The Supercomputer Institute: A Systems-Focused Approach to HPC Training and Education

J. Lowell Wofford
Los Alamos National Laboratory
Los Alamos, NM
lowell@lanl.gov

Cory Lueninghoener
Los Alamos National Laboratory
Los Alamos, NM
cluening@lanl.gov

ABSTRACT

For the past thirteen years, Los Alamos National Laboratory HPC Division has hosted the Computer System, Cluster and Networking Summer Institute summer internship program (recently renamed "The Supercomputer Institute") to provide a basis is cluster computing for undergraduate and graduate students. The institute invites 12 students each year to participate in a 10-week internship program. This program has been a strong educational experience for many students through this time, and has been an important recruitment tool for HPC Division. In this paper, we describe the institute as a whole and dive into individual components that were changed this year to keep the program up to date. We also provide some qualitative and quantitative results that indicate that these changes have improved the program over recent years.

KEYWORDS

training, education, recruiting, student programs, system management

1 INTRODUCTION

For the past thirteen years, Los Alamos National Laboratory HPC Division[10] has hosted the Computer System, Cluster and Networking Summer Institute (CSCNSI)[4][1] summer internship program to provide a basis is cluster computing for undergraduate and graduate students. The institute invites 12 students each year to participate in a 10-week internship program. The program is aimed at students interested in a broad range of HPC related fields, but provides a systems design and management focused curriculum. A number of recent articles have proposed training programs in HPC[15, 18], but these programs have been focused on applications and have only scratched the surface of lower–level HPC systems. We believe that the inclusion of a systems focused program can provide depth and perspective to many students, regardless of the HPC related field they intend to enter.

The institute breaks the program into two parts: (1) a "boot camp" running approximately two weeks covering fundamentals of cluster computing; and, (2) an eight-week-long guided research project. At the end of the program, students present their research in both a short talk and a poster.

The boot camp curriculum was significantly redesigned in the last year around a new methodology. Whereas in previous years, the boot camp largely consisted of guided projects to set up a small compute cluster, this year took a more directed education approach to teach the fundamentals of cluster computing before starting research. The theory in these changes was: (1) a ground-up foundation in cluster computing, starting with basic Linux skills, building to how clusters are designed and built, and then building and running parallel applications on them will provide a strong basis for any area of future HPC related research; (2) a curriculum based on practical guides with occasional theory lectures will provide a stronger foundation than self-guided projects; (3) frequent feedback through anonymous as well as named survey evaluations allow for day-by-day adjustments to the curriculum.

This approach has proven very successful based on comparative analysis of survey results from both students and project mentors, as well as the quality and complexity of the research results achieved in this institute. In this paper, we will layout the structure of the institute, the motivations, and changes made to the boot camp curriculum and qualitative and quantitative analysis of the institute outcomes. Our focus will be the evaluation of the impact this ground-up foundation in cluster computing has on subsequent student research. Because the new curriculum has only been provided for one year, the sample size of students is relatively small, however, the results suggest a strong positive impact on both students' assessment of the program and students' productivity in the research portion.

2 THE CSCNSI

2.1 Overview

The CSCNSI summer program is a 10-week paid summer internship sponsored by the High Performance Computing Division at Los Alamos National Laboratory (LANL). Each year's program starts in the fall with a recruitment and application process, from which 12 participants are selected based on qualities such as their existing skills, their current progress in school, and interests they express in their application materials. In parallel, HPC Division staff members propose projects that they would like to have CSCNSI students work on during the upcoming summer. Four projects are selected each summer, and each project is assigned a team of three students. When the participants arrive for the program, their teams and their project/mentor matches are already defined and they are immediately ready to start the program.

[1] Since August 2019, the CSCNSI has been renamed "Supercomputer Institute."

https://doi.org/10.22369/issn.2153-4136/11/1/12

The first two weeks of the program consist of a "cluster boot camp". This portion of the program focuses strongly on building base HPC systems knowledge, and includes work with bare metal hardware; booting and provisioning systems; system configuration and management; developing and running parallel applications; and looking at current and future HPC technology. For this portion of the program, each student team is given a 10-node HPC cluster to work with. Each team starts with an uncabled cluster with blank disks, and by the end of the two weeks they have built a fully-functioning 10-node, Infiniband–connected cluster that is capable of running real-life HPC applications.

