MATCH project focuses on a tiered support model (Figure 1). The objective of this model is to successfully decrease the time to solution while still maintaining high quality service. The base of the pyramid represents maximum usage of CI by users while providing minimal human interaction, while the apex of the pyramid represents a small number of users who likely have special cases requiring more consultation. By providing a mixture of support solutions tailored to different researchers' needs, the MATCH project can help researchers resolve problems quickly, efficiently, and successfully. It is important to note that in practice, researchers will not actually step through the tiers; rather, recognizing that researchers vary significantly in their need for assistance and their preferences for obtaining it, the MATCH project has designed a variety of resources, outlined below, to help them accomplish their goals.

2.1 Tier 1 — Easy-to-use Tools and Services

The Tier 1 support model, shown at the bottom of the pyramid in Figure 1, leverages existing tools in the high performance and high-throughput computing space that better enable research through ease of use. The highly successful Open On-Demand (OOD) tool [3], developed at The Ohio Supercomputer Center, supports computational researchers by providing an interface that allows for the efficient use of remote computing resources. The OOD interface removes the underlying complexity of CI by providing a web interface to enable efficient file management, command line shell access, and job management and monitoring across a variety of servers and resource managers. Templates can be utilized to submit jobs to a scheduler, transfer data, and monitor services.

The MATCH project will integrate the OOD tool with the Pegasus workflow tool [1]. The Pegasus project simplifies complex data workflows by helping applications execute in a variety of environments. An application can be easily mapped onto available compute resources and execute a multi-step process in an appropriate order, managing workflows with millions of computational tasks. The Pegasus tool also eases complex workflow management for end users through reproducibility, ensuring data integrity, and enabling provenance.

The OOD and Pegasus tools will be enhanced, improved, and integrated to further reduce the complexity of using CI systems. Beginning in September 2022, three pilot projects involving three different institutions will inform the future of this effort.

The Pegasus and OOD tools will also be integrated with the ACCESS Support Portal (ASP), so that a user can login to one place to use all of these tools. The ASP has underpinnings in the Connect.CI portal [4]. This tool provides a window into the ACCESS service, allowing users to monitor and apply for allocations and support, conduct account management activities, submit tickets, and join affinity groups. By providing a seamless, singular tool to utilize a variety of support and account information, researchers are able to quickly work toward their goals.

2.2 Tier 2 — Self-help Knowledge Base

The goal for Tier 2 MATCH services is to assist researchers with quickly finding answers to their computational problems by providing a single location to source information. In the MATCH knowledge base, a researcher may search for an answer to their question and be provided with a number of solutions from a variety of MATCH approved information sources. These sources may include documentation or training materials written by ACCESS staff members; information resourced from existing HPC centers across the globe; documentation, code templates, tutorials, or other materials provided by the MATCH community grants program (described below); or answers to questions that have previously been posed in a community forum (https://ask.cyberinfrastructure.org/) [5]. The MATCH program is also developing and will work with members of a *Computational Science Support Network (CSSN)* to identify gaps in computational resources intended to support researchers and to create or resource new or existing materials to fulfill that gap. The MATCH effort will provide financial assistance through a community grants program to encourage members of the CSSN to assist with filling these gaps in resources. The ACCESS knowledge base

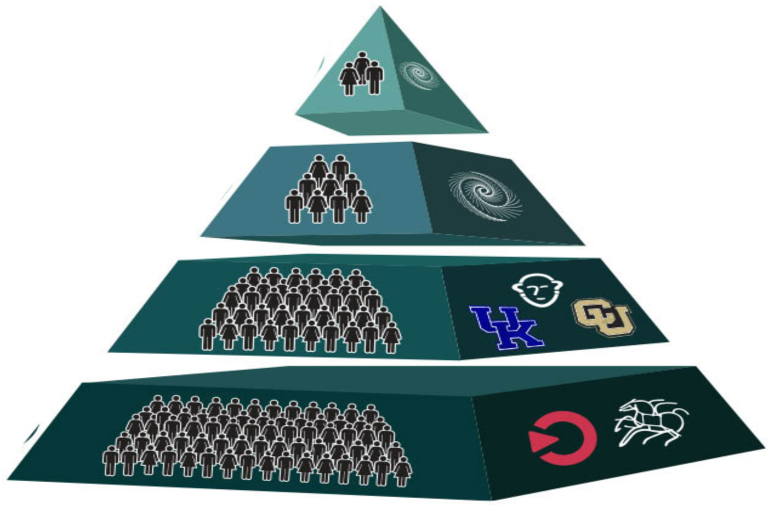

Figure 1. Tiered Support Model in MATCH

can by found here: https://support.access-ci.org/knowledge-base. Membership in the CSSN is as simple as filling out a short form at https://support.access-ci.org/cssn.

The incorporation and development of new and existing affinity groups will play a strong role in this effort. MATCH affinity groups are formed with community members who have a common, shared interest in a computational issue, scientific or engineering research endeavor, diversity and inclusion effort, or any other connection point that will appropriately serve the ACCESS community. The MATCH team will provide each affinity group with access to specific Slack channels, question and answer forums, news and outage alerts, events and training materials, and other relevant knowledge base resources. The affinity groups structure will play a significant role for all of the ACCESS teams to connect with a variety of community members, including researchers as well as those needing allocations support or ACCESS Resource Providers. Assisting science and engineering research fields in community building efforts is anticipated to be a manifestation of this effort.

2.3 Tier 3 — MATCH Plus

The Tier 3 MATCH Plus engagement supports researchers in need of short term assistance to resolve a computational barrier that is slowing the time to solution. Often times, these computational barriers, such as transitioning to a new resource or optimizing code, may be resolved with short term assistance from a student. Researchers will be able to request a MATCH Plus engagement where a short form will ask for a description of the problem and any tags to help identify students with relevant skill sets who can assist. Students will also be matched with a mentor who will provide guidance to the student for resolution of the issue. The MATCH Plus engagements are intended to be short term commitments (on the order of six months or less). Modeled after the successful Northeast Cyberteam [2] project, organized by several members of the MATCH team, MATCH will facilitate regular touch points between the students and researchers, students and mentors, and among the students, ensuring completion of the project. This model is intended to provide extra support to quickly facilitate completion of computational research projects as well as provide an experiential learning experience for students across a variety of disciplines. Students, researchers, and mentors who join the CSSN will be able to participate.

2.4 Tier 4 — MATCH Premier

Similar to MATCH Plus, MATCH Premier provides support to researchers needing short term engagement to resolve a computational barrier encountered when conducting a research project. The primary difference between a Tier 3 and Tier 4 engagement is that the Tier 4 engagement is suitable for computational issues that may be more complex and require the assistance of a staff consultant rather than a student. Similar to the XSEDE Extended Collaborative Support Services (ECSS) program, consultants are available to facilitate the removal of these barriers. However, the MATCH program will only provide the connection points to consultants who are members of the CSSN rather than the full suite of services the ECSS program provided. MATCH is working with researchers to identify consultants that can be funded through proposals being written outside of the ACCESS project. The MATCH team will provide suitable collaborators for the research projects via the same tag taxonomy that is part of the MATCH Plus engagement and, similarly to Tier 3, will facilitate regular connection points between the team members.

3 CONCLUSIONS

The goal of the MATCH effort is to assist the science and engineering research community with the appropriate computational resources, tools, and support to reduce the time to solution for research and allow researchers to focus on their primary interests — solving research questions, securing future grants, attracting quality faculty and students to their institutions, publishing papers, and graduating students. The MATCH approach relies on simplifying what can be a complex process for researchers who use large-scale computational resources as only one component of their research ecosystem. The approach described relies on best practices from existing support models but also moves the community into the next phase of supporting cyberinfrastructure. One of MATCH's primary goals is to provide all users of ACCESS resources with equitable and scalable support. This tiered model approach can create connection points across the community — from those utilizing more traditional workflows to those who do not — that will best support and enable cutting edge research nationally. The MATCH team looks forward to evolving and growing with the needs of the community throughout the length of this project.

ACKNOWLEDGMENTS

This project is supported by the National Science Foundation under OAC-2138286. The MATCH Team would like to acknowledge the work of the ACCESS projects supporting allocations, operations, measurement and monitoring, and the ACCESS Coordination Office, with whom this project is in close collaboration.

REFERENCES

[1] Ewa Deelman, Karan Vahi, Gideon Juve, Mats Rynge, Scott Callaghan, Philip J Maechling, Rajiv Mayani, Weiwei Chen, Rafael Ferreira Da Silva, Miron Livny, et al. 2015. Pegasus, a workflow management system for science automation. *Future Generation Computer Systems* 46 (2015), 17–35. https://doi.org/10.1016/j.future.2014.10.008

[2] John Goodhue, Julie Ma, Adrian Del Maestro, Sia Najafi, Bruce Segee, Scott Valcourt, and Ralph Zottola. 2020. Northeast Cyberteam: Workforce Development for Research Computing at Small and Mid-Sized Institutions. In *Practice and Experience in Advanced Research Computing* (Portland, OR, USA) *(PEARC '20)*. Association for Computing Machinery, New York, NY, USA, 402–406. https://doi.org/10.1145/3311790.3396662

[3] Dave Hudak, Doug Johnson, Alan Chalker, Jeremy Nicklas, Eric Franz, Trey Dockendorf, and Brian L. McMichael. 2018. Open OnDemand: A web-based client portal for HPC centers. *The Journal of Open Source Software* 3, 27 (2018), 622. https://doi.org/10.21105/joss.00622

[4] Julie Ma, Sarah Akbar, Torey Battelle, Kevin Brandt, Eric Brown, Dana Brunson, Dhruva Chakravorty, Thomas Cheatham, Bhushan Chitre, Adrian Del Del Maestro, Andrea Elledge, Vikram Gazula, John Goodhue, James Griffioen, Abigail Hyde, Douglas Jennewein, Shelley Knuth, Bj Lougee, Timothy Middelkoop, Sasmita Mohapatra, Sia Najafi, Kaylea Nelson, Lisa Perez, Bruce Segee, Julia Sheats, Andrew Sherman, Christopher Simmons, Ermal Toto, Scott Valcourt, Lucas Varella, and Ralph Zottola. 2021. The Connect.Cyberinfrastructure Portal. In *Practice and Experience in Advanced Research Computing* (Boston, MA, USA) *(PEARC '21)*. Association for Computing Machinery, New York, NY, USA, Article 51, 4 pages. https://doi.org/10.1145/3437359.3465606

[5] Julie Ma, Torey Battelle, Katia Bulekova, Aaron Culich, John Goodhue, Jacob Pessin, Vanessa Sochat, Dana Brunson, Tom Cheatham, Sia Najafi, Chris Hill, Adrian Del Maestro, Bruce Segee, Ralph Zottola, Scott Valcourt, Zoe Braiterman, Raminder Singh, Robert Thoelen, and Jack Smith. 2021. Ask.Cyberinfrastructure.org: Creating a Platform for Self-Service Learning and Collaboration in the Rapidly Changing Environment of Research Computing. *The Journal of Computational Science Education* 12 (Feb. 2021), 37–40. Issue 2. https://doi.org/10.22369/issn.2153-4136/12/2/9

[6] John Towns, Timothy Cockerill, Maytal Dahan, Ian Foster, Kelly Gaither, Andrew Grimshaw, Victor Hazlewood, Scott Lathrop, Dave Lifka, Gregory D. Peterson, Ralph Roskies, J. Ray Scott, and Nancy Wilkins-Diehr. 2014. XSEDE: Accelerating Scientific Discovery. *Computing in Science and Engineering* 16, 5 (Sept. - Oct 2014), 62–74. https://doi.org/10.1109/MCSE.2014.80

Impact of Blue Waters Education and Training

William Kramer
University of Illinois
Urbana, IL
wtkramer@illinois.edu

Scott Lathrop
University of Illinois
Urbana, IL
lathrop@illinois.edu

Steven I. Gordon
The Ohio State University
Columbus, Ohio
sgordon20@gmail.com

Robert M. Panoff
Shodor Education Foundation, Inc.
Durham, NC
rpanoff@shodor.org

Aaron Weeden
Shodor Education Foundation, Inc.
Durham, NC
aweeden@shodor.org

Lizanne DeStefano
Georgia Institute of Technology
Atlanta, GA
ldestefano6@gatech.edu

ABSTRACT
The Blue Waters proposal to NSF, entitled "Leadership-Class Scientific and Engineering Computing: Breaking Through the Limits," identified education and training as essential components for the computational and data analysis research and education communities. The Blue Waters project began in 2007, the petascale computing system began operations on March 28, 2013, and the system served the community longer than originally planned as it was decommissioned in January 2022. This paper contributes to the Blue Waters project's commitment to document the lessons learned and longitudinal impact of its activities.

