Using CloudLab as a Scalable Platform for Teaching Cluster Computing

Linh B. Ngo
West Chester University
West Chester, Pennsylvania
lngo@wcupa.edu

Jeff Denton
Clemson University
Clemson, South Carolina
denton@clemson.edu

ABSTRACT

A significant challenge in teaching cluster computing, an advanced topic in the parallel and distributed computing body of knowledge, is to provide students with an adequate environment where they can become familiar with real-world infrastructures that embody the conceptual principles taught in lectures. In this paper, we describe our experience setting up such an environment by leveraging CloudLab, a national experimentation platform for advanced computing research. We explored two approaches in using CloudLab to teach advanced concepts in cluster computing: direct deployment of virtual machines (VMs) on bare-metal nodes and indirect deployment of VMs inside a CloudLab-based cloud.

CCS CONCEPTS

• **Applied computing → Education**; • **Computer systems organization → Cloud computing**; • **Software and its engineering → Distributed systems organizing principles**;

KEYWORDS

experimental platform, distributed computing, hands-on learning

1 INTRODUCTION

Within a parallel and distributed computing undergraduate curriculum, advanced topics are defined as those carrying significant current or emerging interests [15]. Among these topics, cluster computing remains an advanced yet fundamental subject that provides the background information for other topics such as cloud/grid computing and big data computing. Since the early adaptation of cluster computing into the undergraduate curriculum [2], a large body of materials has been developed to support in-class lectures. The challenge is providing a hands-on environment for students to connect these materials to real-world technical problems and skill sets.

To address this challenge, institutions have turned to virtual technology as an economical and scalable platform. Previous work has shown the applicability of virtual machines in teaching computer networking, operating systems, security, and databases [4, 19]. More recent work has seen virtual machines used in courses for big data [7] and cloud computing [8]. Even with virtual technology, many technical hurdles remain. Most notable is the necessary investment of time, money, and effort from both the instructors and the institutions into setting up and maintaining the virtual machine images and hardware infrastructure as well as the accompanying lecture materials.

In this paper, we present our approach in leveraging CloudLab [16], a publicly available computing resource as a platform to develop and host materials to support teaching topics in cluster computing. Using CloudLab, we demonstrate how instructors can develop scalable, maintainable, and shareable contents that minimize technical hurdles while still exposing students to critical concepts in cluster computing. The remainder of this paper is organized as follows. Section 2 provides an overview about the CloudLab platform. Next, we describes in details our gradual integration of CloudLab into materials for a distributed and cluster computing course over several semesters in Section 3. Lessons learned are discussed in Section 4. We examine related work in Section 5, including those that leverage CloudLab in general computer science education and those that leverage other resources besides CloudLab specifically for high performance computing education. Section 6 concludes the paper and discusses future work.

2 CLOUDLAB

Funded by the National Science Foundation in 2014, CloudLab has been built on the successes of the Global Environment for Network Innovations (GENI) [3] in order to provide researchers with a robust cloud-based environment for next generation computing research [16]. These resources are distributed across several U.S. institutions. As of Summer 2018, CloudLab boasts an impressive collection of hardware. At the Utah site, there is a total of 785 nodes, including 315 with ARMv8, 270 with Intel Xeon-D, and 200 with Intel Broadwell. The compute nodes at Wisconsin include 270 Intel Haswell nodes with memory ranging between 120GB and 160GB and 260 Intel Skylake nodes with memory ranging between 128GB and 192GB. At Clemson University, there are 100 nodes running Intel Ivy Bridges, 88 nodes running Intel Haswell, and 72 nodes running Intel Skylake. All of Clemson's compute nodes have large memory (between 256GB and 384GB), and there are also two additional storage-intensive nodes that have a total of 270TB of storage available. CloudLab is currently expanding under a follow award from the National Science Foundation which will include more network interfaces, new CPU architectures, and the ability to interface with other cloud services including Amazon Web Services and the Massachusetts Open Cloud [17].