During the remaining 8 weeks of the program, each team works with their assigned mentors on the project that was selected for their team. These projects normally make extensive use of the clusters the teams have just built, and may involve building and benchmarking parallel filesystems; writing and testing parallel applications; writing system software related to monitoring, containers, scheduling, or other operating system level topics; testing known security flaws for exploitability and to find mitigations; evaluating new networking technology; or almost any other topic of interest to HPC researchers. At the conclusion of this directed research period, each team gives a 20-minute presentation on their work to the HPC community at the Laboratory as part of the yearly HPC Division Student Showcase. A catalog of past projects and posters can be found at [5].

2.2 Process

The process of preparation for each year's CSCNSI session consists of three main sections: the student selection process, including recruiting, interviewing, and selecting participants in the program; the mentor and project proposal process, which results in the projects that will be worked on in the program; and the project matching process, in which selected students are matched with mentors and projects that fit their interests and skills.

The student selection process begins with recruiting in the fall of the year before a particular CSCNSI session. This recruiting occurs in conjunction with HPC Division's regular student recruitment activities at conferences, including the Grace Hopper Celebration[6], the Richard Tapia Celebration of Diversity in Computing[1], the International Conference for High Performance Computing, Networking, Storage, and Analysis[8], and at university site visits. These recruiting trips include on-site interviews and the ability for HPC Division staff to make spot offers to highly qualified candidates.

Meanwhile, the CSCNSI program is open to applications from other students via its website[4]. Applications typically open at the start of the fall semester, and application materials are typically due by early December. At the close of this application period, a selection committee from HPC Division reviews all applicants and performs phone interviews with the strongest candidates. Offers are made to selected candidates, and these participants are combined with any spot offers made at recruiting trips to make that summer's 12-member CSCNSI class.

In parallel with this process, potential mentors from HPC Division's technical staff propose projects for teams within the program. This process begins with a call for proposals from across the Division requesting the project title and a short abstract describing the project's goals and benefits, as well as the proposed mentors, any extra hardware that would be required by the project, and skills that are needed by students who would work on this project. These proposals are evaluated based on their technical merit, their ability to produce results by the end of the 10-week program, and their ability to expose the students to new technology. They are also evaluated alongside the applicant pool to ensure the skills needed by each project can be met by the selected students each summer. At the end of this process, four projects are selected to be worked on that summer.

After the participants and projects have been selected, the final step of the process is to build four three-person teams and assign them to the selected projects. This is done by comparing knowledge of students' backgrounds and interests gathered from their interviews, resumes, and application materials with the project required skills specified by the mentors, with the goal of matching students with projects that will offer them an opportunity for growth and an opportunity to be successful.

2.3 Technical

The CSCNSI is a multi-discipline program that starts with a two-week "bootcamp" that focuses on HPC systems hardware and software. Each team is supplied with a 10-node, Infiniband–connected cluster, and over the course of the bootcamp they learn to build their cluster from scratch. Their clusters start out as bare hardware: the nodes are racked, but all of the network and power cables are in a box in the rack. Starting with how to properly label and cable a rack of computers, the students spend their bootcamp period installing the operating system, installing scientific libraries, automating the node build process, and finally running MPI applications across all of their nodes. The process gives the students a strong understanding of the underlying technology that makes an HPC system work. The bootcamp curriculum is described in more detail in sections 3 and 4.

With their clusters built, the student teams are ready to work on their main project. The technologies used in these projects vary greatly depending on their focus. Recent projects include transparently running user application containers; testing the overhead incurred by compute node health checks; finding security anomalies in network flows; and testing the overhead introduced by speculative execution exploit fixes. Each of these projects digs deeply into individual aspects of system hardware and software, building on the base that the students learned during their bootcamp session.

3 THE CURRICULUM

3.1 History & motivation

The CSCNSI boot camp curriculum ("the curriculum") has grown organically during its long history, sometimes going for multiple years with only small changes, while at other times receiving large rewrites to update the material to better match updated technology. The program's instructor role has passed between multiple people in recent years, resulting in a series of updates that weren't necessarily self consistent, and this year a decision was made to do a major rewrite of the material. Using the existing material as a

topical guide, a new curriculum was built that included update technologies, removed outdated information, and more closely matched the realities of today's HPC environments.

Additionally, the previous approach to the boot camp left the students to mostly explore the different topics informally on their own, with little guidance. While this informal approach has some strong learning benefits, student surveys and previous instructor feedback indicated that this left some teams struggling to have viable systems for their subsequent research. Additionally, given the rapid pace of the boot camp, this approach severely limited the depth to which certain topics could be explored. All of these factors suggested that a new approach to the curriculum that merged both formal an informal learning may be beneficial.

To achieve a more guided approach to the curriculum, a significant amount of new material was required. For the 2019 curriculum, over 200 pages of technical guides and roughly 300 lecture slides were developed. These materials have been made public, and can be found at [9].