The Blue Waters project pursued a broad range of workforce development activities to recruit, engage, and support a diverse mix of students, educators, researchers, and developers across the U.S. The focus was on preparing the current and future workforce to contribute to advancing scholarship and discovery using computational and data analytics resources and services. Formative and summative evaluations were conducted to improve the activities and track the impact.

Many of the lessons learned have been implemented by the National Center for Supercomputing Applications (NCSA) and the New Frontiers Initiative (NFI) at the University of Illinois, and by other organizations. We are committed to sharing our experiences with other organizations that are working to reproduce, scale up, and/or sustain activities to prepare the computational and data analysis workforce.

Keywords
HPC, Computational science and engineering, Data analytics, Education, Training, Higher education, STEM workforce

1. INTRODUCTION
The Blue Waters proposal to NSF, entitled "Leadership-Class Scientific and Engineering Computing: Breaking Through the Limits," [1] identified education, training, and community engagement as essential components for the computational research and data analysis communities. Blue Waters emphasized working with organizations and individuals where they live and work and proactively recruited, engaged, and retained a diverse community of partners (i.e., similar to what others often refer to as "users") in utilizing the full range of Blue Waters resources and services as well as the expertise of the University of Illinois and our collaborators.

We present information on how training, education, student research experiences, and community engagement activities were organized and how these activities evolved based on external evaluations, community feedback, and attention to the needs and requirements of the target audiences. We also describe the partnerships, cyberinfrastructure, and evaluations that enhanced, supported, and improved the activities. The information presented here complements and extends previous reports we have given [9, 7].

Throughout the University of Illinois's 35+ years of providing national-scale HPC resources and services, there has been a persistent turnover of people using the resources and services (over 50% of whom are graduate students), increased utilization of HPC systems among disciplines that have not traditionally used them (e.g., humanities, arts, and social sciences), and the continued rapid evolution of hardware and software available to accelerate discovery. The critical need for HPC expertise in academia, industry, and government is growing faster than the workforce is able to sustain.

There is a persistent need to develop and deliver new training content while continuing to keep existing materials current. Higher education institutions need to identify strategies to maintain state-of-the-art courses to improve the preparation of students to contribute to and advance computational and data analytics research and scholarship. The education, training, student research experiences, and community engagement activities described herein provide foundational elements that have good potential to be scaled up and sustained.

2. EDUCATION
The education programs were developed to prepare the next generation of computational researchers, educators, and

DOI: https://doi.org/10.22369/issn.2153-4136/13/2/5

[1] "Leadership-Class Scientific and Engineering Computing: Breaking Through the Limits" supported by the National Science Foundation awards OCI-0725070 and ACI-1238993, the State of Illinois, and as of December, 2019, the National Geospatial-Intelligence Agency.

practitioners attending higher education institutions. Blue Waters provided allocations on the Blue Waters system for programs across the U.S. needing access to petascale class resources.

The majority of the activities focused on the creation and/or delivery of formal credit courses, curricular materials, and exercises. The materials that were developed continue to be used in hands-on workshops for faculty and students to demonstrate how computational thinking and computational science can be incorporated into the undergraduate curriculum and enhance STEM education. The materials are disseminated through multiple repositories such as HPC University [19] and the Computational Science Education Reference Desk [18]. The ability for instructors to adopt and/or adapt well tested and vetted materials provides a good foundation upon which to learn and build. The community provides feedback to improve the materials, and they in turn contribute materials tailored to their own teaching and learning needs for dissemination through the repositories.

2.1 Curriculum Development

Blue Waters launched its education initiative even before the Blue Waters computing system was in operation to help faculty begin to think about how to prepare their students for working in a highly parallel high-performance computing environment. The Shodor Education Foundation, Inc., led the effort to engage undergraduate faculty in the development of 30 undergraduate course modules for use in undergraduate classrooms. The materials were designed to prepare students for utilizing a petascale class system to run interactive models and simulations to learn STEM principles. Once the Blue Waters system was deployed, these modules were updated by the Shodor team and collaborating faculty to incorporate the petascale capabilities of the system.

The Shodor team utilized these materials during curriculum development workshops that were conducted from 2009–2011 with over 700 undergraduate faculty across the U.S. By using computational models that scale to hundreds (if not thousands) of processors, the team was able to demonstrate how faculty can teach the complexities of developing parallel codes that cannot be realized using systems with four to eight processors. The workshops supported faculty in adopting and/or adapting these materials into their own courses. In return, the faculty provided feedback for improving the materials, and many contributed their materials to the repositories. Shodor also conducted workshops that were tailored to high school and middle school teachers incorporating computational thinking into their classrooms using many of these same materials.

During 2020, the Shodor team led a group of 22 faculty to develop additional shorter modules in applied HPC modeling and simulation by updating and adapting the previous 30 modules as well as materials previously developed for the Blue Waters Intern Petascale Institute described below in the Student Research Experiences section. Approximately 60 small modules were created. The curricular materials are available [17].

2.2 Virtual School of Computational Science and Engineering

The Blue Waters team prototyped the delivery of shared graduate-level credit courses among a diverse mix of institutions. The project focused on delivering HPC-related topics that were not able to be provided at collaborating academic institutions. Six graduate credit courses for the Virtual School of Computational Science and Engineering were conducted from 2013 to 2016. These courses were modeled upon the distributed courses offered by the Big Ten Academic Alliance Online Course Sharing Program. Computational science experts taught HPC topics to graduate students at 31 collaborating academic institutions. Three of the institutions were Minority Serving Institutions, and 12 were in EPSCoR [14] jurisdictions. Faculty at the collaborating institutions were the official class instructors, allowing a total of 633 students to register and earn graduate credit at their own institutions. This approach made it possible for the high-end computational science course topics to be offered at colleges and universities that would otherwise not have been able to offer them.

Flipped classrooms were employed for a combination of synchronous and asynchronous learning. Flipped sessions provided students with materials to study on their own followed by live (in-person and virtual) discussion groups with their local instructors. A number of the faculty indicated that they planned to continue to use the flipped model even when not conducting shared virtual classes.

The sharing of courses among institutions brought changes to the participating institutions. The faculty became more informed about the subject matter and gained confidence subsequently to teach the subject matter on their own. The students earned credit from their own institutions even though the content was developed and delivered by faculty at other institutions.

These efforts demonstrated the potential for scaling up access to advanced topics and the potential to prepare more faculty to teach these advanced topics on their own. To be sustainable, the benefits need to be recognized and supported by multiple faculty, and the courses must become institutionalized by the department chairs with support from their deans. In addition, inter-institutional agreements are needed for cost-sharing among participating institutions. Other organizations continue to explore the scalability and sustainability of these types of initiatives. A report of lessons learned delivering online graduate credit courses is available [5].

2.3 Education Allocations

The Blue Waters education allocations provided access to the computing system, enabling U.S.-based organizations to engage people in a variety of computational science and data enabled learning opportunities. The community proposed innovative teaching and learning activities that would benefit from access to petascale computing resources. The types of sessions that were supported included workshops, institutes, hackathons, undergraduate and graduate courses, research experiences for undergraduates, internships, and graduate fellowships. Formative feedback led to streamlining the process for submitting requests and receiving timely decisions within two weeks. Applications could request up to 25,000 node-hours (a measure of the number of computing elements used times the amount of time used), although a few exceptions were made for large credit courses with over 100 enrolled students.

Education allocations supported a broad range of learning activities, and we encouraged innovative approaches to support learning, with 188 education allocations supporting over 5,000 participants at 160 institutions, including 41 institutions in EPSCoR jurisdictions and 14 Minority Serving Institutions. The education allocations caused minimal interference to the research allocations running alongside them on the computing system. The graduate fellowships were the only categories that requested and received increased allocations based on documented progress reports and ongoing computational research plans.

3. TRAINING

The audience for training activities included undergraduate and graduate students, faculty, researchers, and staff from among academic institutions, industry, business, government agencies, decision makers, and other sectors of society. Graduate students represented the largest fraction (over 50%) of participants. Anecdotally, we learned from high school teachers that some of the training materials were used in their classes. Training activities focused on developing and delivering materials to enable participants to learn about cyberinfrastructure tools, methods, and resources via informal learning events to empower participants to conduct research and scholarship at scale.

In-person training sessions were offered at conferences, institutes, hackathons, workshops, the two-week Blue Waters Intern Petascale Institute, the International HPC Summer School events, the annual Blue Waters Symposium, on multiple campuses, and at other events that brought people together in one location.

While in-person training events are considered to be very effective, the majority of the Blue Waters partners/users were spread across the country, and most of them were unable to travel to in-person training events. The most time- and cost-effective method for serving the community was by providing the majority of the training virtually. Face-to-face virtual sessions can be conducted effectively using high-definition video conferencing infrastructure. The ability for the presenters and participants to see each other and interact in real-time, even though remote from each other, enriches the teaching and learning environment. High-definition video conferencing, interactive communications tools, and expert online advisors support learning at scale at reasonable costs.

While many people asked that sessions be recorded, our experience was that a very small fraction (less than 5%) of participants accessed these recordings. Our experience with the uptake of self-paced training tutorials and webinars demonstrates that it is valuable to the community when the recordings are split into smaller modules, which does incur post-production labor costs.

We evaluated the training events to capture feedback and recommendations from participants for improving the delivery and identifying additional topics to offer. However, there is an ongoing challenge to identify quantitative and qualitative methods for assessing the long-term benefits and a return on investment (ROI) for providing training.

3.1 Virtual School of Computational Science and Engineering

Virtual School of Computational Science and Engineering (VSCSE) summer schools were conducted from 2008 through 2016 in cooperation with the University of Michigan. The infrastructure team enhanced the video conferencing quality to utilize high-definition video conferencing facilities to broadcast the sessions among multiple sites simultaneously. The program delivered training content to an initial set of four locations in the first year and expanded over time with experience and improved infrastructure to deliver content to 24 locations simultaneously. Speakers presented from various sites, with participants able to conduct face-to-face Q&A sessions from their home institutions. The video conferencing training rooms were updated over time with additional cameras and screens to enhance the environment for instructors and participants. We gained confidence in the ability to scale up support to accommodate the management and coordination of more than 20 sites simultaneously.

This approach facilitated access by participants at 146 different academic institutions, including 10 Minority Serving Institutions and 63 institutions in EPSCoR jurisdictions, reaching a total of 5,865 people. Three papers about the VSCSE including lessons learned are available [4, 3, 29].

3.2 Petascale Computing Institutes

Building on the VSCSE experiences, two week-long Petascale Computing Institutes were conducted in 2017 and 2019 and engaged over 600 registered participants at 21 sites, including two international sites, all simultaneously. Individuals were able to participate from their home/office/lab to view the broadcasts and pose questions, although these individuals did not receive accounts on the institute HPC systems since we could not validate their identities. The institutes were possible due to collaborations with content providers from multiple HPC centers and the cooperation of hosts at multiple campuses. These events demonstrated that it is practical and cost-effective to engage a large and diverse community of learners.

Improvements were made to facilitate additional online Q&A sessions, allow individuals to participate from their own offices, spend more time testing exercises in advance, and ensure that participants could use the computing systems before hands-on activities began. More time was also spent to test A/V connections and to allow time for presenters to become familiar with the physical setup in advance of their presentations. The advance testing helped to reduce problems during the presentations and hands-on sessions.

3.3 International HPC Summer School

Blue Waters contributed support to the in-person International HPC Summer School for multiple years by providing access to an education allocation on the Blue Waters system as well as providing technical staff to teach sessions and serve as mentors to the graduate students participating in the program. Each summer school supported 80 graduate students from the U.S., Canada, Europe, and Japan.

Over time, the Summer School provided more depth of technical training in response to the participant feedback and evaluations, brought back students from previous summer schools as peer mentors, provided more in-depth constructive advice on being good mentors, and increased the time for mentoring. Backup technology plans were added to address challenges experienced in previous years.

3.4 Hackathons

Blue Waters hosted hackathons to focus on helping selected research teams of two-to-six members with improving the performance of their codes on the Blue Waters system. An emphasis was placed on helping the participants take advantage of the GPU capabilities of the system. Mentoring assistance was provided by NVIDIA staff and academic professionals and researchers.