In order to provision resources using CloudLab, a researcher needs to describe the necessary computers, network topologies, startup commands, and how they all fit together in a resource

https://doi.org/10.22369/issn.2153-4136/10/1/17

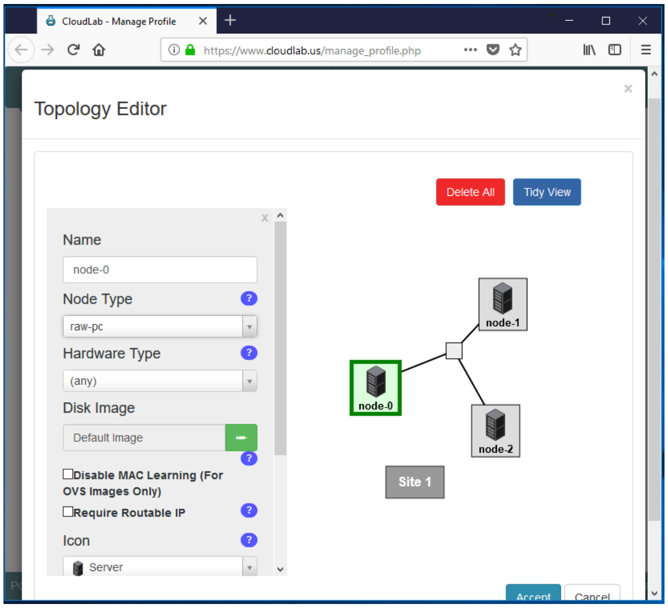

Figure 1: CloudLab's GUI for designing experimental profiles

description document. CloudLab provides a graphical interface, as shown in Figure 1, inside a web browser that allows users to visually design this document through drag-and-drop actions. For large and complex profiles, this document can also be automatically generated via Python in a programmatic manner as demonstrated in Listing 1. Starting Fall 2017, CloudLab supports a direct integration between publicly readable git repositories and their profile storage infrastructure. This significantly minimizes the effort needed to modify existing profile while still maintaining a complete history of previous changes.

Listing 1: A CloudLab profile written in Python to describe a 6-node experimental profile

```
import geni.portal as portal
import geni.rspec.pg as pg
import geni.rspec.igext as IG

pc = portal.Context()
request = pc.makeRequestRSpec()

link = request.LAN("lan")
for i in range(6):
  if i == 0:
    node = request.XenVM("head")
    node.routable_control_ip = "true"
  elif i == 1:
    node = request.XenVM("metadata")
  elif i == 2:
    node = request.XenVM("storage")
  else:
    node = request.XenVM("compute-" + str(i))
    node.cores = 2
    node.ram = 4096

  node.disk_image = "urn:publicid:IDN+emulab.net+
      ↪ image+emulab-ops:CENTOS7-64-STD"

  iface = node.addInterface("if" + str(i-3))
  iface.component_id = "eth1"
  iface.addAddress(pg.IPv4Address("192.168.1." + str(
      ↪ i + 1), "255.255.255.0"))
  link.addInterface(iface)
pc.printRequestRSpec(request)
```

The resource description document provides the blueprints for CloudLab to provision resources and instantiate the experiments. Once the resources are allocated and images for the computing components are booted on top of bare metal infrastructure, CloudLab users are granted complete administrative privilege over the provisioned infrastructure. Like XSEDE, CloudLab allows instructors to apply for educational projects and to add students to these projects.

3 USING CLOUDLAB TO TEACH CLUSTER COMPUTING CONCEPTS

At Clemson University, the distributed and cluster computing course is intended to present students with a broad overview of key components and concepts in distributed and cluster computing. The topics covered include the Beowulf model of networked computers, distributed file systems, the message-passing programming paradigm, scheduling on a cluster of computers, big data and data-intensive computing, and the map-reduce programming paradigm. The goal of this course is to provide students with the fundamental concepts in distributed and cluster computing and hands-on exposure to latest real world technologies and platforms built and expanded on these concepts.

3.1 Course History

The distributed and cluster computing course is at junior level, but most students in the class wait until the first or second semester of their senior year before registering. Typically, the enrollment ranges between 35 and 40 students. With the availability of Clemson University's centralized supercomputer with more than 2000 compute nodes, early offerings of the course in 2012 through 2014 had focused primarily on the large-scale programming aspects for both high performance and big data/data-intensive computing. This includes MPI-based concepts such as pleasantly parallel, divide-and-conquer, and synchronous computing and the MapReduce programming paradigms for big data. While the course covered distributed infrastructure knowledge such as the Beowulf cluster architecture, parallel and distributed file systems, and the Hadoop Big Data infrastructure, students' understanding of these concepts were assessed only through in-class examination. All take-home assignments were designed to be programming-based.