3.2 Methodology

The objective of the new curriculum, aside general updates and improvements, was to provide more formal learning components than the previous curriculum. This would allow the students to achieve the practical objectives of the boot camp—getting their teams' compute clusters deployed into a useable state—while also allowing more depth to be explored in more topics. Meanwhile, we did not wish to lose the learning benefits of the previous largely informal learning approach.

The previous curriculum split the boot camp into lecture and lab segments. The lecture segments were generally very short, with one to two presented per day. The lab segments would consist of an unguided list of tasks. Teams would go off to achieve these tasks with as-needed assistance by the instructors.

The alterations in the approach of the new curriculum were two fold: (1) to extend the content of the lectures to include more technical depth and more technical areas; (2) to replace the labs with "practica." These practica take the form of staged guides that have a mix of free exploration prescribed steps. These guides will be explained in more detail below.

At a high level, the boot camp curriculum builds the students' skills in stages. Since students come from diverse backgrounds with varied experience, we start with basic skills in using and installing the GNU/Linux operating system. By the end of the 12-day curriculum, the students have fully functional Linux compute clusters controlled through configuration management and using industry-standard HPC tools for provisioning, monitoring, and resource management. Students are introduced to a combination of facilities, systems, programming and visualization concepts, and tools.

Organizationally, the curriculum was divided into chapters. Each chapter begins with a theory lecture, followed by practical written guides, or practica. Most of the time is spent working through these guides. The guides are further subdivided into steps. It is expected that all of the students work through the guides and synchronize at the end of each step. This helps ensure that the entire class stays roughly on the same content throughout. Maintaining synchronization of the students is important for both efficiency in teaching and assistance, as well as making sure that students are focused on the same kinds of tasks at the same time. Keeping students in sync means that questions from other students remain timely and relevant, and other students are actively working on the same projects, and therefore are more prepared to assist fellow students. During each step the instructor and assistants help teams that had questions or were stuck with portions of the guide. At the end of each step, the instructor summarizes the step, performs the step on an example cluster, handles any high-level questions related to that step, and briefly introduces the goals of the next steps. For most guides, each step has an accompanying slide with additional notes for that step.

To keep the more advanced students occupied as well as introduce more advanced concepts such as advanced shell scripting, for most guide steps a "challenge" problem was assigned. These challenge problems leverages material from the section, as well as requiring some outside information that the students must research. Examples of challenge problems include: using the "find" command to do a recursive find-and-replace operation and writing a shell script to do a ping scan on a network. Teams that finished the challenge were asked to present their solutions to the group, along with explanations, and group discussion of the different solutions was encouraged.

4 CURRICULUM OVERVIEW

The curriculum for the boot camp is divided into 11 chapters. See Table 1 for a syllabus of the curriculum. For the condensed two-week boot camp, each chapter approximately represents the curriculum content for one day. Each chapter is designed to both add relevant HPC skills and further the process of bringing the teams' 10-node compute clusters into a usable and maintainable state for the later research portion. Below we outline the curriculum by chapter, for each chapter summarizing the structure, content and motivations for that chapter. The chapters fall into logical groupings based on their overarching objective. We have broken them out by these groupings below.

4.1 Introducing HPC

The first two chapters of the curriculum provide a general introduction to the course and some higher level concepts of high-performance computing, systems, hardware, workflows, and facilities. These chapters provide a backdrop and motivation for the remainder of the course, and the ideas introduced in these chapters are designed to develop throughout the course. There is an emphasis on the kinds of problems that HPC helps to solve, how to design systems to solve these problems, and the subsequent challenges of these system designs.

4.1.1 Chapter 1: Introduction to HPC. This chapter provides the motivation for the rest of the course. While the course works by building up HPC systems knowledge from the ground up, the introduction takes a top-down approach to understanding HPC. In the introduction, we focus on the kinds of problems that scientists may need to solve. We then lay out how cluster computing designs provide an effective architecture for solving these problems. This helps to motivate the course by starting with a focus on research

