

Teaching HPC Systems Administrators

Alex Younts
ayounts@purdue.edu
Purdue University

Stephen Lien Harrell
sharrell@purdue.edu
Purdue University

ABSTRACT

The ability to grow and teach systems professionals relies on having the capacity to let students interact with supercomputers at levels not given to normal users. In this paper we describe the teaching methods and hardware platforms used by Purdue Research Computing to train undergraduates for HPC systems-facing roles. From Raspberry Pi clusters to the LittleFe project, previous work has focused on providing miniature hardware platforms and developing curriculums for teaching. Recently, we have developed and employed a method using virtual machines to reach a wider audience, created best practices, and removed barriers for approaching coursework. This paper outlines the system we have designed, expands on the benefits and drawbacks over hardware systems, and discusses the failures and successes we have had teaching HPC System Administrators.

KEYWORDS

hpc, syspros, systems professionals

1 INTRODUCTION

As leadership computing facilities draw closer to exascale and academic research computing centers mature around the world, the need for competent HPC System Administrators is increasing. Similarly, the complexity of HPC systems is increasing with the slowing of the Moore's Law trend and node heterogeneity becoming all but a necessity. Gone are the days when commodity hardware connected with some cheap Ethernet switches were a viable solution to solving the world's science problems. Today, system administrators need to tackle accelerators, big data technologies, AI and ML frameworks, ever changing network fabrics, and a quickly changing ecosystem of core architectures. In the same way that this complexity has increased, HPC system administration training approaches must also mature in complexity and scope.

1.1 Roles of HPC System Administrators

Although system administration, as a professional practice, is well established, HPC adds a layer of complexity that requires its own community, documentation, and training. While operating system skills are the same, the "High Performance" in HPC requires understanding of CPU architecture, exotic networks, computer architecture, and parallel technologies in a way that is foreign to most system administrators. As a background for teaching HPC

System Administrators, it is important to set a baseline of roles and responsibilities needed to run an HPC machine.

There are four major roles that encompass the operation of HPC machines (clusters):

- Network Administration
- Parallel Storage Administration
- Data Center and Hardware Administration
- System Administration and Automation

Additionally, these roles often have different responsibilities within an HPC context than they would in a typical industry role. For instance, a traditional network system administrator may need to know *a)* routing protocols, *b)* network hardware administration, and *c)* TCP/IP, whereas an HPC network administrator would be expected to not only know these topics, but be expected to understand low latency interconnects, such as RDMA networks, as well as understanding how the network layer can impact parallel jobs running over it. Similarly, while storage administrators may need to know *a)* NFS, *b)* CIFS, and *c)* storage appliances in a traditional industry role, within HPC they would be required to know parallel file systems (Lustre, GPFS), file systems over RDMA and even tape archival systems that often play a vital role in research-focused computing. Data center focused administrators for large system installations largely have the same concerns as their HPC-focused colleagues. Lastly, systems administrators, who are responsible for the OS, create and submit hosts and nodes of a cluster, set up the scheduler, the general deployment of the entire cluster, and the ongoing configuration management, are expected to understand a completely different technology stack. While each of these roles requires considerable expertise, many time systems facing staff are expected to be experts in a few and knowledgeable in all of these roles. The more roles staff are knowledgeable about, the better understanding of each component (systems, network, storage) and how each component interacts with the whole system. This knowledge is crucial to understanding failures and tuning the system to be truly High Performance.

2 HPC SYSTEMS TRAINING APPROACHES

The traditional method for teaching system administration is the apprentice model [14]. Within this model, an "expert" slowly feeds tasks of increasing difficulty to the "apprentice", while at the same time being a resource for topics intrinsic to a specific task, but also as a guide to the self-learning process. While this is an important and time-tested method for training system administrators of every variety, this approach does not scale and is extremely high touch. It requires an "expert" to have plenty of time and the right demeanor for the apprenticeship to create a competent HPC System Administrator.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright ©JOCSE, a supported publication of the Shodor Education Foundation Inc.