Over time, we had begun to realize several short-comings of the course regarding its practicality for students. There exists other courses that focus primarily on parallel programming in MPI and

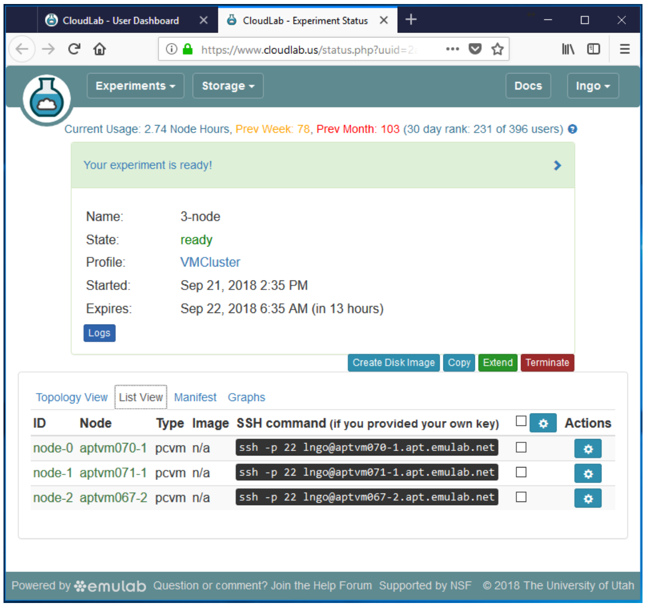

Figure 2: A 3-node cluster using XenVM on CloudLab for Assignment 2, Fall 2015

CUDA that can provide students with a more in-depth programming knowledge. While students can pick up some Linux skills by working with the supercomputer for programming assignments, these interactions are limited in user space and are barely on-par, if not less than, the amount of knowledge gained from a Linux administration course. To improve the course, we need to find a way to enable students to gain hands-on experience for the core system concepts regarding distributed infrastructures. Prior to CloudLab, several approaches had been explored including accessing virtual machines and creating a special dedicated cluster from a subset of the supercomputer. While they provided students with more access to the system aspects of distributed infrastructure, significant technical overhead remains [13].

3.2 First CloudLab Interaction: Fall 2015

CloudLab was first incorporated into the course's materials in Fall 2015, immediately after it became publicly available. For lecture materials, CloudLab was first used as a direct example of how a network of computers can be deployed such that it presents a transparent and unified interface to the users, yet is consisted of compute nodes placed at geographically distributed locations. This is to emphasize the development of distributed computing infrastructures, starting from on-site computer clusters, moving to grid computing, and then to cloud computing.

CloudLab was also incorporated in the second individual assignment whose purpose is to help students to become more familiar with working in distributed environments and moving between computing sites through a command-line terminal. The assignment was divided into two parts. In the first part, students learned how to log into the local supercomputer, write a job script, and submit this job to the scheduler for execution. The job requests three nodes