	Title	Purpose	Practical
Chapter 1	Introduction to HPC	Overview of HPC systems, hardware, workflows, and facilities.	N/A
Chapter 2	HPC Facilities	Space, power & cooling challenges for HPC.	Cable and label cluster racks
Chapter 3	Exploring Linux	Basic Linux system operating system concepts, install, use, and administration.	Master node is installed with Linux.
Chapter 4	Networks & Services	Basics of netoworking and common Linux services.	Master network, NAT, ssh configured.
Chapter 5	Netboot provisioning	How to stateless netboot a node from scratch.	Some nodes provisioned, 1st pass.
Chapter 6	HPC provisioning	Using HPC provisionig tools to provision the whole cluster.	All nodes provisioned, 2nd pass.
Chapter 7	HPC tools	Overviews of common HPC tools for system management, scheduling & fabric management.	Clusters configured with workload management, high-speed network, power, and console control. First jobs run.
Chapter 8	Version Control & Config Management	Learn version control and configuration management tools and motivations.	Clusters re-provisioned, configured with version controlled configuration management.
Chapter 9	Monitoring & Benchmarking	Overview of tools used for benchmarking and active/passive monitoring clusters.	Monitoring and log analysis framework installed. Baseline benchmarks taken, system verified.
Chapter 10	Parallel & Cluster Programming	Introduction to parallel programming concepts and challenges. Introduction to cluster programming with MPI. Visualization tools.	Job submissions and MPI functionality tests. First parallel runs. Real scientific application run & visualized.
Chapter 11	Future technology	Discuss revolving topics of future interest to HPC.	N/A

Table 1: Syllabus for boot camp curriculum. The title, purpose of the chapter are given, and practical lessons are given for each chapter. Shading represents groupings used in section 4.

problems, motivating the general cluster architecture, discussing some important particular details of that architecture, and then working on the tools to practically build a system with a clustered architecture.

The toy research problems that are used as motivation for the *Introduction to HPC* chapter reappear in later chapters as job and programming examples that can be practically run on the systems that the teams deploy throughout the boot camp. This aims at keeping the students focused on why the systems are being built while constructing them in stages from the ground up.

4.1.2 Chapter 2: HPC facilities. The general discussion on HPC system design in the previous chapter naturally segues into a discussion of the kinds of physical, power, and cooling concerns surrounding large clusters of computers. The second chapter overviews the HPC facilities topics.

The HPC facilities introduction also provides the first practical lesson for the students. As part of the facilities introduction, the students are introduced to particular racking, cabling, and data center organization techniques. With this introduction, the students are then guided through physically cabling and labeling their teams' clusters[2].

4.2 System Management

While we do require some Linux experience for admission to the program, the level of Linux experience has varied widely among the students. To achieve a baseline of knowledge in Linux, the chapters 3 and 4 cover some of the basic Linux skills required for HPC, ranging from basic commandline skills to basic network and system service configurations. Throughout the guides for these chapters, "challenge" questions are offered to students who finish early to start building shell scripting knowledge. Each question pushes the students to find a new way to explore the Linux system by writing a script.

4.2.1 Chapter 3: Exploring Linux. This chapter is longer than other chapters and spanned two days. We begin with a lecture on an overview of Linux. This lecture splits into three parts. The first part covers some history of Linux as well as Linux and open source community issues. It also touches on why we use Linux for HPC. The second part of is focused on the Linux kernel and operating system theory. The third part provides an overview of Linux distributions.

Discussion of distributions in the lecture leads to a lab where the students install CentOS Linux[3] on their cluster master nodes following a basic install guide. Students perform the rest of the work throughout the bootcamp on this system.

Following the install procedure, students work through two guided practica on using Linux. Students are instructed in the use of the tmux[13] tool to share terminal sessions across their individual workstations. The first guide covers a wide range of common Linux tools with an emphasis on tools of particular use in HPC environments. The second guide focuses on those tools dedicated to inspecting the Linux system status and health.

[2]For our boot camp, due to time and safety concerns, the clusters are pre–racked but un–cabled when the students arrive.

4.2.2 Chapter 4: Networks & Services. Chapter 4 continues the exploration of the Linux. A beginning lecture covers fundamentals of networking and Linux networks, as well as Linux network services.

The lecture is followed by a guide that explores setting up and using various network settings and services in Linux. An emphasis is put on verification steps as each configuration step is performed. This guide also includes an exploration of systemd and service unit files. At the end of this guide, the master nodes have a complete network configuration and NTP, SSH and nginx services have been configured.

4.3 Cluster Provisioning & HPC Tools

Chapters 5 through 8 center around cluster provisioning. This is done in three stages. The theory is to start by provisioning manually, and add useful layers of abstraction in stages. First, the system is provisioned by manually creating a stateless booting cluster using common services and a combination of provided and student-developed scripts. Next, the system is re-provisioned using a common cluster provisioning system (Warewulf[14]). In the third iteration, the systems are re-provisioned again using configuration management (Ansible[2]) in conjunction with cluster provisioning (Warewulf). Chapter 7 is injected in the middle of this sequence to introduce core HPC tools not related to provisioning, such as the workload manager, before moving on to the final stage of provisioning. At the end of Chapter 8 the teams should have fully-functional, useable compute clusters.