2.1 Workshops

Currently there are two workshops that include HPC System Administration training and tutorials. The first and oldest, Linux Clusters Institute (LCI), "provides education and advanced technical training for IT professionals who deploy and support High Performance Computing (HPC) Linux clusters" [4]. LCI provides two separate workshops, one for Linux novices and one for Linux System Administrators that are trying to learn HPC. The second workshop, The TACC Institute Series Immersive Training in Advanced Computation: Designing and Administering Large-scale Systems, also provides a week-long workshop where students are "provisioning nodes, installing and configuring resource managers, maintaining a sane user environment, and addressing security concerns" [6]. Both of these workshop provide hands-on activities as well as lectures that provide context for the training.

2.2 Student Cluster Competition

The Student Cluster Competition (SCC) [9] is described as "a microcosm of a modern HPC center that teaches and inspires students to pursue careers in the field. It demonstrates the breadth of skills, technologies and science that it takes to build, maintain and utilize a supercomputer." [18]. This event, while not solely focused on HPC Systems Administration, includes opportunities for undergraduates to learn and practice HPC System Administration in the heart of the competition.

2.3 Undergraduate Training at Purdue

At Purdue University, as well as other institutions, the HPC systems staff provides job opportunities for undergraduates, as well as HPC and clustering classes on campus. The material for the classes grew out of the Purdue SCC program and have evolved into their own topics over the years.

3 PURDUE HPC SYSTEMS TRAINING THROUGH THE YEARS

Purdue University's Research Computing center started hiring undergraduates to do HPC systems work in 2003. This was the beginning of our HPC systems administrator training which primarily used an apprentice model. Starting in 2007, Systems staff mentored students in the Student Cluster Competition series. Since then, staff have been iterating on how to best teach HPC concepts and system administration to undergraduates and have tried many technology platforms to provide consistent, affordable, and reliable platforms for teaching HPC.

3.1 Early Years

In 2007 and the following few years, training initiatives at Purdue were based around the Student Cluster Competition. [10] HPC systems staff partnered with faculty on campus to provide overviews of parallelism, however these early years of the competition were heavily geared toward HPC system administration, more so than later years. The classes were primarily an open lab format which could be categorized as a distributed apprentice model.

3.1.1 Training Platform: SCC Competition Hardware. The hardware platforms chosen for these competitions varied from year to

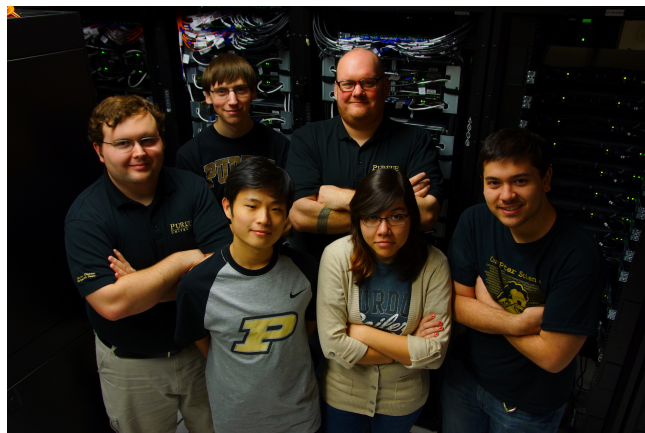


Figure 1: Purdue Student Cluster Competition Team

year, and as the years progressed, the clusters became more complex. In 2007, the student cluster was architected after a traditional Beowulf cluster with simple servers and Ethernet networking. A year later, the Purdue team took an experimental SiCortex many-core cluster composed of hundreds of MIPS cores and a custom interconnect [2]. Later, the cluster concept turned from homogeneous compute nodes to hybrid nodes prominently featuring GPU's for compute acceleration.