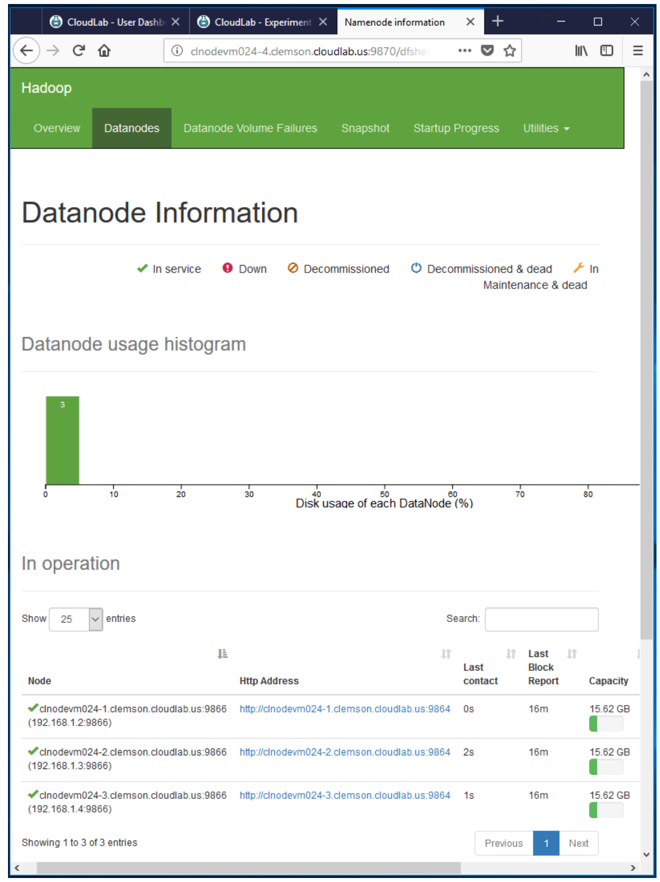

Figure 3: Public Web-UI of the Hadoop cluster deployed inside CloudLab

from the supercomputer and then runs the *pbsdsh* command which would in turn execute the *hostname* command on each of these nodes and return the domain names. In the second part, students first deployed a 3-node cluster on CloudLab and acquired their IP addresses manually. Next, they modify their job script from part 1 so that each local compute node would contact a CloudLab node via SSH and request the CloudLab node's host name. The deployment and instruction to access these nodes are displayed on CloudLab's web interface as shown in Figure 2.

Later in the semester, CloudLab was utilized once again as part of a hands-on learning lecture about Hadoop Distributed File System [18]. After students learned about the principles behind Hadoop's architectural design and implementation, they spent one class to practice setting up and deploying a 3-node Hadoop cluster on CloudLab and also observing how data placement and data movement processes happened. This process is performed as a team exercise. Each member of the team will be responsible to log in and set up appropriate Hadoop components (NameNode or DataNode) on each CloudLab node. In Figure 3, the Web-UI of a Hadoop cluster is shown accessible via a normal browser. This is due to CloudLab's availability of public IP addresses that can be requested and attached to nodes in an experiment.

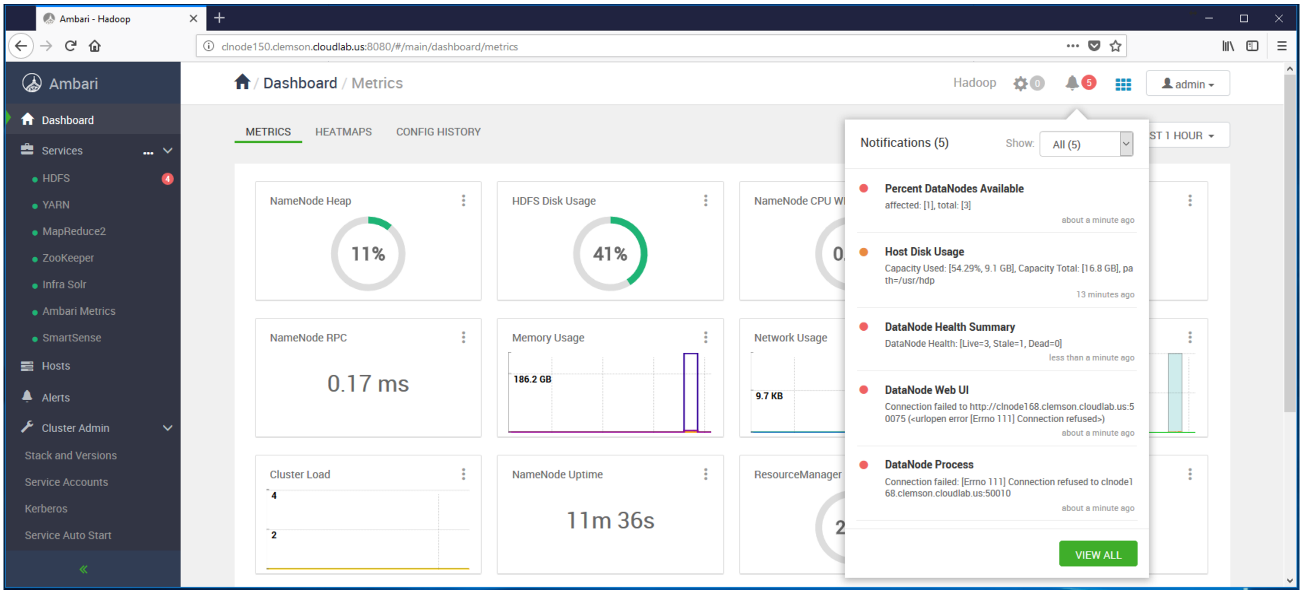

Figure 4: Public interface using Apache Ambari of the Hortonworks Hadoop Distribution cluster deployed inside CloudLab

3.3 CloudLab and On-site Computing Resource: Fall 2016

By Fall 2016, a new dedicated Hadoop cluster was deployed on site at Clemson University. To further expose students to the difference between the core open-source version of Hadoop and its industrial counterpart, another assignment has been added to the course. In this assignment, students were asked to once again deploy Hadoop on CloudLab. However, they were to use an open-source variety of Hadoop that was offered through Hortonworks [11], whose Hadoop distribution is used in the deployment Clemson University's Hadoop cluster. This distribution of Hortonworks is open source, and the complex deployment and management process is performed through Apache Ambari [21], another open source product from Hortonworks. Once the Hadoop cluster had been deployed, students were to deactivate one of the DataNodes and observe the behavior of the Hadoop Distributed File System in the event of failure. Through the process, the students would also have a chance to see how Apache Ambari helps providing a user-friendly interface to help managing all components of a Hadoop infrastructure, as shown in Figure 4.