4.3.1 Chapter 5: Netboot provisioning. Chapter 5 consists of one long guide that steps the students through everything necessary to perform a stateless (diskless) network boot of a compute node. This follows directly on the discussion of network and services in the previous chapter, and configures the core services (DHCP and tftpd) required to perform a PXE boot. The students are also guided through the process of manually constructing a node image to be provided to the compute nodes. Finally, the students are given a base initramfs image[3] that they can use to construct the staged boot required by most stateless compute clusters. This simplified initramfs has been constructed with the intent of educating, so the provided init stage scripts choose simplicity and readability over features. By the end of this chapter, the teams' clusters have two nodes provisioned using this method, in addition to the manually installed master node.

4.3.2 Chapter 6: HPC provisioning. Chapter 6 builds on Chapter 5 by showing how HPC provisioning systems, in this case Warewulf[14] can be used to dramatically simplify the process that was worked through in Chapter 5. Because Warewulf simplifies the netbooting process, this also affords the opportunity for the teams' to build more configuration complete and feature rich images for their nodes. At the end of this section, the entire cluster has been provisioned with Warewulf. In order to simplify access to packages and HPC tools, the OpenHPC project[11, 17] is used for additional HPC software repositories.

[3]The initramfs source can be found under the "Supplements" folder in the curriculum materials repository at [9]

4.3.3 Chapter 7: HPC tools. Up until this point, the students have not been introduced to some of the fundamental tools for HPC. It is useful to pause to look at some of these tools before the final provisioning step in order to make them available in the final provisioning of the clusters.

Several tools are introduced in this section that provide console and power access to the nodes and InfiniBand fabric management. Particular attention is paid to workload management and scheduling. Using the existing Warewulf install, the Slurm workload manager is installed, configured, and tested.

A final section of this chapter provides a short guide for working with Slurm as a user. This includes various forms of job submission, job inspection, and batch job scripting.

4.3.4 Chapter 8: Version Control & Config Management. The final step the provisioning process involves moving all of the work that has been done into a version controlled repository containing a configuration management specification. Git is used for version control and Ansible is used for configuration management.

The chapter begins with a short lecture covering Git concepts, followed by a practical learning guide for basic Git usage. Next, a lecture is given on general configuration management concepts with an emphasis on Ansible. The Ansible practical guides are divided in two. The first guide teaches basic practical Ansible examples. The second of the Ansible guides takes the users through the process of re-provisioning their clusters with Ansible and Git. A base Ansible repository is provided to the teams, and they are left to integrate their cluster-specific changes as well as the examples worked out in the previous tutorial into this Ansible repository. Finally, the students are instructed to completely re-install their master nodes, and re-provision their entire clusters with this Ansible repository. Any further changes to the system are affected through version controlled Ansible. At this stage, the teams have fully functional compute clusters managed using modern configuration management techniques.

4.4 Monitoring & Benchmarking

Chapter 9 is focused on monitoring and benchmarking tools. These are related but disparate topics. They are presented separately but combined into the same chapter for brevity and to emphasize their common use to verify the state of the cluster.

4.4.1 Chapter 9, Part 1: Monitoring. We first start with a short lecture that covers the basic terminology used in HPC monitoring such as active versus passive monitoring, out-of-band monitoring, and a summary of HPC monitoring concerns.

The practical portion of the monitoring section focuses on passive log analysis. The students work through configuring rsyslog log aggregation from the compute nodes to the master node. They are then guided through setting up and using Splunk for log analysis, including setting up Splunk alerts.

4.4.2 Chapter 9, Part 2: Benchmarking. This section of the chapter also starts with a short lecture covering common terminology around benchmarking, and the role of benchmarking in typical HPC acceptance processes. Emphasis is placed on real versus synthetic and micro versus macro benchmarks. A brief survey of these different benchmarking methods is presented, including several

industry-standard benchmarks are introduced, including HPL and various synthetic micro-benchmarks (e.g. STREAM, iozone, and the MOFED IB benchmark tools).

In the practical guide for benchmarking, several of these tools are used to gather information about the teams' clusters. This also serves as an "acceptance" stage for the teams' clusters in which they can verify that performance of the systems is as expected and comparable to the performance of other teams. The students are also instructed to run some scaling benchmarks, where a benchmark starts at a single thread, and scales until it comprises the entire system. This gives an idea of how well the system will scale with idealized parallel applications.

