
Using Cloud Computing to Support Higher Education in Networks and Systems

Jianwei Liu, Yunhui Fu, James Martin*

Clemson University

*Corresponding author

Abstract: Virtual Machine technology is firmly in place in University computing infrastructures. Here at Clemson University, and we think this is also true at other institutions, there is no standard mechanism in place to manage VMs. As a consequence, departments or individual faculty are likely to setup VM-based infrastructure to support education on an as-needed basis. Engineering programs, and in particular programs such as Computer Science, Computer Information Systems, Computer Engineering, or even areas in mathematics or operations research, must teach problem solving using a wide variety of software systems that likely include high performance computing (HPC) or big data programming environments. These systems require specific systems of computers and software to provide appropriate learning environments. Cloud computing is emerging as a possible approach that meets the rapidly evolving computing requirements faced by higher education.

The technology and ideas surrounding cloud computing have evolved significantly over the last several years, further obfuscating the term. At the highest level, cloud computing provides services to users over a network. The nature of the service, the target user community, and the economics that drive the use and adoption of the system all contribute to a diverse range of cloud computing systems. We focus on cloud computing to support higher education. To provide focus to our work, we focus primarily on courses that involve networks or systems. These courses tend to be hands-on, lab oriented courses that involve writing and testing software in realistic system environments. Prior to VMs, these courses might require designated lab machines. Over the last 5 years, these courses tend to be the main users of VM technologies. Further, while the courses might be repeated regularly, the VM images must be updated and extended (sometimes during the semester!). We begin our evaluation with the open source cloud computing platform referred to as the Virtual Computing Lab (VCL) [5]. The main goal of VCL is to make available dedicated, custom compute environments to users. The compute environments can range from something as simple as a virtual machine running productivity software to a cluster of powerful physical servers running complex HPC simulations. NCSU and other schools in North Carolina rely on VCL for managing compute resources for faculty, staff, and students (<https://vcl.ncsu.edu/>). We built a small VCL cloud and trialed its use to support several networking courses in the 2013/2014 academic year. This paper provides a snapshot of our project. We provide the details of our VCL deployment as well as the learning environments that we make available and incorporated in our courses. The following learning environments have been developed:

- 1) A set of images that formed a virtual network testbed that was used by students studying TCP/IP network protocols and performance.

- 2) A set of images that offered hands-on experimentation with OpenFlow systems

- 3) A set of images that provided a network simulation modeling platform.

The paper includes a summary of related or alternative directions. We describe our experiences in the context of the related work that has been published in the area of network testbeds that support higher education.

Keywords: Cloud Computing, Network Testbeds, Technology for Higher Education

1 Introduction

Nowadays, increasing number of courses in universities requires computer software to assist the teaching procedures. As more and more softwares are requested to be installed on campus computers, software licensing and management in any university is becoming a challenging problem. Meanwhile, security and failure-resilience of those systems is more critical than personal computing, since they are open to large amount of users. Traditionally, a group of campus computers are imaged to the same operating system and software set, while instructors do not have the permission to change the image. This approach has several disadvantages. Firstly, it is not flexible enough. When met software related limitation, the best thing instructors can do is to submit an IT help ticket and wait until the IT staffs to re-image all or a group of computers. (And due the risk of upgrading, this procedure is usually time-consuming). Moreover, some courses may require root permission of computers for hands-on experiments. For example, the RPC (Remote Process Communication) lab we designed for our undergraduate networking course can not be run on the Linux lab machines at Clemson, since the service was turned off for security reason on those machines. Secondly, this approach is not flexible in software licensing, which leads to waste of licensing expenses. It's very common that software for some course only need a small portion of the departmental computers. Installing everything on the same system will also lead to deteriorated performance and potential software conflicts.

Virtual Machine (VM) is a very promising solution to the above issues. As images of VMs can be tailored by instructors based on their unique teaching need, software on VMs will well suit the need of instructors, or even a specific homework assignment by one instructor. Since running in some container, users can be safely provided with root permission without worrying about the security issue that may follow as in the bare-metal machine approach above. At the same time, as users being able to take snapshots of the VMs, it is much more resilient to system failure. However, with large number of images, different configurations of computers and various user permissions, a system that can manage all the resources is highly desirable.