3.4 CloudLab Bare-metal: Spring 2017

Starting in Spring 2017, efforts have been made to introduce project-based learning to the course, with the goals of tying all knowledge units taught in the course to a comprehensive picture. To this end, most of the previous semesters' assignments were taken a way and replaced with a semester-long, team-based project, in which students are required to develop a mini-cluster that has all the relevant components including a head node, compute nodes, a scheduler, and a parallel file system. Three individual assignments preceded the project to prepare students for the upcoming work in Linux environment. The CloudLab assignment using XenVM in Fall 2015 was among these assignments. Programming assignments were no longer assigned individually for grading purposes but offered first as in-class activities and later as validation mechanisms for various stages of the project.

The project was divided into five stages, built upon one another and scattered throughout the semester. In the first stage, students created the architecture design for their CloudLab-based research cluster. In the remainder of the stages, they implemented and deployed various cluster components, which correspond to the progression of the lectures [12]. The detailed descriptions for the stages are as follows:

- Stage 1 (2 weeks): Each team was tasked with first researching the various XSEDE sites to find a computing cluster site to be used as a motivation for their projects. To avoid conflict, each team must report on three different sites, ranked based on the team's preference. Subjects to be investigated including topological design, hardware configuration, selection of scheduler and file systems, and selection of open source scientific software.
- Stage 2 (2 weeks): Once a motivation site was selected, each team would first develop a similar (does not have to be exact) topological design and map this design to the corresponding CloudLab resources. An example of such design is shown in Figure 5. Based on this design, each team deployed their computing component, which is consisted of a network of three to four computers. OpenMPI was installed on each node and a demonstration using in-class programming materials was required to show that the nodes were properly connected. Unlike the XenVM-CloudLab assignment, the deployment process was completely done using CloudLab's scripting approach. The selection, configuration, and connection of the compute nodes as well as the installation of software on

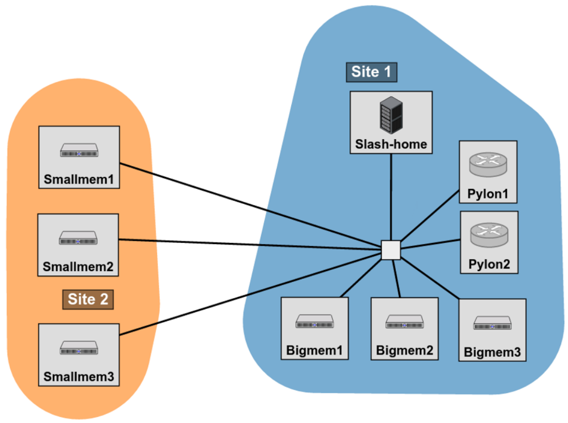

Figure 5: Example of a topological design based on the Bridges Supercomputer from Pittsburgh Supercomputing Center

each node were expressed using the Python programming language.

- Stage 3 (1.5 weeks): In this stage, the computing cluster was to be augmented with a network file system. A new node was to be added to the cluster, and the deployment code was also modified so that OpenMPI was reinstalled on this node and mounted across the compute nodes. In addition, this node also hosted a shared scratch file system. To validate this stage, a MPI heat-exchange programming example that write (in a non-parallel manner) to this scratch location was used.
- Stage 4 (3.5 weeks): Several additional nodes hosting a parallel file system were added to the cluster. This stage was when the teams began to pursue different technical paths due to the variety in parallel file system selections at the XSEDE sites. The options we had included LustreFS, CephFS, and OrangeFS (PVFS2). In-class MPI-IO demos from the lecture was used to validate this stage.
- Stage 5 (2 weeks): In the last stage, a scheduler was added to the node hosting the networked file system. Like the previous stage, different teams would work on different software, which could be one of PBS' varieties, Moab, or SLURM. For validation, each team had to provide a submission script that run one of the examples from the previous stages and can be submitted to their cluster's scheduler.