It is important to identify mentors with both technical and scientific knowledge appropriate for each team. It is also important that the mentors are able to spend a considerable amount of time with their assigned team. A number of teams benefitted from spending time optimizing their CPU codes and realizing considerable speedup before considering incorporating GPU capabilities.

3.5 Webinars

The Blue Waters team organized a series of nearly 200 webinars, most of which lasted about an hour, although a few sessions were two or three hours in length. All sessions were recorded for access by people who were not able to attend at the scheduled time. As the number of webinars grew, they were categorized to assist people in finding recordings of the sessions of interest. The webinar content spanned technology, research, and workforce development topics. YouTube was used to maximize ease of access to live and recorded sessions and for providing automatic transcriptions.

There were more than 45,000 views of the webinars, including a large number of views of the recordings. This made the time and effort to record the sessions worthwhile. We were advised to allow at least three weeks of advance notice to allow time for people and organizations to promote the webinars. Based on the positive feedback, the webinar series will continue past the Blue Waters funded period with support and leadership from the New Frontiers Initiative.

3.6 Self-Paced Training

For many people, just-in-time training is essential to their learning needs. We have recommended Cyberinfrastructure Tutor [25] and the Cornell Virtual Workshop [1] as two popular sources for self-paced HPC learning resources. Reports from these providers indicate that these tutorials are very heavily used. Evaluations indicate that some viewers are looking for answers to specific questions rather than going through the full tutorial, which makes short, indexed modules quite useful to the community.

3.7 Training Event Planning

The following are lessons we have learned to help make training sessions more effective.

3.7.1 Audience

There is a persistent challenge that no matter how well one describes the prerequisites, content, and learning goals, one will still likely have an audience with diverse background and knowledge. For a session with a large audience, one needs to decide strategically how slow/fast one plans to proceed and how to address an audience with a broad base of novice and advanced skills. Regardless of one's choices, some members may become disengaged, so it is important to make content and target audience clear from the outset. For a session with a small or targeted audience, it can be useful to conduct a short survey of the registered attendees to gain a better sense of their knowledge so that you can better tailor the content based on the audience's knowledge and topics they want to learn.

Everyone should be made to feel welcome to encourage asking questions they may feel are too "simple." Attendees are more comfortable asking questions if they know their background and experience are comparable to those of other participants. Initial ice breakers will make them more comfortable asking questions and talking with fellow participants.

The ability for participants to post questions and receive quick responses throughout each session enhances the learning experience. While this can be accomplished through direct interaction with the instructor, Q&A interactions with content experts through collaborative tools (e.g., Slack) can be equally useful and less disruptive to the instructional flow. These collaborative tools also help to reduce shyness and encourage people to ask questions they may be uncomfortable verbalizing.

3.7.2 Content

All sessions should be clear about the learning goals, content to be covered, and prerequisites. Sessions that encourage participants to bring their own codes/applications to use during laboratory sessions will enhance their ability to practice what they learn with their own code and data during and after the sessions.

No matter how much content you may want participants to learn, it is not possible to cover everything you feel that they "may" need to know. Do not pack sessions tight; leave time for people to absorb the content, practice with exercises, talk with the instructors and each other, and have ample time for breaks. Providing sufficient food helps people relax. For multi-day events, plan to end early on Friday so people can go home and relax; the audience will likely thin out early regardless of the schedule.

3.7.3 Instruction

Good teaching practices can be more effective than having the most technically qualified presenter. The level of content (i.e., introductory, intermediate, or advanced) should be made clear to the instructors in addition to the participants. We strongly encourage instructors to practice what they teach. When teaching students problem solving approaches using interactive tools, the instructors should practice with the same tools and approaches.

3.7.4 Exercises

A process for assisting participants with running exercises should be established in advance. This should include ensuring there is an adequate level of technical support relative to the number of participants. For remote participants, plan for effective mechanisms for "looking over the shoulder" of participants to help diagnose any problems they encounter. Exercises allow the participants to ascertain they are learning the concepts. It is essential that exercises are well tested in advance to work on the platforms that are being utilized by the participants.

Some participants appreciate the ability to earn a "recognized" badge of knowledge gained that they may add to their resume or biography. Some participants appreciate the ability to combine badges and/or earn a certificate of knowledge gained that is verified through some type of exam or quiz (as opposed to a certificate of participation that some people need for their jobs).

3.8 Impact

We can conservatively report that Blue Waters engaged over 45,000 participants from 224 academic institutions in 49 states, Puerto Rico, and 22 countries. There were participants from 20 Minority Serving Institutions and 68 institutions in EPSCoR jurisdictions. There were over 250 education and training events, 188 education allocations, and over 90 course modules. Presentations were made at numerous national and international conferences, institutes, summer schools, and workshops each year for which no records of the number of participants were reported.

Many of the materials, slides, and video recordings from the training sessions described above are available [26] along with the slides and video recordings from the Blue Waters webinar series [27] and a more detailed report of the Blue Waters training experiences [11].

4. STUDENT RESEARCH EXPERIENCES

Student engagement programs were conducted to directly engage undergraduate and graduate students in computational science and data analytics research projects. The goal was to enhance their motivation to pursue advanced studies and careers to advance research and scholarship. The programs included year-long student internships for 139 undergraduate students as well as graduate fellowships for 55 PhD students from across the U.S.

4.1 Graduate Fellowships

The Blue Waters Graduate Fellowship program engaged 55 PhD students across the U.S. from 2013 through 2022. The fellowship program was modeled upon the NSF Graduate Research Fellowship Program but with an emphasis on supporting PhD candidates needing access to petascale class high performance computing resources to support their computational and data-enabled science and engineering research. The fellows received up to $50,000 in financial support, which included a stipend, tuition support, and travel to Blue Waters events. They also received an initial allocation of 50,000 node-hours on the Blue Waters petascale computing system.

4.1.1 Recruitment and Selection

The recruitment process began with a national awareness campaign starting in the fall of each year among people with allocations on Blue Waters and the Extreme Science and Engineering Discovery Environment (XSEDE) [24]; through social media and HPC media companies; through consortia and professional societies; and with assistance from individuals with mailing lists of faculty, researchers, professionals, and historically excluded groups. Each year, approximately 90–100 applications were each reviewed by three faculty and researchers with comparable scientific background. A technical review was also conducted to ensure that the proposed research was appropriate to be conducted on the Blue Waters system. A final review was used to ensure gender diversity, institutional diversity, and geographical diversity among candidates who had been evenly ranked.

4.1.2 Start-up

The fellows attended an inaugural meeting at NCSA in the early fall timeframe to provide an overview of their research and computational plans. We found that a two-day meeting allowed time for the fellows to consult with their points of contact (POCs) and other staff and researchers on campus. To make effective use of the petascale computational resources, the fellows benefitted from some start-up training. The fellows completed a short survey of their HPC experience that helped us to fine-tune the training to be most useful for them.

Each fellow was assigned a primary point of contact; this had already been proven to be an effective mode of support for research teams with allocations on the Blue Waters system. The fellows repeatedly reported that this support model made a significant positive impact on their ability to accelerate their computational research plans.

We encouraged the fellows to invite their advisors to attend the start-up meeting; however, few advisors attended. Recognizing that some of the fellows helped to impact the computational efforts of their advisors and fellow students, we would encourage making participation by advisors a stronger component of the program.

4.1.3 Research Experience

The fellows were able to spend the year focused on their research goals. The POCs engaged in regular (at least monthly) contact with each fellow to minimize barriers to progress that normally occur over time. Quarterly calls were arranged with all the fellows and the support staff. These were useful to share information, identify problems, and propose solutions, many of which tended to benefit multiple fellows. Each fellow submitted a quarterly report and a final report to document publications and presentations and pose questions to the staff, and the fellows were interviewed by the external evaluators.

The fellows were invited to two Blue Waters Symposia to present posters about their proposed research in the first year and to give presentations on their research findings near the end of their fellowships. Through these events, the fellows learned about common challenges from people across multiple disciplines. They indicated that their participation helped them improve their own research practices, presentations, and publication skills and improve their confidence.

While every fellow received an initial allocation of 50,000 node-hours, it became apparent that many of the fellows needed supplemental allocations of time. It was important to provide an easy process for the fellows to request supplemental allocations by providing an update on research progress to date, a justification for the quantity of resources needed, and plans for future research activities.

4.1.4 Post-fellowship

Most of the fellows continued with their computational research beyond the fellowship period while working to complete their PhD. We supported extended access to the Blue Waters system for these fellows and for fellows entering postdoc positions at accredited academic institutions. We did not extend this offer to fellows that went to national laboratories or industry positions, as we felt that those organizations could support any ongoing computational needs.

4.1.5 Impact

A total of 55 graduate students were selected through competitive national application processes. The fellows represented 38 academic institutions across 27 states. Eleven of the institutions are in EPSCoR jurisdictions and one institution is a Minority Serving Institution. Twenty-two of the fellows (40%) identified as female and/or a historically excluded race or ethnicity.

We maintained communication with 96% of the fellows through ongoing communications and surveys to learn of their career progress and to capture their advice on how the fellowship program could be improved. We conducted a final survey of all fellows in March 2022. We were pleased to learn that eighteen of the fellows reported being in a postdoc position, twelve in a professional position, and eight with a faculty appointment. Additional information about the program is available [10].

4.2 Student Internships

The Blue Waters Student Internship Program was launched in 2009 and continued through 2019 to provide undergraduate students across the U.S. with training, a year of financial support, access to leading-edge petascale HPC resources, and the opportunity to work with a mentor on a computational research project. The program was managed by the Shodor Education Foundation, Inc. We previously reported details of the program in 2014 [8].

4.2.1 Recruitment and Selection

We promoted the Student Internship Program and conducted reviews of the applications in a similar manner as the Graduate Fellowship Program. The program proactively recruited individuals from groups historically excluded from HPC. The formative evaluations helped the Shodor team to modify the selection process to introduce anonymous reviews.

Initially, traditional methods of selecting from among the applicants did not yield success engaging adequate diversity among the participants. We adjusted our methods by asking application reviewers to avoid only selecting the applicants who were the "best of the best" and already had significant experience and instead to consider which of the applicants had the greatest potential for learning and would benefit the most from participating but who had not yet had opportunities to gain experience. Our working motto in the selection process was to "promote excellence rather than just rewarding excellence." This change had the profound impact of increasing the percentage of selected students from groups historically excluded from HPC from 13% in the first two years of the program to 43% over the course of the whole program. Additional information about our early efforts is available [6].

4.2.2 *Start-up*

During the first year of the internship program, participants were selected for summer internships. We found the interns frequently had insufficient background or experience to contribute significantly to their research projects to which they were assigned. To address this, we extended the internships to a full year and brought all of the interns together at NCSA for a two-week Petascale Institute at the start of each internship. The institute focused on providing the students with a background in parallel programming, quantitative analysis, and computational science tools and methods.

Subsequent feedback from the mentors indicated the students were much better prepared to quickly engage with and contribute to their research projects. Extending the length of the internships to a year resulted in the students being able to accomplish much more progress than students involved in a traditional summer internship.

The materials developed for the Petascale Institutes are available [23].

4.2.3 *Research Experiences*

Undergraduate student engagement benefited from committed mentors. A limiting factor in being able to support enough students was having enough projects for the students to work on with mentors who cared and could devote time to supporting undergraduate students. Thus, it was important to focus recruiting efforts on reaching out to potential mentors. We encouraged mentors to recruit students from their own institutions in whom they saw potential for being able to contribute to their research. We also encouraged mentors to be open to working with students at other institutions, since many students had interest in working on computational projects for which there was no suitable faculty member at their own institution.

Every intern was strongly encouraged to submit manuscripts sharing their research findings and documenting their learning outcomes to the Journal of Computational Science Education (JOCSE) [21] or another journal in their specific discipline.

A few interns were selected each year to present posters on their research at the annual Blue Waters Symposium. Participation in the Symposia enhanced the interns' research experiences, allowed them to network with graduate students and researchers, and became a capstone event for their year-long effort. A few interns were also supported to attend the SC conference to make connections with mentors and collaborators. These events helped form bonds among the interns and build community.

4.2.4 *Post-internship*

A number of the interns were motivated by their experiences to return as peer mentors and instructors for subsequent Petascale Institutes. Having peer mentors was an advantage to building community and enhancing the experience for the participants.

4.2.5 *Impact*

While there were added costs for extending from a 10-week summer program to a full year program, the benefits of student preparedness and deeper research engagement and progress outweighed the costs. The students were also more likely to be supported by their research mentors past the internship period.