As a final step in the benchmarking processes, the students were provided with a ready-to-run real applications. We chose a GROMACS[7, 16] molecular dynamics simulation, as it is a well supported, open source code, and was easy to force to run for predictable time periods. The students ran this code over night. This both provided a way to perform a "real" benchmark on the clusters, as well as data for visualization exercises in the next section.

4.5 Parallel & Cluster Programming and Visualization

Chapter 10 introduces parallel programming concepts. We should note that the focus of the boot camp is not on parallel programming - LANL offers the "Parallel Computing Summer Research Internship"[12] that focuses in this area for students who would like more emphasis on programming. Instead, the objective of this brief introduction to parallel programming concepts is two fold: 1) to introduce a basic common understanding of parallel programming techniques and pitfalls as may be relevant to their research projects; 2) to introduce cluster programming and message passing concepts through a simple introduction to MPI programming. These, together, help to give a more complete understanding of how the systems the students have assembled will be used in addition to providing more hands-on experience with the high-speed networking fabric.

The chapter is introduced with a lecture covering some basic theory and terminology of parallel programming, including message passing versus shared memory models. The practical guides are divided into two. The first guide walks the students through some simple threaded programming tasks in Python. The second guide extends these concepts to span multiple nodes using the Python mpi4py module.

Both the threaded and MPI guides follow examples of developing and enhancing code that illustrates two examples that were given in Chapter 1 as illustration of why we build HPC systems as clusters. The first example illustrates a simple parallel summing algorithm. The second example is more complex and implements different versions of a 3D box of colliding particles[4].

Finally, both the 3D box example and the results of the GROMACS sample simulation provide input data for some brief visualization experiments. For the 3D box example, students are guided through making a 3D rendered movie with ParaView. For the GROMACS example, the students are guided through a 3D visualization using VMD, the molecular dynamics visualization tool.

4.6 Future Technology

Chapter 11 concludes the boot camp with a discussion of upcoming HPC technologies. In the current year, chapter introduced ideas like Linux container and cloud computing. It is anticipated that this chapter would change substantially over time to match new upcoming technology trends. Due to time constraints for the boot camp, this chapter consisted solely of a lecture. Ideally, it might include some short, practical guides for working with some of the new technologies mentioned.

5 EVALUATION

The CSCNSI is a program that we have traditionally had difficulty evaluating. Unlike some summer programs, its value isn't as much in the results of the students' final products as it is in foundation we give them for understanding the fundamentals of high–performance computing. For many students, this means that we do not have a good way to evaluate the value of the program once they have left for the summer unless we make efforts to track them down and check on their progress in school and their careers. However, as we have already mentioned, this program is an important recruiting vehicle for LANL's HPC Division, meaning that we can put at least one numeric score on the success of each year. This, in conjunction with student surveys conducted throughout the program give us some qualitative and quantitative ideas of the success of the program.

5.1 Short Term: Qualitative Evaluation

Students who attend the CSCNSI are encouraged to give feedback, and they are given frequent opportunities to do so. This year, the instructor implemented a daily "sticky note" survey: each team was given a pad of sticky notes at the start of the day, and each member was asked to briefly summarize their feelings at the end of each day. These notes were treated anonymously and were used by the instructor to tailor the pace of the class and the topics being covered each day to fit the needs of the students.

At the end of the summer, the students and the mentors were asked to fill out a longer, more formal survey about their experiences that summer. Afterward, the results of these surveys were used to evaluate how the class went, decide which students should be followed up with by our recruiting team, and begin planning for the next year.

The surveys between the summers of 2018 and 2019 were also significantly updated, so it is difficult to directly, quantitatively compare the two results. Qualitatively, however, the student evaluation of the boot camp was significantly more positive than previous years[5]. Additionally, the overall approval rating of the program improved. Given that the boot camp was the most significant change between the years, it is reasonable to assume that the overall evaluation of the program benefited strongly from the updated boot camp curriculum. The curriculum changes introduced this year

[4]Source code for the 3D box simulation, called "gas", can be found in the curriculum Supplements at [9]

[5]The authors were unable to obtain releases for the survey data, so we are only able to speak subjectively and qualitatively about that data.