After the project was completed, the final cluster had an average of six to eight nodes, and each node was deployed directly on a bare-metal CloudLab node. Students were to submit a project report and their final CloudLab deployment script and any other installation and configuration scripts that were needed.

3.5 CloudLab's Openstack Profile: Spring 2018

The contents and target platform for the project was further modified for Spring 2018. In this semester, CloudLab's default experimentation cloud profile which uses Openstack [5] was leveraged. This allowed each team to launch a cloud environment using only three physical CloudLab nodes. Given the relatively large amount of resources available on physical CloudLab nodes, it was possible to launch smaller virtual machines inside this cloud environment, enabling the deployment of a full-scale cluster using fewer physical nodes. Furthermore, an update during Fall 2017 to CloudLab's user interface enabled direct linking between a Github repository containing the CloudLab resource description document and CloudLab's experimentation deployment interface. This significantly reduced the turn-around time for students' projects.

Instead of having each group model their cluster after a distinct XSEDE site, the Spring 2018 project used a single model, Clemson University's Palmetto Cluster, which used Oracle Linux 7 (OL7). This enabled a common required knowledge base, allowing various groups to collaborate and share their expertise. Another pre-project individual assignment was added, in which students first tasked with developing a baseline virtual image running OL7 inside VirtualBox on their personal computers. This assignment served as a preliminary training stage to further prepare students for the necessary Linux administrative tasks for the project, in addition to the XenVM/CloudLab assignment. From a architectural perspective, this did not significantly change how the computing cluster for the project was designed. If anything, restricting the model cluster to Clemson's supercomputer provided a uniform set of software selection, including OrangeFS for the parallel file system and PBS Pro for the scheduler.

On the other hand, the transition to using an Openstack profile on CloudLab to deploy the nodes significantly changed the technical approaches to this project. With Openstack, the entire project became more modularized, with the two processes, installing and configuring individual nodes and deploying them on CloudLab, are clearly separated. These changes are can be captured as follows:

- From the baseline (OL7) image from the initial assignment, students were to develop different images: a) a compute image for the computing nodes, b) a metadata server for OrangeFS, c) an I/O server for OrangeFS, and d) a login node that also server as the metadata server for the PBS Pro scheduler.
- Network topologies were determined prior to the installation and configuration of the images. Domain names and IP addresses were hard-coded into the images.
- Configured images were uploaded to file sharing servers (Dropbox, Box, or Google Drive). The default Openstack profile from CloudLab were modified to enable the downloading and instantiating of these images. For cluster components that had multiple nodes (computing or parallel file system), the same base image was reused for additional instantiation.

Two in-class technical sessions were hosted after the fourth and final stages, in which groups who passed specific technical hurdles shared their experience with others. The amount of computing nodes for the final cluster were six nodes, but most project submission required only two bare-metal CloudLab nodes to run. Figure 6 shows example of a successful project submission.

4 DISCUSSION

While CloudLab has been used in the course described in this paper since 2015, the extensive usage of the computing environment in

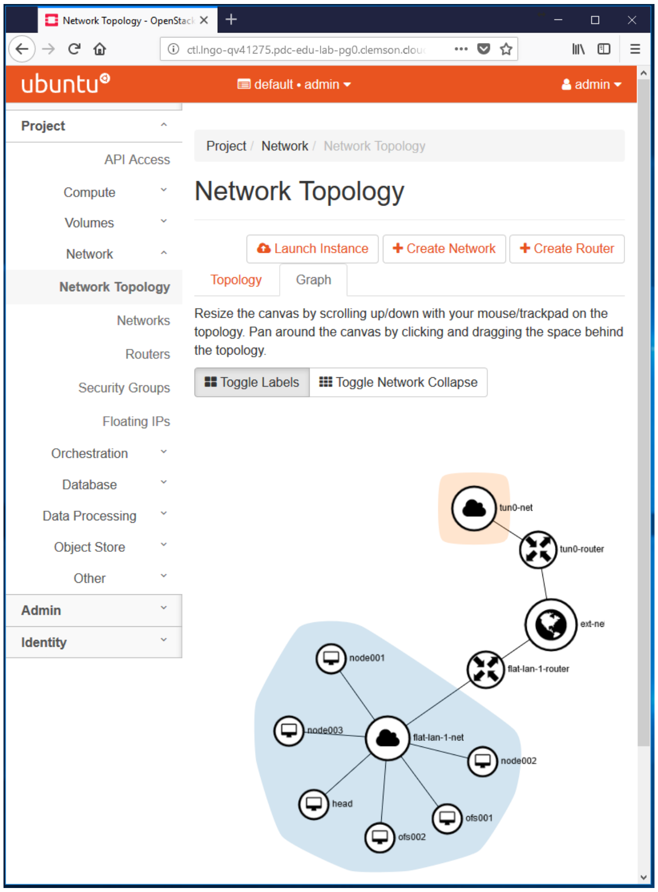

Figure 6: Example of a topological design deployed through CloudLab's default Openstack profile