A total of 139 undergraduate students were selected for research internships through national application processes. The students represented 73 academic institutions across 32 states and Puerto Rico. Thirty-seven of the students were from institutions in EPSCoR jurisdictions and 26 of the students were from Minority Serving Institutions. Sixty-one of the interns (43%) identified as female and/or historically excluded races/ethnicities. Thirty-six of the interns published a peer-reviewed paper in the Journal of Computational Science Education (JOCSE), at least 16 others published in other journals, and at least 17 others presented posters at various meetings and conferences. A number of undergraduate and graduate students funded by XSEDE student engagement programs also participated in the two-week Petascale Institute.

Over the years, the formative evaluations allowed the Shodor team to improve the content and student experiences. The topics and materials used in the Institutes were updated each year based on formative and summative evaluation feedback from the external evaluators, comments from the students, and observations from the instructors [31]. Additional information about the internship program is available [13].

The lessons learned from the Blue Waters Student Internship Program were applied to the XSEDE EMPOWER program [30] funded by the XSEDE project and managed by Shodor. The program recognized that providing different levels of internships — from learners, to apprentices to interns — is an effective strategy to provide an on-ramp to computational science for students based on their background and experience.

4.2.6 *Human-interest*

One intern, Aaron Weeden (co-author of this paper), was in the first group of interns in 2009, assisted as a peer mentor and assistant instructor in the first Petascale Institute in 2010, was later a staff mentor for a project involving four interns in 2014, led the curriculum development and instruction of the last five Petascale Institutes, and coordinated the internship program in its final three years. This experience allowed Aaron to grow as a mentor and leader and have a fulfilling experience enabling many other students to have impactful internship experiences.

5. COMMUNITY ENGAGEMENT

Blue Waters was funded to serve the U.S. research and education community with access to petascale resources and services. A significant challenge was disseminating information about the resources and services to the people who would most benefit. Traditional users of previous HPC resources were generally well informed about the resources funded by NSF. Our challenge was to continue to recruit and support traditional audiences while working to raise awareness and engage organizations and individuals who were not well informed.

We placed an emphasis on engaging a diverse community. We refer to the term diversity to include: 1) historically excluded groups including women, historically excluded races and ethnicities, and people with disabilities; 2) individuals from all academic institutions including Minority Serving Institutions, institutions in EPSCoR jurisdictions, PhD granting institutions, primarily undergraduate institutions, two- and four-year institutions; and 3) researchers from among all disciplines.

To recruit, engage, and support a large and diverse community of participants, partners, and collaborators required an extensive and ongoing effort to build a diverse national list of contacts. The contacts included PIs and researchers who had allocations on the Blue Waters and XSEDE systems. Participants in Blue Waters events were added to our mailing list. Blue Waters staff visited many campuses to raise awareness among faculty, staff, administrators, and students.

Students told us that they most often learned about opportunities from their faculty and advisors; thus, we pursued multiple avenues to contact faculty, deans, and department chairs.

Considering the challenge of raising awareness nationally, we included intermediaries who in turn shared our message with individuals and groups they knew may be interested. Among our intermediaries were the Campus Champions, the Coalition for Academic Scientific Computation (CASC), the Campus Research Computing Consortium (CaRCC), media organizations (e.g., HPCwire), and professional societies (e.g., the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE)). Contacts were made with student-oriented organizations (e.g., Women in Engineering, ACM Student Chapters). To engage Minority Serving Institutions, we contacted technology leaders at the Hispanic Association of Colleges and Universities (HACU) that represents Hispanic Serving Institutions, National Association for Equal Opportunity in Higher Education (NAFEO) that represents Historically Black Colleges and Universities, and American Indian Higher Education Consortium (AIHEC) that represents Tribal Colleges and Universities. We used social media to promote the resources and services.

5.1 Building Community

The external evaluators gave special recognition and praise to the interdisciplinary activities of the Blue Waters project. The annual Symposia placed a strong emphasis on bringing together PIs, researchers, professionals, and students. Discussions with participants demonstrated that the participants realized the value of sharing common algorithms and techniques for addressing computational methods that were common across multiple disciplines. The ability of fellows and interns to display their own posters and make presentations on their research gave them greater confidence in their research activities.

The fellows consistently mentioned the Blue Waters effort to build community as very important. The fellows appreciated getting to know their fellow colleagues, the Blue Waters staff, and other researchers attending the Blue Waters Symposium and other conferences. They valued the ability to learn from one another and mentioned that some of these friendships lasted far beyond the timeframe of their fellowship. A common response from students was that more community building should be pursued. This includes greater use of social media and other communication platforms (e.g., Slack).

Activities like the Petascale Institute for the student interns included social time for the students to get to know one another and the support staff while also providing time to relax and unwind. Sharing meals helped to foster conversations and connections.

On a lighter note, many HPC-related songs have been written and sung by Bob Panoff [22]. A parody of a folk song, “The Water is Wide,” performed by Shodor intern Krista Katzenmeyer, is posted on YouTube [16].

5.2 Impact

We were successful in engaging over 45,000 participants from 225 academic institutions, 20 Minority Serving Institutions, 68 institutions in EPSCoR jurisdictions, 41 laboratories and centers, and 22 industries in 49 states, Puerto Rico, and 41 international organizations in 22 countries. Additional presentations were made at numerous national and international conferences, institutes, summer schools, and workshops every year, during which data on participation were not collected.

6. SUPPORT INFRASTRUCTURE

The intellectual efforts described above were enhanced by strategic partnerships, cyberinfrastructure, and evaluations.

6.1 Strategic Partnerships

Developing partnerships and collaborations with national and international organizations was a strategic goal for scaling up and sustaining Blue Waters programs and activities.

The broad range of learning needs and requirements of the HPC community, from introductory materials to advanced skills, exceeds the capabilities of any one institution, organization, or consortium to address. There continue to be new researchers every year — whether undergraduates, graduates, or researchers — who see the potential for HPC resources to accelerate their computational research. In addition, new hardware, tools, and applications continue to emerge for which training is needed to inform and prepare the community for making effective use of the resources.

6.1.1 Project Management Partners

Partnerships were developed with the Shodor Education Foundation, Inc., to 1) develop curricular materials for classroom instruction, 2) work with faculty to incorporate HPC resources, tools, and methods into the curriculum, and 3) engage undergraduate students with internships. The Ohio Supercomputer Center supported the Blue Waters Graduate Fellowship program, enhanced the HPC University portal, and coordinated the Virtual School of Computational Science and Engineering shared graduate credit courses. The Center for Education Integrating Science, Mathematics & Computing (CEISMC) at the Georgia Institute of Technology conducted formative and summative evaluations of the education and training activities.

Blue Waters coordinated with the XSEDE project to conduct complementary education and training activities, avoiding duplication of effort and helping to facilitate quick sharing of lessons learned. Over time, Blue Waters collaborated on multiple training activities with multiple HPC centers funded by NSF, DOE, and DOD as well as with international HPC organizations including Compute Canada, PRACE, and RIKEN.

6.1.2 Collaborative Partners

We have seen the benefits derived from collaborations with educational institutions to co-teach credit courses that could not otherwise be offered at the member institutions. Collaborations with HPC centers and higher education institutions allowed the delivery of HPC training to hundreds of participants via multi-day sessions at minimal cost to the cooperating organizations and institutions. Professional societies (e.g., ACM SIGHPC Education Chapter) are bringing together trainers, educators, and staff to foster ongoing sharing of best practices and lessons learned, collaborate on the organization of education and training workshops at major international HPC conferences, and form working groups with common goals. Professional societies are extending community cooperation beyond the sharing that occurs at annual conferences and allowing member organizations to achieve greater impact.

6.1.3 Human-interest

Of special note, the SIGHPC Education Chapter announced in August 2022 that Dr. Robert Panoff, Shodor Education Foundation, was named as the first recipient of the award for Outstanding Contributions to Computational Science Education, to be honored during the awards ceremony at the SC22 Conference.

6.2 Cyberinfrastructure

The physical infrastructure included the computing resource, the high-definition video conferencing facilities, and the learning repositories. The evaluations also played a key role in helping to identify ways to improve the activities and document the longitudinal impact of the activities.

6.2.1 Reliable and Accessible Cyberinfrastructure

The infrastructure being used to deliver the content needs to be well tested in advance. This includes confirming that all presenters and facilitators have tested and used the A/V, computing infrastructure, and communications channels (e.g., Slack) in advance of the session. A plan for testing and confirming that all participants can access, log in, and submit a simple task to the computing platform in advance of the first use of the system will reduce frustrations and delays during the instructional periods.

To accommodate large numbers of participants running short exercises, arrange special queues for small jobs to get through the HPC system quickly. Advise the system operators of the impending load well in advance to preclude any downtime, system upgrades, or large-scale jobs running at the same time.

We highly recommend that a backup plan should be established to accommodate possible technology failures or interruptions that could potentially arise that would disrupt the flow of an event. Anticipating issues and being ready to mitigate them avoids wasting time trying to fix them on the fly while people "twiddle their thumbs."

6.2.2 Video Conferencing Infrastructure

As a member of NCSA, the Blue Waters project was able to utilize the existing human talent and physical audio/video infrastructure that existed within NCSA. As Blue Waters expanded the use of high-definition video conferencing capabilities, NCSA expanded the physical infrastructure and capabilities to match the usage and needs for full two-way communications.

NCSA's video conferencing rooms utilize multiple cameras to ensure that the instructor and the audience members can be seen along with a view of the instructor's desktop. Microphones are placed in front of every participant with a mute/unmute button within their control. The room monitor can mute all participants, if needed, to minimize background noise. Large screens are used to ensure that all participants can see the views presented to all locations. Each site also has the capacity to see "picture in picture" views of all locations joined into the video conference.

Specifications and setup for the NCSA video conferencing setup are available [28].

6.2.3 Learning Repositories

Many educators and instructors of formal and informal learning sessions do not know how to start to develop training content and incorporate computational methods into the curriculum. This challenge can be addressed by sharing well-tested materials and exercises via public repositories (i.e., portals, collections, or libraries). We have demonstrated significant uptake of modules from such repositories by faculty and staff who have incorporated computational methods into the curriculum.

Many people use search engines (e.g., Google) to find training and education materials they need. The curation of education and training materials enhances the ability for people to find materials that are most relevant to their needs. Curation helps to ensure that materials are high-quality and up-to-date. Curated collections may include verification, validation, and accreditation information [12]. They may also include roadmaps (sequences for learning), validated exercises, translations to multiple languages, and community reviews.

Blue Waters collaborated with Shodor and XSEDE to augment the list of training and education resources accessible through the repositories HPC University [19], the Computational Science Education Reference Desk [18], and Interactivate [20]. Over six million page views of these repositories were recorded in April 2022.

There are hundreds (if not thousands) of repositories in the world, and this exceeds the capacity for any one person or group to compile and categorize an index of materials that are available. The next most useful stage of evolution of repositories for the community is to develop common metadata and implement automatic searching across multiple repositories. The ACM SIGHPC Education Chapter is pursuing this strategy to expand access to education and training materials on an international scale.

6.3 Evaluations

From the outset of the Blue Waters project, it was determined that evaluating community engagement programs was critical to learning how to improve them. Formative evaluations are most important for identifying issues that can be addressed to improve the content, delivery, and overall experiences for participants, instructors, and facilitators.

External evaluators conducted focus groups, surveys, and one-on-one discussions. The evaluators provided anonymized reports to the Blue Waters project office and discussed strategies for addressing the findings with the project managers on a regular basis.

The formative evaluations led to many improvements in the EOT activities. Their impact is reflected in the lessons learned. The summative evaluations helped to document the overall impact through longitudinal analysis of the impact of the programs. The evaluators previously reported on their methods and findings [2].

7. SUMMARY

HPC centers place a high importance on making effective use of expensive shared computing systems since demand for access far outweighs the available resources. Training is a key component in helping to ensure that people know how to make effective use of these systems and how to maximize the potential for accelerating their discoveries. The need for training is a direct result of the lack of formal preparation of today's undergraduate and graduate students for understanding how HPC resources can be applied to advance research and development in all disciplines.

Faculty, staff and students conducting computational science and data analysis come from many backgrounds and discover the need for computational resources at very different times in their educational and professional careers. As such, a successful program needs to incorporate many different approaches.