What remained the same throughout the years of building the SCC clusters was that the hardware was always a short term loan from a sponsoring vendor. Students, especially those handling the team's system administrator needs, were always presented with the most recently released hardware and the time challenge of preparing it for the competition.

3.1.2 Outcomes and Lessons Learned. While being a naive implementation, this first method of teaching undergraduates HPC systems-facing topics was somewhat effective and an important first step. Those first classes, which were approximately ten students each, were responsible for three students becoming HPC system administrators and are currently working in the HPC community today. Additionally, students were introduced to the academic writing process and published two experience papers on the subject [12] [22].

We found that not having formal classes for something like the SCC was a detriment to student participation. Student attendance and commitment were sometimes low. Additionally, giving students complex HPC-centric hardware right away created a very steep ramp for students to overcome.

3.2 Formalized Classes

In 2011, to combat low participation in the SCC meetings and inconsistency in training, classes were formalized beyond an open lab. While the classes were still centered around the SCC, we worked to create a curriculum that could be reused and contained important parallel computing and HPC systems topics.

3.2.1 Assignments.

- (1) Introduction to HPC and the SCC

- (2) Usage of Supercomputers (login and compile HPC)
- (3) Introduction to Computer and HPC Architecture
- (4) Linux Installation, Daemons, Configuration Files and Basic Networking
- (5) Batch Schedulers, Advanced SSH
- (6) Compiling with MPI and OpenMP
- (7) Strong and Weak Scaling Studies and Bottleneck Identification
- (8) Specific SCC Applications and Strategy for the Rest of the Course

3.2.2 Training Platform: Recycled Desktops. Our first attempts at providing a dedicated training platform over short term vendor loans was to re-purpose desktops [13] after their life in the student computer labs ended. Every student was given 4 desktops, cables, and an Ethernet switch. This was a fairly adequate solution when new parts were available yearly, but as budgets and technology have changed, computer labs changed their refresh cycles and are no longer on a predictable schedule so this effort was sustainable long term.

3.2.3 Outcomes and Lessons Learned. This era of classes was a marked improvement over previous years. Although the classes still included practical labs around the SCC, the instructors touched on more “theory” topics than before such as computer architecture.

Additionally as the desktops aged, replacement parts were necessary, especially after inexperienced students performed repairs. Plus, full sized desktops take up a lot of floor space. During an inspection of the space, a glib manager was heard saying “It would be terrible if a student got trapped under an avalanche of chalk-dust encrusted desktops.”

3.3 Bare Hardware to Computational Visualization

In 2013, we went back to an open lab that revolved around the SCC and began a separate class that ran alongside the SCC open lab. This was open to anyone that was interested in HPC, not just SCC participants. It was also the time that we started to focus on inspiring undergraduates as well as teaching. A breadth-first approach was taken with the idea of having a final project where students could see the fruits of their labors in an accessible way. We chose to run weather code to forecast as weather maps from a visualization of a forecast is a common and accessible experience for almost everyone, regardless of background. [8]

3.3.1 Assignments.

- (1) Introduction to HPC
- (2) Basic Linux Installation
- (3) Automating Installs
- (4) Hardware Setup and Install [lab]
- (5) Schedulers
- (6) Interconnects and Storage
- (7) DHCP and DNS for clusters [lab]
- (8) Shared Storage for Clusters [lab]
- (9) Scheduler Setup [lab]
- (10) Installing MPICH
- (11) Installing WRF
- (12) Troubleshooting WRF and MPI