a semester-long class project only started in Spring 2017. For this semester, there were 60 students in the class, divided into 15 teams. The group number decreased to 14 later during the semester due to students dropping from the course. In the next offering of the course and the class project, there were 33 students, divided into 11 teams. The nature of the project enables students to have extensive experience interacting with the Linux environment. This has led to overwhelmingly positive feedback from students, particularly regarding the hands-on and collaborative nature of the project, as well as the applicability of the course's content. However, there remains several challenges in using CloudLab in a project-based learning environment.

The largest issue that we noticed in using CloudLab, as with any remote and shared resources, was the potential competition with researchers and students from other institutions for computing hours. We experienced this issue in Spring 2017, since each team deployed directly on bare-metal CloudLab nodes. As a result, provisioning enough resources for the entire class became a challenge, and this hindered students' ability to complete some stages of the project in a timely fashion. The changes to the project made in Spring 2018 had been done partly to address this issue. By deploying virtual nodes inside fewer physical nodes, students could request adequate resources for their work. Furthermore, they could work offline to develop the baseline images and only used CloudLab for large-scale testing and validation. A minor issue is the amount of time it took to fully instantiate a complex CloudLab experiment. In both versions of the project (Spring 2017 and Spring 2018), the average time required to have the project ready ranged between 45 minutes to one and a half hours. This reduced the effectiveness of in-class sessions.

In our first version of the project in Spring 2017, we observed several pedagogical challenges to our learning approach. Firstly, a significant portion of students lacked the fundamental Linux skills to get started. A single XenVM CloudLab assignment was not enough to equip students with the necessary Linux administration skill to meaningfully contribute to the team. Secondly, since each group used a different XSEDE site as a model for their mini-cluster, this led to different software stacks and network topologies. As a result, in-class collaboration between the groups was not fruitful. The second version of the project in Spring 2018 had been modified to address these challenges.

5 RELATED WORK

There exists previous work describing the usage of local computing resources to teach distributed and cluster computing. In this section, we focus on related work that uses publicly available resources. The Extreme Science and Engineering Discovery Environment (XSEDE) is the largest and most popular resource for computationally-enabled sciences [20]. In addition to supporting scientific computing, XSEDE also makes resources available for computer science education. As XSEDE is a research production environment, the majority of its computing sites support programming topics in distributed and parallel computing rather than system-related topics [6, 10, 22]. Jetstream is another large-scale resource that is part of XSEDE. However, unlike the traditional HPC model of XSEDE, Jetstream offers users the ability to launch custom virtual machines images on bare-metal nodes for extended amount of time. Educational workshops and courses have leverage this capability of Jetstream to support topics in large-scale data-enabled sciences such as genomics, neuroimaging, and big data management [9]. CloudLab has also been used in a number of computer science education courses, but they primarily focus on areas such as computer and network security, networking, and cloud computing [1, 14].

6 CONCLUSIONS

In this paper, we have discussed our experience in using CloudLab to teach cluster computing topics. Between the two modes of deployment, bare-metal and cloud-based, we were able to provide students with a computing environment that enabled both hands-on and project-based learning. The flexibility, availability, and scale of CloudLab bring significant applicability to other topics in computer science, including operating system, networking, and cyber-security.

REFERENCES

[1] Jay Aikat, Michael K Reiter, and Kevin Jeffay. 2016. Education Modules for Networking, Cloud Computing, and Security in Systems Courses. In *Proceedings*

of the 47th ACM Technical Symposium on Computing Science Education. ACM, 400–400.

[2] Amy Apon, Rajkumar Buyya, Hai Jin, and Jens Mache. 2001. Cluster computing in the classroom: Topics, guidelines, and experiences. In *Cluster Computing and the Grid, 2001. Proceedings. First IEEE/ACM International Symposium on*. IEEE, 476–483.