We have demonstrated methods by which HPC centers can work with educational institutions to provide students with better preparation by working together to co-teach courses, update the curriculum, and ensure a continuum of improvement in courses over time for all institutions to benefit students in all disciplines. We have demonstrated that a comprehensive education, training, and student engagement program can effectively engage and serve a large national audience.

The impact of the lessons learned persists. Many organizations have adopted the use of high-definition video conferencing for events, the Texas Advanced Computing Center (TACC) supports an HPC graduate fellowship program, XSEDE adapted the Blue Waters Student Internship Program for the EMPOWER program [30], XSEDE scaled up their webcast training to serve people at remote institutions, groups like PICUP [15] are offering computational science and engineering workshops for faculty using Shodor's materials, more researchers are incorporating GPUs and other emerging tools, the ACM SIGHPC Education Chapter is building community and making learning repositories more accessible, the Ohio Supercomputing Center is using NSF funding to prototype shared credit courses among Minority Serving Institutions, and there are over five million page views per month of materials in the repositories supported by Shodor.

We are pleased to share our experiences with the community and to see our best practices and lessons learned being applied by other organizations. There are more detailed internal reports on these activities [11, 10, 13]. We are committed to continuing to offer our expertise, insights, resources, and services. We welcome you to contact us to discuss these topics in greater depth.

8. ACKNOWLEDGMENTS

This work is part of the Blue Waters sustained-petascale computing project, "Leadership-Class Scientific and Engineering Computing: Breaking Through the Limits", which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) the State of Illinois, and as of December, 2019, the National Geospatial-Intelligence Agency. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.

9. REFERENCES

[1] Cornell University. 2022. Cornell Virtual Workshop. Retrieved November 2022 from https://cvw.cac.cornell.edu/default

[2] Lizanne DeStefano and Jung Sun Sung. 2019. Initial impact of Evaluation in Blue Waters Community Engagement Program. *Journal of Computational Science Education* 10, 1 (January 2019), 32–39. DOI:https://doi.org/10.22369/issn.2153-4136/10/1/6

[3] Sharon C. Glotzer and Thom H. Dunning. 2008. Great Lakes Consortium for Petascale Computation Virtual School of Computational Science and Engineering: Educating a New Generation of Computational Scientists and Engineers for Discovery and Innovation in the 21st Century. (May 2008). Retrieved November 2022 from https://uofi.box.com/s/frdzwfjotla0k7bln2xsggd2rbskvrru

[4] Sharon C. Glotzer, Bob Panoff, and Scott Lathrop. 2009. Challenges and Opportunities in Preparing Students for Petascale Computational Science and Engineering. *Comput. Sci. Eng.* 11, 5 (Sept.–Oct. 2009), 22–27. DOI:https://doi.org/10.1109/MCSE.2009.134

[5] Steven I. Gordon, James Demmel, Lizanne Destefano, and Lorna Rivera. 2015. Extending Access to HPC Skills Through a Blended Online Course. In *XSEDE '15: Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*. July 26, 2015, St. Louis, Missouri. Association for Computing Machinery, New York, NY, USA, 1–5. DOI: https://doi.org/10.1145/2792745.2792760

[6] Jennifer K. Houchins, Jeffrey D. Krause, Scott A. Lathrop, and Robert M. Panoff. 2011. A multi-cultural success story for achieving diversity in multi-core/many-core internships. In *TG '11: Proceedings of the 2011 TeraGrid Conference: Extreme Digital Discovery*, July 2011. Salt Lake City, UT. ACM. Article 51, 1 page. DOI:https://doi.org/10.1145/2016741.2016796

[7] Jennifer Houchins, Scott Lathrop, Robert Panoff, and Aaron Weeden. 2020. Blue Waters Workforce Development: Delivering National Scale HPC Workforce Development. *Journal of Computational Science Education* 11, 1 (January 2020), 26–28. DOI:https://doi.org/10.22369/issn.2153-4136/11/1/5

[8] Patricia Jacobs, Phillip List, Mobeen Ludin, Aaron Weeden, and Robert M. Panoff. 2014. The Blue Waters Student Internship Program: Promoting Competence and Confidence for Next Generation Researchers in High-Performance Computing. In *2014 Workshop on Education for High Performance Computing*, November 16, 2014, New Orleans, LA. IEEE. 49–55. DOI:https://doi.org/10.1109/EduHPC.2014.6

[9] Scott A. Lathrop, Katharine Cahill, Steven I. Gordon, Jennifer Houchins, Robert M. Panoff, and Aaron Weeden. 2020. Preparing a Computationally Literate Workforce. *Comput. Sci. Eng.* 22, 4 (July–Aug. 2020), 7–16. DOI:https://doi.org/10.1109/MCSE.2020.2994763

[10] Scott Lathrop and Steven I. Gordon. Impact of the Blue Waters Graduate Fellowship Program. Retrieved November 2022 from https://uofi.box.com/s/0z1kirremhfrtw3zm8u8xagvnepnxq5d

[11] Scott Lathrop, Steven I. Gordon, Aaron Weeden, and Lizanne DeStefano. Impact of Blue Waters Education and Training. Retrieved November 2022 from https://uofi.box.com/s/k4gpj704cnv2irt49txue5wahwu5bmqo

[12] Scott Lathrop, Marilyn McClelland, Steven Gordon, D. E. Stevenson, and David Joiner. 2005. Applying verification, validation, and accreditation processes to digital libraries. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '05)*, June 7–11, 2005, Denver, Colorado. IEEE. 382–382, https://doi.org/10.1145/1065385.1065485

[13] Scott Lathrop, Robert M. Panoff, and Aaron Weeden. Impact of the Blue Waters Student Internship Program. Retrieved November 2022 from https://uofi.box.com/s/i65thkjwo80rop95yjavdhm002i75q8k

[14] National Science Foundation. Established Program to Stimulate Competitive Research (EPSCoR). Retrieved November 2022 from https://beta.nsf.gov/funding/initiatives/epscor

[15] PICUP. 2022. PICUP: Partnership for Integration of Computation into Undergraduate Physics. Retrieved November 2022 from https://www.compadre.org/PICUP/

[16] Shodor. 2022. Blue Waters is Wide. Retrieved November 2022 from https://www.youtube.com/watch?v=Dor2_B_OkR8

[17] Shodor. 2022. Blue Waters Petascale Education Program. Retrieved November 2022 from http://shodor.org/petascale/

[18] Shodor. 2022. CSERD. Retrieved November 2022 from http://shodor.org/cserd/

[19] Shodor. 2022. HPC University. Retrieved November 2022 from http://www.hpcuniversity.org/

[20] Shodor. 2022. Interactivate. Retrieved November 2022 from http://shodor.org/interactivate/

[21] Shodor. 2022. Journal of Computational Science Education. Retrieved November 2022 from https://jocse.org/

[22] Shodor. 2022. Old Songs, New Words. Retrieved November 2022 from http://shodor.org/songs/

[23] Shodor. 2022. Petascale Institute Curriculum. Retrieved November 2022 from http://shodor.org/petascale/materials/institute/

[24] John Towns, Timothy Cockerill, Maytal Dahan, Ian Foster, Kelly Gaither, Andrew Grimshaw, Victor Hazlewood, Scott Lathrop, Dave Lifka, Gregory D. Peterson, Ralph Roskies, J. Ray Scott, and Nancy Wilkins-Diehr. 2014. XSEDE: Accelerating Scientific Discovery. *Comput. Sci. Eng.* 16, 5 (Sept.–Oct. 2014), 62–74. DOI:https://doi.org/10.1109/MCSE.2014.80

[25] University of Illinois Board of Trustees. 2022. Cyberinfrastructure Tutor (CI-Tutor). Retrieved November 2022 from https://researchit.illinois.edu/resource/71

[26] University of Illinois Board of Trustees. 2022. Blue Waters Training Resources. Retrieved November 2022 from https://publish.illinois.edu/bluewaters/blue-waters-training-and-outreach/blue-waters-training-resources/

[27] University of Illinois Board of Trustees. 2022. Blue Waters Webinars. Retrieved November 2022 from https://publish.illinois.edu/bluewaters/blue-waters-training-and-outreach/blue-waters-webinars/

[28] Video Conferencing Infrastructure: Distributed Synchronous Instructional Laboratories. Retrieved November 2022 from https://uofi.app.box.com/s/5cvnbznsv9yyo91f8fxarzxdi75sdydd

[29] Virtual School of Computational Science and Engineering: Lessons Learned. Retrieved November 2022 from https://uofi.app.box.com/s/1ffybmchwlz2uvsrkgacb5bpu9qalfhl

[30] Aaron Weeden. 2021. XSEDE EMPOWER: Engaging Undergraduates in the Work of Advanced Digital Services and Resources. *Journal of Computational Science Education* 12, 2 (February 2021), 22–24. DOI:https://doi.org/10.22369/issn.2153-4136/12/2/5

[31] Aaron Weeden, Colleen Heinemann, Skylar Thompson, Cameron Foss, Mobeen Ludin, and Jennifer K. Houchins. 2019. The Blue Waters Petascale Institute: Longitudinal Impact and Assessment-Driven Development of an Intensive, Hands-on Curriculum for Teaching Applications in HPC. In *PEARC '19: Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning)* 85 (July 2019), 1–8. DOI:https://doi.org/10.1145/3332186.3337771

Enhancing HPC Education and Workflows with Novel Computing Architectures

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

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

Jeffrey Valdez
Georgia Institute of Technology
Atlanta, Georgia
valdez@cc.gatech.edu

Sam Jijina
Georgia Institute of Technology
Atlanta, Georgia
sam.jijina@gatech.edu

Xueyang Liu
Georgia Institute of Technology
Atlanta, Georgia
xliu791@gatech.edu

Michael D. Weiner
Georgia Institute of Technology
Atlanta, Georgia
mweiner3@gatech.edu

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

Semir Sarajlic
Georgia Institute of Technology
Atlanta, Georgia
semir.sarajlic@oit.gatech.edu

ABSTRACT

Recent HPC education efforts have focused on maximizing the usage of traditional- and cloud-based computing infrastructures that primarily support CPU or GPU hardware. However, recent innovations in CPU architectures from Arm and RISC-V and the acquisition of Field-Programmable Gate Array (FPGA) companies by vendors like Intel and AMD mean that traditional HPC clusters are rapidly becoming more heterogeneous.

This work investigates one such example deployed at Georgia Tech — a joint workflow for processor design and reconfigurable computing courses supported by both the HPC-focused Partnership for an Advanced Computing Environment (PACE) and GT's novel architecture center, CRNCH. This collaborative workflow of HPC nodes and 40 remotely accessible Pynq devices supported over 100 students in Spring 2022, and its deployment provides key lessons on sticking points and opportunities for combined HPC and novel architecture workflows.

KEYWORDS

HPC education, Novel architecture workflows, Job scheduling, Field-Programmable Gate Arrays, Jupyter notebooks

1 INTRODUCTION

Many HPC education efforts have focused on extending student experiences from the traditional CPU-based systems found in their phones and laptops to novel accelerators like GPUs for applications like graphics programming and machine learning. At the same time, there are more instances of novel hardware that do not easily fit into our current understanding of what an HPC cluster is, including quantum computers, neuromorphic processors, and reconfigurable computing platforms like Field Programmable Gate Arrays (FPGAs).