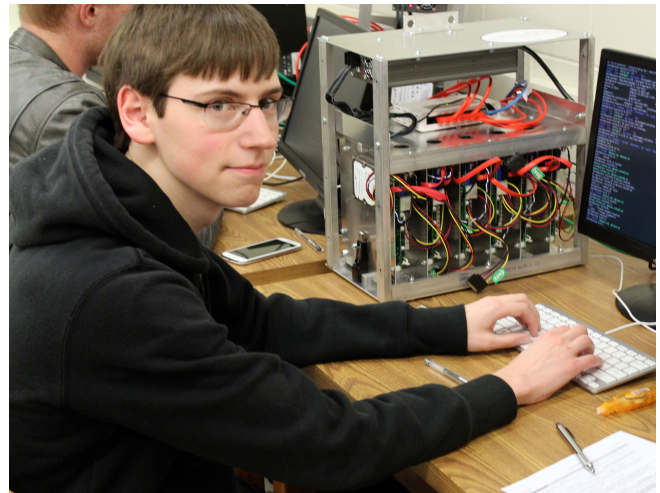


Figure 2: Student with LittleFE cluster.



Figure 3: LittleFE Computer Lab.

- (13) Introduction to Python
- (14) Visualizing wrfout
- (15) Automating WRF Runs with Python

3.3.2 Training Platform: LittleFE. The “little iron” project brings together curriculum and a hardware plan that many schools have implemented to teach students high performance computing [16]. The platform itself is made of bare motherboards and metal rods to act as a mounting platform. A student or a small team of students get their own cluster. Each cluster costs a reasonable amount for the number of computers involved and students can get the visceral, hands on experience putting together their machine [17]. Also, it is possible to get low-end consumer video hardware build onto the node motherboards for lightweight accelerator work as well. While initial investment per seat is manageable, and cheaper than a half-rack of real cluster compute nodes, the costs are not insignificant.

After outgrowing the “pile of desktops” solution, we sought an internal grant and built 15 LittleFE clusters to continue as our teaching platform. Initially, it seemed like we could procure each seat for a reasonable sum of money, but after the build was over

and all receipts were collected, the price had risen to approximately 20% due to needing additional pieces. This solution provided many benefits that we were looking for but hit a wall when we needed to scale the number of students we reached. Not only did the per-seat cost end up being more expensive, but the desk space and off-semester storage of the hardware became a logistical problem.

3.3.3 Lessons Learned. The breadth approach, while well intentioned, was too much to include in one semester. Students were often faced with hardware or OS problems when they were supposed to be running WRF or writing python to visualize. Although this method may work if the topic list is paired down, one must be careful not to overwhelm the students. In the end, some students did not have the time to complete the visualizations, thus negating the point of this method.

The LittleFe hardware suffered a number of growing pains as we progressed through the build and the course. The materials were difficult to acquire through the University procurement system, a web tool designed to buy complete computers and not just a stack of parts. The mounting hardware shined a light on both cooling problems and static electricity issues, both causing general instability for students. Given the overall mounting solution, the clusters were essentially immobile and required us to open additional lab periods so students could complete their work. Although this hardware was not well suited for this task, it has been used quite successfully for single day hack-a-thons and is sufficiently stable for that use case. If we had the resources to develop a second generation of LittleFe instruments, we believe many of these issues could be mitigated.

4 CURRENT EFFORTS

As an amalgamation of our previous experiences, the courses that are instructed today [11] have two tracks. First a scientific computing track, which provides students with some basic Linux skills but focuses on running and visualizing scientific codes. Then an HPC Systems Track, which truly focus on important aspects of building systems. This was a hybrid approach of inspiring undergraduates but still focusing where the students interests lie in order to reduce the amount of topics from our previous efforts.

4.1 HPC Systems Track

The HPC Systems Track was integrated into the new course curriculum as a way to engage a more diverse set of students. Students in the sciences had a firm footing for many of the course activities, but students from the Polytechnic school and Computer Science and Engineering majors were provided this path to understand the technical work behind the scenes of supercomputing. The course was broken into three modules and this track was offered as an alternative to the second module. The first module covered introductory materials and labs and the third module was a crash course in simulating fluid dynamics problems using OpenFOAM.