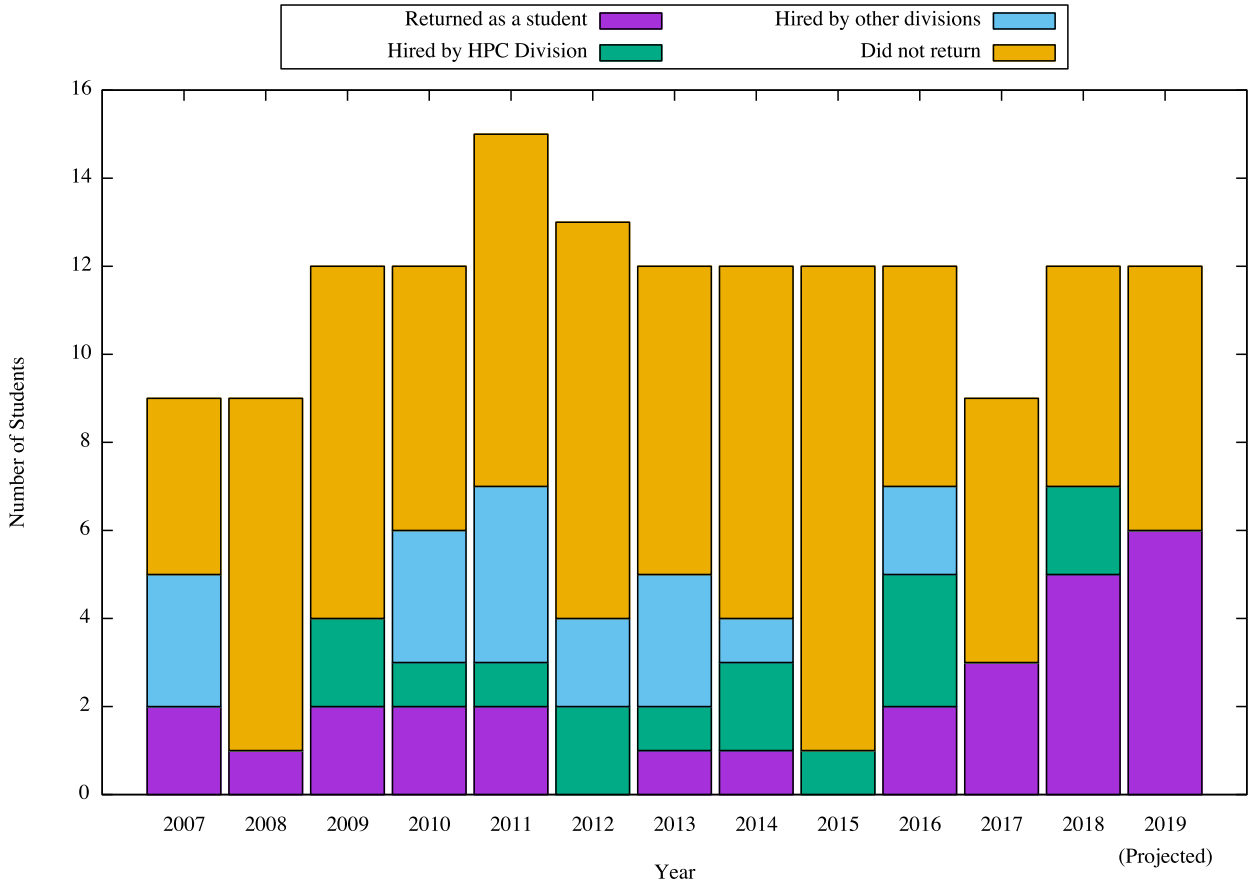

Figure 1: Student outcome statistics for CSCNSI from 2007 to present.

were significant enough that they offer us an opportunity to draw a strong distinction between the "old" and "new" curricula, which will give us a good place to do comparisons as the "new" curriculum ages and matures.

5.2 Long Term: Quantitative Evaluation

The CSCNSI has been an important recruiting tool for both HPC Division in specific and the laboratory as a whole. Since 2007, we have been maintaining records of the students who went through the program, which ones came back again as a student in our general student program, and which ones were hired as full time LANL staff in HPC Division or other divisions at the laboratory. Ignoring 2019, which is too early to count in the statistics, we have had 142 individual attendees in the program (with some years varying from the standard 12 attendees). Of these, about 15 returned to LANL as a student again, another 15 were hired as full time employees in HPC Division, and nearly 20 more were hired by other divisions at LANL. While we do not directly track statistics on students who end up at places other than LANL, we do know that several more have ended up at other laboratories and nearby businesses. The skills that the CSCSNI students learn during their summer are clearly applicable with HPC Division directly, but also with a variety of other scientific disciplines and industries. Figure 1 shows the outcomes, where known, of CSCNSI students from 2007 to present.

6 FUTURE WORK

We anticipate further developing the curriculum, the research, and the mentoring segments of the CSCNSI program going forward. Through the various sources of survey information, we will be making further minor curriculum adjustments, but overall feel that the new curriculum has provided a solid foundation to work on.