These new types of platforms can vary from accelerators (like GPUs) to self-hosted devices like NVIDIA's Jetson platform or Xilinx's PYNQ Z-2 [29] board. In the case of the Xilinx PYNQ device, a $150 board provides a small FPGA fabric as well as a two-core Arm 32-bit CPU that can run a full Ubuntu operating system. Typically these boards are used by students in a hands-on fashion where

Figure 1. PYNQ Cluster as part of Georgia Tech's CRNCH Rogues Gallery Testbed.

https://doi.org/10.22369/issn.2153-4136/13/2/6

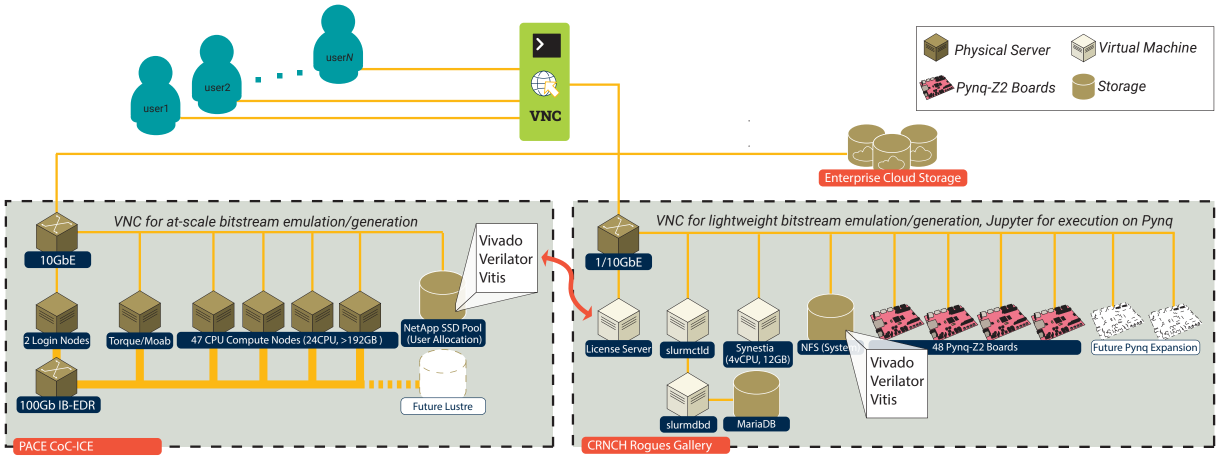

Figure 2. ICE and CRNCH RG Clusters.

they create a program using software on their laptop and then program the device using a USB connection. In fact, related FPGA clusters [3, 9, 18, 31] have recently focused on supporting these types of efforts using cameras and web interfaces for the hands-on portions of the boards. However, our goals for this infrastructure are mostly focused on supporting a larger number of students using scalable, community-driven tools and allowing them to interact with the devices using supported interfaces like Jupyter notebooks with a low barrier to entry.

These goals for an FPGA-based cluster are further stretched by rapid growth in our undergraduate and graduate education enrollment and supply chain issues that have meant that students cannot actually purchase the required number of boards for a single course. This leads us to the the following key challenges that we looked to address:

- How can we support larger numbers of students for a novel FPGA cluster that ties in with Georgia Tech's existing Instructional Cluster Environment (ICE) to support educational objectives?
- How do we support key requirements like data separation and privacy for student data while allowing for free flow of data between isolated clusters and local access points?
- What kind of training would students require to migrate from the "hands-on" infrastructure to a remotely scheduled cluster that uses a scheduler like Torque/Moab or Slurm?

This work describes our approach to answer these questions and offers our insights into the infrastructure design choices that worked well and which obstacles proved challenging to overcome. We also describe some student response and feedback that we anticipate using to improve the system configuration and user workflows over time. The described infrastructure will also be shared in an open-source fashion so that others may implement their own novel architecture clusters based on similar concepts.

2 CLUSTER OVERVIEW — ICE AND CRNCH RG

Figure 2 shows the layout of the two clusters that were used in Spring 2022 to support reconfigurable computing and computer architecture classes.

The PYNQ cluster is part of the NSF-funded novel architecture testbed, the Rogues Gallery [34], a center-based testbed that is focused on near-term "post-Moore" computing including next-generation HPC, neuromorphic, near-memory, and reversible computing amongst other topics. The PIs of this Center for Research into Novel Computing Hierarchies (CRNCH) work closely with Georgia Tech's high-performance computing organization, the Partnership for an Advanced Computing Environment (PACE). In a sense, the Rogues Gallery helps to prototype small, next-generation systems while PACE enables the large-scale deployment of cutting-edge HPC and HTC environments that include the NSF funded Hive project [20] and Buzzard project [2], as well as the Phoenix cluster [11] that ranked #277 on the Top500 November 2020 list.

In addition to the aforementioned research computing clusters, PACE also supports two instructional clusters, CoC-ICE and PACE-ICE [1]. CoC-ICE is a dedicated instructional HPC cluster administered as a collaboration between the College of Computing (CoC) and PACE and hosts courses in computing and, as appropriate, electrical engineering. Comparatively, PACE-ICE supports courses that teach and use scientific computing in engineering, physical and social sciences, and humanities and arts, plus training workshops in research computing offered by PACE to the entire campus community. Automated tools retrieve course registration lists from the registrar and provision student access and storage allocations. The heterogeneous CoC-ICE cluster is comprised of 45 nodes featuring Intel's Cascade Lake processors, and the cluster hosts 36 Tesla V100 and 24 RTX6000 Nvidia GPUs. All nodes are connected via a 100 Gbps InfiniBand network to support parallel computing. A NetApp device offers network storage to all students and instructors as well as shared space for instructors to distribute course materials or install custom software. As a part of the PACE ecosystem, CoC-ICE provides the full suite of scientific software and tools available through the PACE software stack and trains students in using an HPC cluster, including submitting jobs to a scheduler (Moab/Torque), loading software modules with lmod, and managing network and local storage.

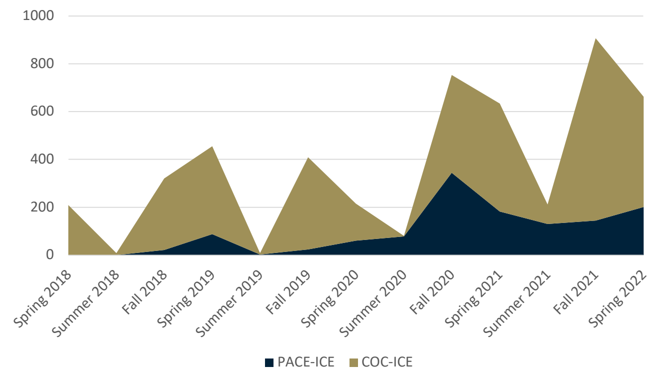

Figure 3. The CoC-ICE and PACE-ICE clusters have enhanced education for an increasing number of students since inception. The graph shows the number of students who ran jobs on each cluster during each semester, based on data from PACE's Open XDMoD [25] instance.

The instructional clusters have received significant demand and use from students and faculty. In 2021, the two ICE clusters hosted 20+ courses per semester and 60+ workshops, enabling over 1600 students to submit about 220,000 jobs and consume over 550,000 CPU hours. Since the launch of ICE in 2018, the clusters have supported course access for over 4,000 GT students. Figure 3 provides the breakdown of active students per semester, which is derived from the data presented by Open XDMoD [25].

More recently, the ICE clusters have seen an increase in demand for resources to support less traditional workflows and computational disciplines. Support requests from faculty include CPU/GPU heterogeneity for architecture-specific algorithm design, OpenGL-capable GPUs for simulation visualizations, and cloud-facing application interfaces to support hybrid activities, to name a few. Additionally, interest in ICE access for courses representing the so-called "long tail of science" [17] has been increasing. For example, ICE resources were utilized for "Data Analytics and Security" from the School of International Affairs and "Computational Musicology" from the School of Music, as highlighted in the Fall 2021 issue of the PACE Newsletter [21].

Through PACE's facilitation of courses on the ICE clusters and direct feedback from faculty teaching the courses, the major challenges that PACE observed and learned about involve preparing undergraduate students. Many students have little or no prior computing experience, especially on an HPC cluster, leading to issues with code editing and compilation, SSH access to and navigation within the ICE clusters, and batch submission system entry to compute resources. In working with faculty on facilitating their courses on ICE clusters, some have expressed that at least 50% of the in-class time is spent troubleshooting technical computing issues rather than focusing on the interesting scientific problems and applications.

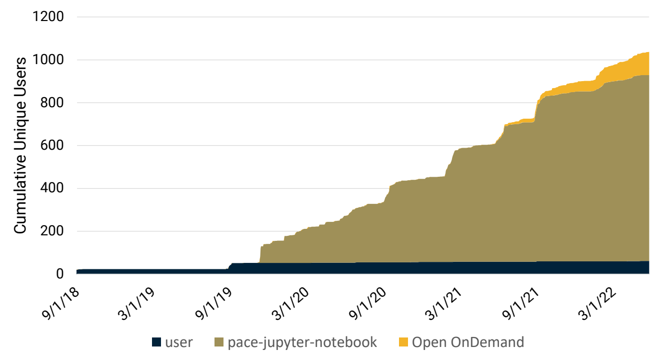

Figure 4. The deployment of PACE's Jupyter notebook wrapper in Fall 2019 eased access to Jupyter notebooks on PACE clusters and quickly achieved high utilization. Its use has continued to grow, and PACE's recent beta testing of Open OnDemand has added to the number of students and researchers using Jupyter notebooks to perform calculations on PACE clusters.

These challenges have driven the adoption of user-focused tools for interactive computing (e.g., Jupyter Notebook) as well as PACE-developed wrapper scripts for workflow abstraction [27, 28]. In late 2019, PACE deployed this prototype wrapper service across all PACE resources to make Jupyter Notebooks and VNC more accessible. This effort led to a rapid spike in the usage of Jupyter notebooks across resources as shown in Figure 4, where the number of unique users and the frequency of Jupyter notebook use quickly increased each month to several hundred unique users per month. Furthermore, these efforts set the stage for subsequent initiatives to continue lowering the barrier to entry and further democratize access to advanced research computing resources, such as the Apache Airavata [26] based Hive science gateway [19], and multiple Open OnDemand [8] instances for PACE clusters [22–24]. This has led to a strong collaboration with CRNCH's Rogues Gallery in deploying similar interactive computing wrappers and eventually Open OnDemand on the PYNQ FPGA and Arm-based Octavius [10] clusters.

3 CLASS REQUIREMENTS FOR NOVEL ARCHITECTURES

In contrast to the CPU and GPU hardware hosted in the ICE environments, novel architecture ecosystems used for class present challenges in scalability and resilience. In particular, FPGAs do not inherently provide the layer of indirection supported by CPU servers that is needed to allow multiple users to simultaneously utilize the underlying hardware resources; instead, this functionality must be set up by administrators as described in Section 4. Without this indirection, a separate FPGA board with power and networking would be required for each student. It is painfully apparent how quickly this methodology becomes unscalable.

The two classes that were targeted for integration into the combined ICE and CRNCH environments were an undergraduate CS class built around processor design (CS 3220) [15] and a graduate

ECE class focused on the fundamentals of parallel programming for FPGAs (ECE 8893) [7]. These two classes had an enrollment of 90 students and 45 students, respectively.

The primary objective of CS 3220 is to teach undergraduate computer science students the fundamentals of logic design and FPGAs by designing a 5-stage processor pipeline. The students achieve this goal by writing Verilog code, simulating their designs using Verilator [30] and Vivado [33] tools and then generating a bitstream that is targeted for the PYNQ Z2 FPGA [29] platform. To achieve this goal, we needed a cluster of PYNQ FPGA boards that could be networked and managed without user intervention and that could give the end user the perception that they had full control over a single FPGA board.

This processor design class utilized the PYNQ cluster for two homeworks (10 and 11) and class projects (4 and 5). Project 4 [13] involved students adding branch prediction with a Branch Target Buffer (BTB) to their pipeline design and then generating a wrapper for their design. This wrapper would allow their designed processor pipeline to interface a programmable logic (PL) design with the processor side (PS) of the PYNQ FPGA. Students would then generate the bitstream and use a Jupyter Notebook to interface with the bitstream and produce a result. Project 5 [14] involved students creating a edge filter with High-Level Synthesis (HLS) and performing optimizations to improve the design latency and resource utilization. Students also generated their own intellectual property (IP) using Vitis HLS, imported it into a Vivado project, generated the bitstream, and verified results on the PYNQ-Z2 boards. Homeworks 10 and 11 [12] provided supporting material and background to achieve the steps needed for successful completion of projects 4 and 5, including accessing the remote PYNQ FPGA cluster, bitstream generation, and Vitis HLS.

For ECE 8893, the primary objective is to teach graduate students the basic architecture of FPGAs and System-on-Chips (SoCs), HLS programming and algorithm design considerations, and application opportunities for FPGAs. These students leveraged the Xilinx Vivado tools and Vitis HLS [32] to map C code to FPGA-based IP and accelerators. As with CS 3220, the cluster of PYNQ FPGA boards was used to provide students with a managed device that could run the same workflows as if the user had access to the physical hardware.

Graduate students in ECE 8893 utilized the PYNQ cluster for lab 3 [6], which was comprised of two tasks to familiarize students with the hardware and workflow. The first task asked students to read a randomized array from DRAM, increment each element by 1, and write it back to DRAM. The second task required students to implement a portion of their final project (or lab 2 code if their final project did not include HLS programming). Both tasks required submission of the host code, bitstream, and hardware description file. The second task required a report summarizing the correctness of the result and a comparison of the execution time and the HLS synthesis report prediction.