The System Track included practical activities in the data center to work on the University's real resources but focused on providing a hands-on-keyboard experience to learn the guts of HPC clusters. The goal was for students to be able to explore a working system and replicate it themselves without copying the example. Each assignment had a final stretch goal that allowed us to judge whether

students were simply copying configuration files around or actually exploring and learning the material.

4.1.1 Assignments.

- (1) Introduction to HPC
- (2) Tutorial on the Advanced Linux Shell [lab]
- (3) Presentation on Cluster Architecture
- (4) Data Center Tour and Hands-on Lab
- (5) Basic Linux Virtual Machine Installation [lab]
- (6) Master Node xCAT Installation [lab]
- (7) Building Compute Node Images [lab]
- (8) Installation of Slurm [lab]
- (9) Running Sample Jobs [lab]

4.2 Virtual Labs

As we expanded our scale to dozens of students per semester and planned for even wider reach, it became clear the monetary investments in physical infrastructure and the time investment getting low grade hardware to cooperate were detracting from reaching our goals. We evaluated several commercial cloud-based offerings to host the lab environment but the options seemed geared towards traditional client-server IT teaching. We also researched using infrastructure as code tools, like Terraform [1], to automate lab environments in AWS, but found the variable costs very difficult to quantify and potentially quite large. We needed a new way forward that fit with the campus's available cyberinfrastructure and could be delivered remotely. We came up with the virtual HPC lab concept.

4.2.1 Implementation. The basics of the method was to enable our Scholar cluster [5], which is a supercomputer dedicated to computational research teaching, to run scripted virtualized clusters for students. We required the solution involve no privileged system access (e.g. sudo access) or access to the underlying network infrastructure [20] (e.g. Linux Ethernet bridges). We used the popular QEMU system emulator along with the Virtual Distributed Ethernet (VDE) userland networking stack [7].

Students had the ability to launch a script that brought up their virtual lab through our ThinLinc remote desktop in a web browser and get an empty, semi-configured, or completed configured cluster environment [21]. The script, just a bash script run by students at a terminal, lets students choose the lab to launch, handles creating copy on write snapshots from golden image masters, lets students continue progressing on current labs, and the ability to reset a lab back to a checkpoint if something goes wrong.

The lab environment spawns several windows, each representing the QEMU console to a running virtual machine or console access to a VDE network instance. Students are able to adjust the running parameters of the QEMU instances (e.g. inserting a boot disk) and, with some limitations, have essentially identical access to the lab as if it was running on real hardware.

4.2.2 Lessons Learned. Using previous courses as comparison, the first readily apparent success of the virtual lab concept was that students were learning valuable HPC skills in the first lab. Student frustration was also significantly down, as rolling back to a working check point or starting over did not take an hour waiting for the RedHat installer to run. Students also appreciated the ability to

work on the labs and assignments outside of class since the Scholar environment is available remotely any time of the day.

The class sessions themselves progressed fairly normally. The lecturer presented context and background information at the start of class. They could demonstrate any tips or tricks to the class using a snapshot of the environment at the same stage of progress as the students' copies. Teaching assistants were available to assist individual students on their laptops as problems arose or remotely by using ThinLinc session sharing.

While the experience was overall very positive, one speed bump did sneak up. Scholar, built as a platform for teaching science, had a scratch file system primarily tuned for standard HPC workloads. Having numerous QEMU virtual machines running with their disk images doing random small block I/O did take a toll on overall system responsiveness. The scratch file system was based on the ZFS file system and adding a small quantity of SSD disks resolved the issue.

4.3 Future work

After seeing a pair of courses successfully run using the virtual HPC lab environment, we are encouraged that our goal to some day offer our courses widely is possible. As we move forward to publish our curriculum and environments, we hope to build momentum to provide the academic HPC community the skill sets that are desperately required.

5 OTHER TRAINING PLATFORMS

Although these platforms have not been used in any HPC Systems Administration classes at Purdue, they have been evaluated and may fit the needs of others depending on the availability of resources and time.