This years changes have emphasized the need for capturing better metrics on the performance of the program. Some of this may be achieved through ongoing improvements to the survey process. We are also examining the possibility of introducing entrance and exit exams to track student development.

Some students and mentors have pointed out that it would be desirable to spread out the boot camp curriculum through the program and to introduce the research components earlier. We are taking under consideration that the initial boot camp could be shortened

to include only include Chapters 1 through 8, with the remaining chapters taught in a more spread–out fashion throughout the remainder of the program.

Finally, we have opened the curriculum[9] to the broader community in hopes that it may both benefit the broader HPC educational community, as well as open a forum for community curriculum development. We have begun speaking with outside institutions that may be interested in helping to develop the curriculum for an academic course our workshop. The time frame of the CSCNSI limits the boot camp to an intensive two week period, but we believe this curriculum could be adapted and expanded to a semester course.

7 CONCLUSIONS

The CSCNSI program has a proven track record of demonstrating that a broad systems–based background in cluster computing can be a valuable background for students in a variety of HPC related fields. We have seen this qualitatively, through student and mentor surveys, and quantitatively, through the hiring pipeline it has provided. The changes to the CSCNSI program in the past year have marked a turning point for the program. We anticipate that the improved curriculum will further emphasize the benefits of a ground–up, systems based background in HPC. Though it is difficult to make broad conclusions given the limited sample size after one year, initial results are promising that this new mix of formal and informal learning will lead to an even stronger program going forward.

REFERENCES

[1] 2019. ACM Richard Tapia Celebration of Diversity in Computing. (2019). Retrieved Sep 23, 2019 from http://tapiaconference.org/

[2] 2019. Ansible is a simple IT automation tool. (2019). Retrieved Sep 24, 2019 from https://www.ansible.com/

[3] 2019. CentOS Project. (2019). Retrieved Sep 24, 2019 from https://www.centos.org

[4] 2019. Cluster System, Cluster and Networking Summer Institute (CSCNSI). (2019). Retrieved Jul 30, 2019 from https://clustercomputing.lanl.gov

[5] 2019. CSCNSI: Past projects. (2019). https://www.lanl.gov/projects/national-security-education-center/information-science-technology/summer-schools/cscnsi/student-projects.php

[6] 2019. Grace Hopper Celebration. (2019). Retrieved Sep 23, 2019 from https://ghc.anitab.org

[7] 2019. GROMACS. (2019). http://gromacs.org

[8] 2019. International Conference for High Performance Computing, Networking, Storage, and Analysis. (2019). Retrieved Sep 23, 2019 from http://supercomputing.org/

[9] 2019. LANL Supercomputing Institute Curriculum. (2019). Retrieved Sep 24, 2019 from https://github.com/hpc/cluster-school

[10] 2019. Los Alamos National Laboratory, HPC Division. (2019). Retrieved Jul 30, 2019 from https://hpc.lanl.gov

[11] 2019. OpenHPC. (2019). Retrieved Sep 24, 2019 from https://openhpc.community

[12] 2019. Parallel Computing Summer Research Internship. (2019). https://www.lanl.gov/projects/national-security-education-center/information-science-technology/summer-schools/parallelcomputing/index.php

[13] 2019. tmux project. (2019). Retrieved Sep 24, 2019 from https://github.com/tmux/tmux

[14] 2019. Warewulf cluster provisioning. (2019). Retrieved Sep 24, 2019 from https://github.com/warewulf/warewulf3

[15] Prentice Bisbal. 2019. Training Computational Scientists to Build and Package Open-Source Software. *Journal of Computational Science Education* 10, 1 (2019), 74–80.

[16] R. van Drunen H.J.C Berendsen, D. van der Spoel. 1995. GROMACS: A message-passing parallel molecular dynamics implementation. *Computer Physics Communications* 91, 1-3 (September 1995), 43–56.

[17] David Brayford et al. Karl W. Schulz, C. Reese Baird. 2016. Cluster Computing with OpenHPC. *SC16: HPCSYSPROS Workshop* (2016).

[18] Kai Himstedt Nathanael Hübbe Sandra Schröer Michael Kuhn Matthias Riebisch Stephan Olbrich Thomas Ludwig Jean-Thomas Acquaviva Anja Gerbes Lev Lafayette Weronika Filinger, Julian Kunkel and Hinnerk Stüben. 2019. Towards an HPC Certification Program. *Journal of Computational Science Education* 10, 1 (2019), 88–89.