4 DESIGNING A NEW STUDENT WORKFLOW FOR FPGAS

Traditionally, Xilinx PYNQ boards are single user devices that run a static Jupyter server as the root user, which provides the ability

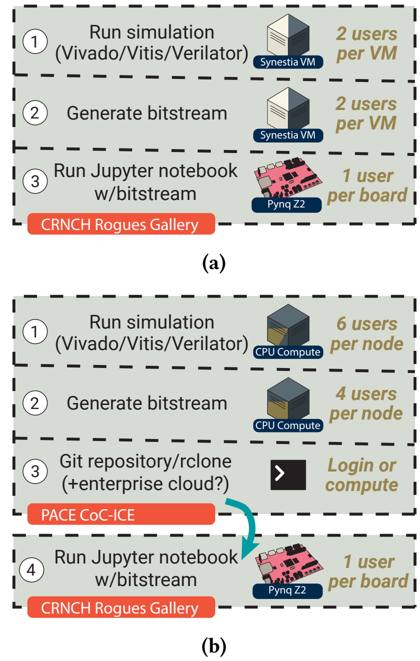

Figure 5. (a) Workflow on CRNCH Rogues Gallery only, with Synestia VMs and PYNQ-Z2 boards. (b) Workflow using PACE CoC-ICE compute nodes and CRNCH Rogues Gallery PYNQ-Z2 boards.

to leverage custom bitstreams and overlays through the Python interface. Typically the student runs as a sudo user, *xilinx*, which allows them to run the notebooks and program bitstreams directly into the programmable logic (PL) part of the board.

Figure 5a shows the design of our previous workflow from earlier semesters where students used a tool like x2go to connect to simulation and bitstream generation VMs in the *synestia* cluster. Then students would SSH or log in directly to a PYNQ board as the *xilinx* user and load their bitstream from a network shared folder.

As previously discussed, this VM setup was not feasible to scale up to over 100 students, and, due to supply chain limitations, there was no option for students to buy or borrow their own board for local usage. For these reasons, we focused on the development and integration of the workflow in Figure 5b, where students used CoC-ICE for simulation and bitstream generation and the Slurm-enabled PYNQ cluster for on-board testing and debugging.

4.1 Filesystem Considerations

As a student-focused project, this effort had several considerations related to student data and privacy that needed to be considered. Specifically, we cannot explicitly share data between the CoC-ICE file server partition and our CRNCH testbed.

The CoC-ICE filesystem is hosted on a NetApp device, providing each student with a 15 GB quota and home directory. Daily snapshots produce backups, and each student's directory is accessible only to that student for privacy. Students on the cluster are not associated with a particular course, only a school, in order to avoid revealing sensitive course enrollment information.

On the CRNCH-hosted cluster (*synestia* and the PYNQ boards), we created a single folder per class to allow students to have a consistent nethome on the login node and on the target PYNQ devices. This setup also allows for easily "resetting" the student nethome between semesters to preserve student privacy.

These two decisions do introduce one additional requirement for students since the two clusters do not share the same data sets. As show in Figure 5b, students need to save their bitstream to a Git repo on CoC-ICE and then pull it onto the CRNCH cluster before proceeding to testing with physical hardware. The instructors and TAs of the course provided information on using Git for this small transfer step as well as how to use rclone to save files to a student-owned Box or Dropbox folder, which are hosted via Georgia Tech's enterprise cloud services for students.

4.2 PYNQ Board Setup

The PYNQ boards run a 32-bit Arm-based version of Ubuntu 20.04 (PYNQ 2.7 image for Z-2 boards) while the hosting *synestia* VMs use 64-bit Ubuntu 18.04. CoC-ICE servers currently run Red Hat Enterprise Linux 7.6.

To deploy Slurm and appropriate settings for the PYNQ boards, we used Ansible scripts and hand-built deb files for Slurm 21.08 for both the PYNQ nodes and the Synestia VM. For the exam data collection in Section 6, a common Slurm database daemon (DBD) was used that is hosted to track and account for all CRNCH Rogues Gallery resources.

We also modified the local sudoers file to allow normal users to run the Jupyter notebook as a root user without a password. This tweak was made to allow users both to access bitstreams they generated and to program them on the local PYNQ board.

4.3 Adoption of PACE Wrapper Scripts for PYNQ Cluster

We adopted and modified the PACE wrapper script for Jupyter notebooks on the CoC-ICE cluster to support Slurm in place of Torque/Moab that the PACE clusters currently utilize. More specifically, we create an sbatch script that users call from a wrapper script that is shown in Listing 1. This script creates a Jupyter notebook instance on the PYNQ node and then returns the port forwarding information that a user can use to forward the remote port 8888 to a randomized local port that is then forwarded over an SSH tunnel.

Listing 2 shows the output of running this sbatch script from the user's perspective. The user launches a new Slurm job with a simple script, and they are then directed to open a Jupyter notebook at port 58786 in their local web browser.

In addition to scripting changes, we updated the Slurm prolog and epilog to link a user's nethome directory into the /root directory and then unlink it at the end of the job. This allowed students to run the job script, log in to a notebook as a root user, and run bitstreams from their nethome directory.

```
n+nbcd n+nv$SLURM_SUBMIT_DIR
n+nvPIPEFILEo=l+s+si${n+nv1l+s+si}
rm l+s+si${n+nvPIPEFILEl+s+si}
mkfifo l+s+si${n+nvPIPEFILEl+s+si}
o(jupyter notebook --no-browser
 ↪  --ipo=l+s+si${n+nvHOSTNAMEl+s+si} --porto=l+m8888
 ↪   l+m2>p&l+m1 p| tee l+s+si${n+nvPIPEFILEl+s+si} o) p&
kwhile n+nbread -r line
kdo
    n+nbecho l+s+s2"n+nv$linel+s+s2"
    kif o[[ l+s+s2"n+nv$linel+s+s2" o==
 ↪   l+s+s2"http://pynq-z2-"* o]]
    kthen
        n+nvHOSTo=k$(sed -e
 ↪   l+s+s1's#http\://\([^:]*\)\:.*#\1#'o<<<n+nv$linek)
        n+nvPORTo=k$(sed -e
 ↪   l+s+s1's#http\://[^:]\:\([0-9]*\)/?.*#\1#'o<<<n+nv$linek)
        n+nvTOKENo=k$(sed -e
 ↪   l+s+s1's#.*/?\(.*\)#\1#'o<<<n+nv$linek)
        n+nbbreak
    kfi
kdone < l+s+si${n+nvPIPEFILEl+s+si}
n+nbecho l+s+s2"Host: n+nv$HOSTl+s+s2"
n+nbecho l+s+s2"Port :n+nv$PORTl+s+s2"
n+nbecho l+s+s2"Token: n+nv$TOKENl+s+s2"
```

Listing 1. Slurm Jupyter notebook script.

5 DEPLOYMENT AND DEBUGGING

We deployed the infrastructure for the two courses and collected feedback from students throughout the two months when these classes were most active. With the feedback from earlier homeworks and projects, we improved the cluster robustness and resolved issues, including incompatible bitstreams and too many active user sessions. Most students in CS 3220 and ECE 8893 were able to verify their design on the cluster via homework and projects. We were also able to support an exam where 41 boards were available and 35 students were actively working on the cluster during a 3-hour time period.

A commonly-encountered issue was getting *[Error 110] Connection time out error* when loading a bitstream through the PYNQ overlay library. The TAs for 3220 helped to confirm that the root cause of this issue was using a very similar but slightly incompatible bitstream generated for another board that then broke the board logic. We also noticed that once a board was programmed with an incompatible bitstream, the board remained dysfunctional until it was rebooted. To resolve this issue, we experimented with using a cron script and Slurm epilog scripts to reboot the board at the end of every user session. The next generation of this infrastructure will allow for remote power cycling of these boards, which should also help to mitigate bad bitstreams by using cold reboots of the hardware.

We also observed that when there were no extra board resources available, a request to allocate new resources would sometimes fail with an SSH hostname error. We attempted to mitigate this error by

```
Submitting job via sbatch slurm-jupyter-notebook.batch...
Submitted batch job 14

Job successfully submitted!
Waiting for job to start

Starting jupyter notebook...

Connect to your jupyter notebook via the following steps:
    1) Press SHIFT + ~ then SHIFT + C to open an SSH console
↪   (The prompt 'ssh>' should appear on the next line)
      ***Note: '~' MUST be the first character on the line
↪   to be recognized as the escape character,in which case
↪   it will not appear on your terminal.***
      ***If you see the '~' character when you start typing,
↪   delete it, hint 'ENTER'and type 'SHIFT' + '~' + 'C'
↪   again.***
    2) Type -L 58786:pynq-z2-42:58786 and then ENTER
    3) Connect your browser to http://localhost:58786/ and
↪   enter 'xilinx' in the password prompt.
```

Listing 2. Slurm notebook script output.

providing more detailed feedback on failures in the wrapper scripts used to launch jobs.

At the beginning of the FPGA cluster deployment, we informed the students about the basics of Slurm infrastructure so that they understood how the cluster allocated resources and were able to monitor and manage their session. However, we still encountered situations where students would attempt to allocate multiple boards when a wrapper script initially failed to launch a job. To prevent this from happening, we asked students to check the status of their sessions with commands like *squeue* and to terminate any failed or idle jobs. This guidance worked well for students with a background in HPC systems, but a minority of students still struggled with using the new Slurm-based infrastructure. Ultimately, we determined the best course for future semesters is to provide more detailed Slurm training and examples as well as to set explicit job limits for students using Slurm accounting.

6 STUDENT OUTCOMES AND SENTIMENTS

While this infrastructure is brand new and should still be considered "beta" due to its accelerated deployment, we were able to pull a few statistics from the undergraduate course's final exam period using SlurmDBD statistics, and we report on student sentiment on the hardware based on a post-course survey.

6.1 Slurm Reporting for Final Exam

Slurm accounting information was collected in the Slurm database over the time frame of 4 hours (from 2pm–6pm) for a final exam. In this time frame, 132 jobs were started for 36 unique users. Jobs started in this 4-hour window ran for a mean of 40.72 minutes (sd = 35.17 minutes). At peak job workload during the time frame for the final exam, 36 jobs ran for 25 concurrent users (at 4:33pm). Table 1 demonstrates how a final exam can be a "peak workload" scenario using this PYNQ cluster.

Time	Unique Users	Jobs
02:00pm	0	0
02:30pm	3	3
03:00pm	11	12
03:30pm	18	24
04:00pm	19	24
04:30pm	25	33
05:00pm	25	34
05:30pm	18	24
06:00pm	13	16

Table 1. Number of unique users and jobs during final exam (2–6pm).

Eight students had more than two jobs running at once that we needed to limit manually in order to provide fair access to available FPGA resources. This points to a need for priority reservation for exam participants and also stricter limits on the maximum number of jobs run as enforced by Slurm accounting.

6.2 Survey Results

The survey results shown in Figure 6 demonstrate some of the challenges and opportunities of this cluster and the related integration of educational workflows. Students appreciated not needing to buy boards, but they also expressed frustration at needing to understand Slurm and PYNQ-related errors that had not typically been encountered in previous semesters.

Our main feedback from this survey is a positive one. We believe that this workflow is what students would like to engage with as long as we can improve the scripting and documentation to better support the dual workflow of simulation and synthesis with CoC-ICE and the physical execution on CRNCH RG FPGAs.