5.1 Raspberry Pi Clusters

A small stack of Raspberry Pi's are all the rage across the Internet. From business [15] to education to Department of Energy labs [19], everyone seems to be building tiny clusters [3]. These clusters do everything from compiling and testing pipelines, to simple MPI scalability testing, and to running production workloads through Kubernetes.

Aside from reliability issues with early Raspberry Pi's and clone boards, the best part about a stack of single board computers is the cost. A student can readily be provided a cluster for under \$500. However, we believe that many of the drawbacks found in using dedicated hardware are still present in a cluster of Pi's.

5.2 Cloud Environments

Commercial cloud providers are a potential avenue to explore in greater depth. The core technology requirements *a)* isolated network segments, *b)* snapshots and rollback of instances, and *c)* the ability for remote student assistance already exist as features on all the various providers' platforms. Additionally, while the scripts written for Scholar could be portable to other institutions and system resources, some effort will be required. We hope to keep that effort to a bare minimum as we move towards publishing our work further, but we acknowledge the strong advantages of a universal platform with consistent lab materials and curriculum.

The largest drawbacks at the present time to the cloud are a lack of a cohesive interface for students and the variable costs an institution will incur depending on student usage of the environment.

6 CONCLUSIONS

Purdue Research Computing's training methods for HPC System Administrators and the hardware platforms have supported these efforts. We have found that running our own virtualized environment for teaching to be effective to meet our goals of low cost, low overhead, and low student frustrations. Additionally, we have split our class to have two separate tracks to focus on the HPC Systems topics while still maintaining our goal of inspiring undergraduates.

ACKNOWLEDGMENTS

We thank our faculty partners, Dr. Evans, Dr. Baldwin and Dr. Ward, and all our colleagues and management who made this work possible. We'd also like to thank Christopher Phillips for his keen editorial skills.

REFERENCES

- [1] 2020. Terraform. <https://www.terraform.io> [Online; accessed 17. Jan. 2020].
- [2] 2020. The SiCortex SC series | TOP500 Supercomputing Sites. https://web.archive.org/web/20090531211623/http://www.top500.org/2007_overview_recent_supercomputers/sicortex_sc_series [Online; accessed 17. Jan. 2020].
- [3] Joel C Adams, Jacob Caswell, Suzanne J Matthews, Charles Peck, Elizabeth Shoop, and David Toth. 2015. Budget Beowulfs: A showcase of inexpensive clusters for teaching PDC. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. ACM, 344–345.
- [4] David Akin, Mehmet Belgin, Timothy A. Bouvet, Neil C. Bright, Stephen Lien Harrell, Brian Haymore, Michael Jennings, Rich Knepper, Daniel LaPine, Fang Cherry Liu, Amiya Maji, Henry Neeman, Resa Reynolds, Andrew H. Sherman, Michael Showerman, Jenett Tillotson, John Towns, George Turner, and Brett Zimmerman. 2017. Linux Clusters Institute Workshops: Building the HPC and Research Computing Systems Professionals Workforce. In *Proceedings of the HPC Systems Professionals Workshop (HPCSYSPROS'17)*. ACM, New York, NY, USA, Article 4, 8 pages. <https://doi.org/10.1145/3155105.3155108>
- [5] M. E. Baldwin, X. Zhu, P. M. Smith, S. L. Harrell, R. Skeel, and A. Maji. 2016. Scholar: A Campus HPC Resource to Enable Computational Literacy. In *2016 Workshop on Education for High-Performance Computing (EduHPC)*. 25–31. <https://doi.org/10.1109/EduHPC.2016.009>
- [6] Texas Advanced Computing Center. 2019. TACC Institute Series Immersive Training in Advanced Computation. <https://www.tacc.utexas.edu/education/institutes/designing-and-administering-large-scale-systems>
- [7] Renzo Davoli. 2005. Vde: Virtual distributed ethernet. In *First International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*. IEEE, 213–220.
- [8] Stephen Lien Harrell, Benjamin J Cotton, Michael E Baldwin, and Andrew L Howard. 2013. Developing a Scientific Computing Cluster Course for the Undergraduate Curriculum. (2013).
- [9] Stephen Lien Harrell, Hai Ah Nam, Verónica G. Vergara Larrea, Kurt Keville, and Dan Kamalic. 2015. Student Cluster Competition: A Multi-disciplinary Undergraduate HPC Educational Tool. In *Proceedings of the Workshop on Education for High-Performance Computing (EduHPC '15)*. ACM, New York, NY, USA, Article 4, 8 pages. <https://doi.org/10.1145/2831425.2831428>
- [10] Stephen Lien Harrell, Preston M. Smith, Doug Smith, Torsten Hoefler, Anna A. Labutina, and Trinity Overmyer. 2011. Methods of Creating Student Cluster Competition Teams. In *Proceedings of the 2011 TeraGrid Conference: Extreme Digital Discovery (TG '11)*. ACM, New York, NY, USA, Article 50, 6 pages. <https://doi.org/10.1145/2016741.2016795>
- [11] Elizabeth Hillery, Mark Daniel Ward, Jenna Rickus, Alex Younts, Preston Smith, and Eric Adams. 2019. Undergraduate Data Science and Diversity at Purdue University. In *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning) (PEARC '19)*. ACM, New York, NY, USA, Article 88, 6 pages. <https://doi.org/10.1145/3332186.3332202>
- [12] Andrew Howard, Alex Younts, Preston M. Smith, and Jeffery J. Evans. 2008. Undergraduate experience in clustering at the SC07 Cluster Challenge. In *In Proceedings of the 2008 Linux Clusters Institute*.