[3] Mark Berman, Jeffrey S Chase, Lawrence Landweber, Akihiro Nakao, Max Ott, Dipankar Raychaudhuri, Robert Ricci, and Ivan Seskar. 2014. GENI: A federated testbed for innovative network experiments. *Computer Networks* 61 (2014), 5–23.

[4] William I Bullers Jr, Stephen Burd, and Alessandro F Seazzu. 2006. Virtual machines-an idea whose time has returned: application to network, security, and database courses. In *ACM SIGCSE Bulletin*, Vol. 38. ACM, 102–106.

[5] Antonio Corradi, Mario Fanelli, and Luca Foschini. 2014. VM consolidation: A real case based on OpenStack Cloud. *Future Generation Computer Systems* 32 (2014), 118–127.

[6] Andrew Danner and Tia Newhall. 2013. Integrating parallel and distributed computing topics into an undergraduate CS curriculum. In *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International*. IEEE, 1237–1243.

[7] Joshua Eckroth. 2016. Teaching big data with a virtual cluster. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. ACM, 175–180.

[8] Jesse Eickholt and Sharad Shrestha. 2017. Teaching big data and cloud computing with a physical cluster. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. ACM, 177–181.

[9] Jeremy Fischer, David Y Hancock, John Michael Lowe, George Turner, Winona Snapp-Childs, and Craig A Stewart. 2017. Jetstream: A cloud system enabling learning in higher education communities. In *Proceedings of the 2017 ACM Annual Conference on SIGUCCS*. ACM, 67–72.

[10] Steven I Gordon. 2013. Advancing computational science education through xsede. *Computing in science & Engineering* 15, 1 (2013), 90–92.

[11] Hortonworks. 2018. Hortonworks Data Platform. https://hortonworks.com/products/data-platforms/hdp/.

[12] Linh Bao Ngo. 2018. CPSC 4770-6770: Distributed and Cluster Computing. https://github.com/linhbngo/cpsc-4770_6770.

[13] Linh Bao Ngo, Edward B Duffy, and Amy W Apon. 2014. Teaching HDFS/MapReduce systems concepts to undergraduates. In *Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International*. IEEE, 1114–1121.

[14] Younghee Park, Hongxin Hu, Xiaohong Yuan, and Hongda Li. 2018. Enhancing Security Education Through Designing SDN Security Labs in CloudLab. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. ACM, 185–190.

[15] Sushil K. Prasad, Almadena Chtchelkanova, Frank Dehne, Mohamed Gouda, Anshul Gupta, Joseph Jaja, Krishna Kant, Anita La Salle, Richard LeBlanc, Andrew Lumsnaide, David Padua, Manish Parashar, Viktor Prasanna, Yves Robert, Arnold Rosenberg, Sartaj Sahni, Behrooz Shirazi, Alan Sussman, Chip Weems, and Jie Wu. 2017. NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing - Core Topics for Undergraduates. http://www.cs.gsu.edu/ tcpp/curriculum/index.php.

[16] Robert Ricci, Eric Eide, and CloudLab Team. 2014. Introducing CloudLab: Scientific infrastructure for advancing cloud architectures and applications. *; login:: the magazine of USENIX & SAGE* 39, 6 (2014), 36–38.

[17] Robert Ricci, Kuang-Ching Wang, Srinivasa Akella, Michael Zink, and Glenn Ricart. 2017. CloudLab Phase II: Community Infrastructure To Expand the Frontiers of Cloud Computing Research. https://www.nsf.gov/awardsearch/showAward?AWD_ID=1743363.

[18] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. 2010. The hadoop distributed file system. In *Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on*. Ieee, 1–10.

[19] Mark Stockman. 2003. Creating remotely accessible virtual networks on a single PC to teach computer networking and operating systems. In *Proceedings of the 4th conference on Information technology curriculum*. ACM, 67–71.

[20] John Towns, Timothy Cockerill, Maytal Dahan, Ian Foster, Kelly Gaither, Andrew Grimshaw, Victor Hazlewood, Scott Lathrop, Dave Lifka, Gregory D Peterson, et al. 2014. XSEDE: accelerating scientific discovery. *Computing in Science & Engineering* 16, 5 (2014), 62–74.

[21] Sameer Wadkar and Madhu Siddalingaiah. 2014. Apache ambari. In *Pro Apache Hadoop*. Springer, 399–401.

[22] XSEDE. 2018. Course Catalog. https://portal.xsede.org/training/course-catalog.