7 LESSONS LEARNED

Despite the short timeline for this deployment, we learned several lessons that will be useful for future joint workflow deployments:

(1) Each novel architecture typically has a workflow that involves simulation or emulation, and we can utilize traditional HPC via clusters like CoC-ICE and PACE-ICE to help accelerate this phase of the workflow for larger numbers of students.
(2) Data separation and scheduling constraints become more complex when student data is involved, but we can still use best practices from one cluster to help stand up and measure data from a novel architecture platform like the CRNCH RG PYNQ cluster.
(3) Undergraduate students typically struggle with learning new scheduling concepts that they have not encountered before in their academic career. One possible approach is to provide a standardized scheduling environment (i.e., just support one of Slurm or Torque/Moab), create examples and wrapper

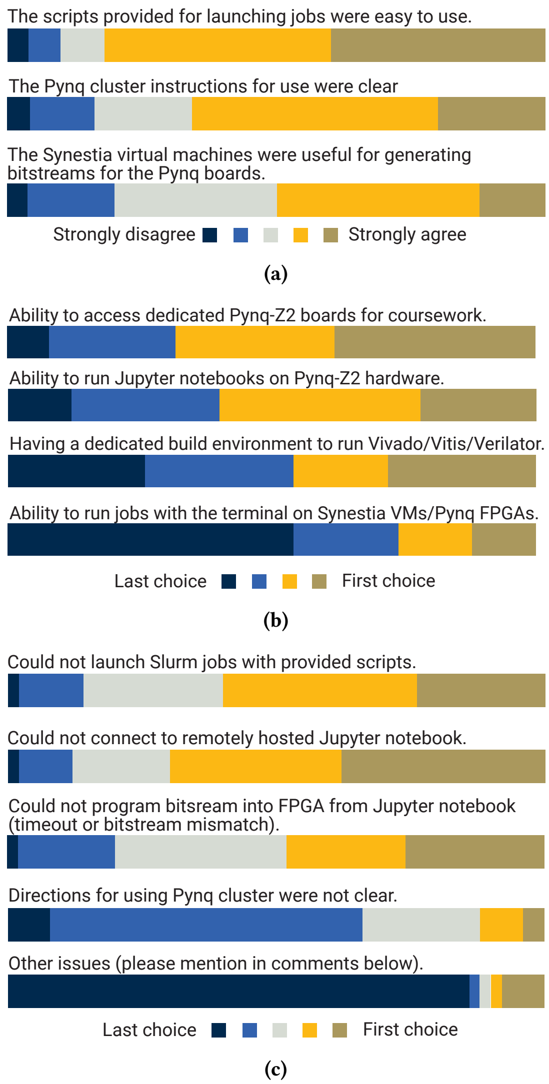

Figure 6. Student rankings of cluster (a) benefits, (b) features and (c) challenges.

scripts for students, and simplify their workflow as much as possible when multiple clusters are used.

8 RELATED WORK

While we believe that this particular implementation of a PYNQ-based FPGA cluster is unique due to its focus on interactive scalability, there are several other research- and educational-oriented clusters that have focused specifically on FPGAs.

At a vendor level, Xilinx supports its Heterogeneous Accelerated Compute Clusters (HACC/XACC) at universities in the US and abroad, and Intel provides its DevCloud for remote access by researchers. It is notable that both of these infrastructures are set up for small numbers of users to join at once and may not be suitable for large, transient student populations. ECELabs.io [16] is a recent cluster that provides students with a "visual" representation of devboards like PYNQ and Nexys FPGAs where students can remotely toggle the buttons on an actual board instance and see the LED output from the board. As mentioned earlier, there are also several interesting FPGA cluster efforts [3, 9, 18, 31] that are focused on supporting cameras and web-based interfaces for hands-on usage of these small devices. These efforts are not mutually exclusive to the Slurm- and Jupyter-based workflow presented here, and our hope is that our public documentation of this work will allow for the wider adoption of both hands-on and scalable approaches for these clusters.

In addition to these vendor efforts, there are many interesting research projects focused on using larger FPGA boards like Xilinx Alveo and Intel's Arria 10 to build larger clusters, including efforts to support data analytics with DASK, DASK on Alveo boards with some PYNQ scaling, and the GreenFlash FPGA cluster [4] for providing real-time control of adaptive optics for the Extremely Large Telescope (ELT). Testbeds focused on using Alveo boards include the Open Cloud Testbed run by Northeastern University [5].

9 CONCLUSIONS

This effort demonstrates the power of traditional HPC along with adaptations for future novel architectures, including FPGAs.

From this work, we expect that several follow-on efforts will continue to improve both CoC-ICE and the CRNCH testbed, including combined support for Slurm and integration of Slurm accounting data with PACE's Open XDMoD [25] instance for metrics collection and monitoring. Deeper integration with XDMoD would allow us to better measure the impact of Pynq cluster. We do need better scripts to help synchronize Slurm accounting and LDAP groups, but this would likely be a joint concern for both clusters.

Secondly, we believe that the addition of new services like Open OnDemand on CoC-ICE and the CRNCH testbed will likely simplify the job launch process since it eliminates the need for users to use and understand SSH port forwarding. As we grow CoC-ICE to 75 nodes in the upcoming academic year, we anticipate that both clusters will benefit from continued joint efforts to simplify and standardize workflows for both traditional HPC and novel architectures.

ACKNOWLEDGEMENTS

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] Mehmet Belgin, Trever C. Nightingale, David A. Mercer, Fang Cherry Liu, Peter Wan, Andre C. McNeill, Ruben Lara, Paul Manno, and Neil Bright. 2018. ICE: A

Federated Instructional Cluster Environment for Georgia Tech. In *Proceedings of the Practice and Experience on Advanced Research Computing* (July 2018) *(PEARC '18)*. Association for Computing Machinery, New York, NY, USA, Article 16, 7 pages. https://doi.org/10.1145/3219104.3219112

[2] Mehmet Belgin, Semir Sarajlic, Ruben Lara, Laura Cadonati, Nepomuk Otte, Ignacio J. Taboada, Gregory L. Beyer, Norman B. Bonner, Michael Brandon, Pam Buffington, J. Eric Coulter, Aaron Jezghani, David Leonard, Fang (Cherry) Liu, Paul D. Manno, Craig A. Moseley, Trever C. Nightingale, Ronald Rahaman, Kenneth J. Suda, Peter Wan, Michael D. Weiner, Deirdre Womack, Dan Zhou, Marian Zvada, Andre C. McNeill, Neil C. Bright, Robert W. Gardner, Paschalis Paschos, Lincoln Andrew Bryant, Judith Lorraine Stephen, James Alexander Clark, Peter F. Couvares, Brian Hua Lin, Todd Tannenbaum, and Gregory Thain. 2022. Buzzard: Georgia Tech's Foray into the Open Science Grid. In *Practice and Experience in Advanced Research Computing* (July 2022) *(PEARC '22)*. Association for Computing Machinery, New York, NY, USA, Article 47, 5 pages. https://doi.org/10.1145/3491418.3535135

[3] Vassilis Fotopoulos, Anastasios Fanariotis, Theofanis Orphanoudakis, and Athanassios N. Skodras. 2015. Remote FPGA Laboratory Course Development Based on an Open Multimodal Laboratory Facility. In *Proceedings of the 19th Panhellenic Conference on Informatics* (October 2015) *(PCI '15)*. Association for Computing Machinery, New York, NY, USA, 447–452. https://doi.org/10.1145/2801948.2801950

[4] D Gratadour, M Andrighettoni, G Angerer, A Basden, J Bernard, R Biasi, U Bitenc, T Buey, R Dembet, H Deneux, et al. 2017. Green Flash: Exploiting future and emerging computing technologies for AO RTC at ELT scale. In *AO4ELT5: Adaptive Optics for Extremely Large Telescopes 5* (January 2017). https://doi.org/10.26698/AO4ELT5.0108

[5] Suranga Handagala, Martin Herbordt, and Miriam Leeser. 2021. OCT: The Open Cloud FPGA Testbed. In *31st International Conference on Field Programmable Logic and Applications (FPL)*.

[6] Cong (Callie) Hao. 2022. ECE 8893 Lab 3 Description. https://github.com/sharc-lab/FPGA_ECE8893/tree/main/2022_Spring/Lab3

[7] Cong (Callie) Hao. 2022. Georgia Tech ECE 8893 Course Page (webpage). https://github.com/sharc-lab/FPGA_ECE8893

[8] David E. Hudak, Douglas Johnson, Jeremy Nicklas, Eric Franz, Brian McMichael, and Basil Gohar. 2016. Open OnDemand: Transforming Computational Science Through Omnidisciplinary Software Cyberinfrastructure. In *Proceedings of the XSEDE16 Conference on Diversity, Big Data, and Science at Scale* (July 2016) *(XSEDE16)*. Association for Computing Machinery, New York, NY, USA, Article 43, 7 pages. https://doi.org/10.1145/2949550.2949644

[9] Rania Hussein and Denise Wilson. 2021. Remote Versus In-hand Hardware Laboratory in Digital Circuits Courses Remote versus In-Hand Hardware Laboratory in Digital Circuits Courses. In *2021 ASEE Virtual Annual Conference (ASEE 2021)*.

[10] Aaron Jezghani, Kevin Manalo, Will Powell, Jeffrey Valdez, and Jeffrey Young. 2022. Onboarding Users to A64FX via Open OnDemand. In *International Conference on High Performance Computing in Asia-Pacific Region Workshops* (January 2022) *(HPCAsia 2022 Workshop)*. Association for Computing Machinery, New York, NY, USA, 78–83. https://doi.org/doi/10.1145/3503470.3503479

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

[12] Hyesoon Kim. 2022. CS 3220 Homework 10+11 Description. https://github.com/gt-cs3220/gt-cs3220.github.io/blob/master/hw10.md

[13] Hyesoon Kim. 2022. CS 3220 Project 4 Description. https://github.com/gt-cs3220/gt-cs3220.github.io/blob/master/project4.md

[14] Hyesoon Kim. 2022. CS 3220 Project 5 Description. https://github.com/gt-cs3220/gt-cs3220.github.io/blob/master/project5.md

[15] Hyesoon Kim, Bhanu Garg, Anurag Kar, and Xueyang Liu. 2022. Georgia Tech CS3220 Course Page (webpage). https://gt-cs3220.github.io/

[16] Junfei Li. 2021. ECELabs.io website. https://www.ecelabs.io

[17] Rob Mitchum. 2012. Unwinding the 'Long Tail' of Science. https://voices.uchicago.edu/compinst/blog/unwinding-long-tail-science/

[18] Abd El-Rahman Mohsen, Mohamed Youssef GadAlrab, Zeina elhaya Mahmoud, Gameel Alshaer, Mahmoud Asy, and Hassan Mostafa. 2019. Remote FPGA Lab For ZYNQ and Virtex-7 Kits. In *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)* (August 2019). 185–188. https://doi.org/10.1109/MWSCAS.2019.8885064

[19] PACE. 2021. Georgia Tech Hive Gateway. https://gateway.hive.pace.gatech.edu Retrieved May 29, 2022.

[20] PACE. 2021. Georgia Tech Hive project. https://pace.gatech.edu/new-hpc-cluster-hive

[21] PACE. 2021. PACE Newsletter - Fall 2021. https://pace.gatech.edu/sites/default/files/pace-newsletter-fall2021.pdf. Retrieved May 27, 2022.

[22] PACE. 2022. Hive Open OnDemand. https://ondemand-hive.pace.gatech.edu/. Retrieved May 29, 2022.

[23] PACE. 2022. PACE-ICE (Beta) Open OnDemand. https://ondemand-pace-ice.pace.gatech.edu/. Retrieved May 29, 2022.

[24] PACE. 2022. Phoenix Open OnDemand. https://ondemand-phoenix.pace.gatech.edu/. Retrieved May 29, 2022.

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

[26] Marlon Pierce, Suresh Marru, Lahiru Gunathilake, Thejaka Amila Kanewala, Raminder Singh, Saminda Wijeratne, Chathuri Wimalasena, Chathura Herath, Eran Chinthaka, Chris Mattmann, Aleksander Slominski, and Patanachai Tangchaisin. 2014. Apache Airavata: Design and Directions of a Science Gateway Framework. In *2014 6th International Workshop on Science Gateways* (June 2014). 48–54. https://doi.org/10.1109/IWSG.2014.15

[27] 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* (July 2016) *(XSEDE16)*. Association for Computing Machinery, New York, NY, USA, Article 54, 5 pages. https://doi.org/10.1145/2949550.2952770

[28] 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* (July 2017) *(PEARC17)*. Association for Computing Machinery, New York, NY, USA, Article 71, 4 pages. https://doi.org/10.1145/3093338.3104178

[29] TUL. 2022. Pynq-Z2 FPGA. https://www.tulembedded.com/FPGA/ProductsPYNQ-Z2.html

[30] Veripool. 2022. Verilator. https://www.veripool.org/verilator/

[31] Marco Winzker and Andrea Schwandt. 2019. Open Education Teaching Unit for Low-Power Design and FPGA Image Processing. In *2019 IEEE Frontiers in Education Conference (FIE)* (October 2019). 1–9. https://doi.org/10.1109/FIE43999.2019.9028694

[32] Xilinx. 2022. Vitis. https://www.xilinx.com/products/design-tools/vitis/vitis-platform.html

[33] Xilinx. 2022. Vivado. https://www.xilinx.com/products/design-tools/vivado.html

[34] 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)* (November 2019). 1–8. https://doi.org/10.1109/ICRC.2019.8914707

December 2022

Volume 13 Issue 2

ISSN 2153-4136 (online)