- [13] Jason St John and Thomas J Hacker. 2017. A Small-Scale Testbed for Large-Scale Reliable Computing. In *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 1251–1258.
- [14] David Jones. 2018. How do you teach Systems Administration?. In *SAGE-AU*.
- [15] Patrick Kennedy. [n. d.]. Oracle Shows 1060 Raspberry Pi Supercomputer at Oracle OpenWorld 2019. <https://www.servethehome.com/oracle-shows-1060-raspberry-pi-supercomputer-at-oow/>
- [16] Mobeen Ludin, Aaron Weeden, Jennifer Houchins, Skylar Thompson, Charles Peck, Ivan Babic, Kristin Muterspaw, and Elena Sergienko. 2013. LittleFe: The high performance computing education appliance. In *2013 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 1–1.
- [17] Charles Peck. 2010. LittleFe: parallel and distributed computing education on the move. *Journal of Computing Sciences in Colleges* 26, 1 (2010), 16–22.
- [18] SC17. 2017. Student Cluster Competition. <https://sc17.supercomputing.org/studentssc/student-cluster-competition/index.html>
- [19] Adam Simpson, Anthony DiGirolamo, and Robert D. French. [n. d.]. Tiny Titan from Oak Ridge Leadership Computing Facility. <https://tinytitan.github.io>
- [20] Julian Stecklina. [n. d.]. A Userspace Packet Switch for Virtual Machines. ([n. d.]).
- [21] Abhinav Thota, Le Mai Weakley, Ben Fulton, HE Dennis, Laura Huber, Scott Michael, Winona Snapp-Childs, Stephen Lien Harrell, Alexander Younts, Daniel T Dietz, et al. 2019. Research Computing Desktops: Demystifying research computing for non-Linux users. In *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning)*. ACM, 54.
- [22] Alex Younts, Andrew Howard, Preston M. Smith, and Jeffrey J. Evans. 2009. Bringing disruptive technology to competition. In *In Proceedings of the 10 th LCI International Conference on High-Performance Clustered Computing*.