IBM and North Carolina State University have developed a cloud computing platform referred to as the Virtual Computing Lab (VCL). VCL is now an open-source Apache project. The main goal of VCL is to make available dedicated, custom compute environments to users. The compute environments can range from something as simple as a virtual machine running productivity software to a cluster of powerful physical servers running complex HPC simulations. NCSU and other schools in North Carolina rely on VCL for managing compute resources for faculty, staff, and students (<https://vcl.ncsu.edu/>). We used VCL to construct our private cloud-computing environment for the network and system classes at Clemson. In this paper, we will detail how we set up the VCL platform with very limited hardware resource, while meeting the requirement of accommodating relatively large number of users. We will also review and compare the related approaches in constructing virtual labs, and summarize the lessons we learned.

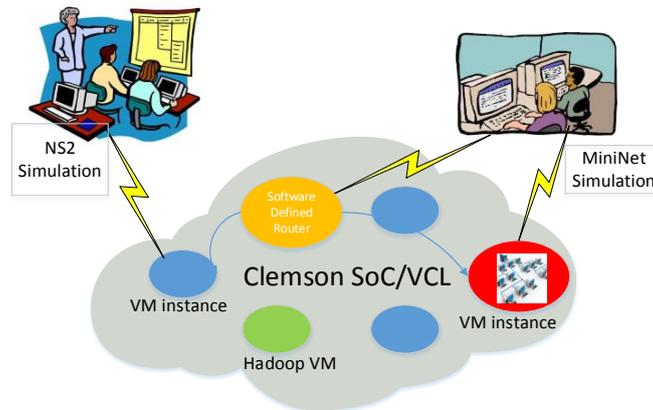


Figure 1: The concept of our cloud based virtual lab

There are several alternative approaches we have thought of when we started to construct our virtual lab environment. The Global Environment for Network Innovations (GENI) [6] is a NSF sponsored effort to create a large scale testbed for network experimentation. GENI is aimed to provide a federated network supporting repeatable network experiments. It's increasingly used in the classroom as reported in the latest GEC 19 [12]. A wide range of tools are created to use GENI for educational purpose, like LabWiki[4] and iRodsWeb [3]. Users can load default or customized images to the VMs located at campuses across the US, and use Software Defined Network (SDN) to stitch them together for experimentations. Currently both inner-aggregate and inter-aggregate experiments are supported. Users can easily design the network topology through web based tools like FLACK [1] and Flukes[2]. We thought of using GENI to provide VM and data collection services. However, with the concern of learning curve of students, reliability and flexibility of the public cloud, we decided to try constructing a private cloud using VCL at last. We believe, as a near-campus cloud, VCL can provide more flexibility. For example, with GENI, it's hard to access some node with VNC even with the VNC sever running on your image. We can not do anything to fix that because it's a platform issue. But, with VCL, we can simply revise the code to enable that. GENI has its own advantages though. First, it is available without the additional deployment efforts, as the infrastructure and functionality are already provided by the GENI network. Second, comparing with VCL, GENI does a better job in software-defined network support, as it is designed with that in mind. OpenFlow is supported on it, and students can design innovative network topology to test interesting ideas. VCL currently still lacks this capacity, which is one of the features we are really looking forward to as we will describe in Section 6.

OpenStack [9] is an enterprise-level VM management framework that is now widely used for data center VM management. Comparing with OpenStack, VCL is much more lightweight, and easier to deploy and configure. SaltStack [10] is another VM management framework which is very popular recently. It's reported that using SaltStack can largely reduce the time of orchestration[13]. However, we believe it is still too complicated to configure comparing with VCL.

Like other major universities around US, Clemson owns its cluster, named Palmetto. Palmetto now has about 1978 nodes and 19880 cores. Users can use PBS script to request certain types of computer resource, e.g. by specifying the number of cores, the amount

of memory and the interconnect type, etc. Comparing with our setup of VCL platform, palmetto is aimed as a general purpose super-computing platform. The resource allocated to users is on bare-metal machines instead of VMs. Therefore, users can not control the images of those computers and are generally not allowed to access them as root users. Even if they run VMs on the nodes, only TAP connection [11] can be used, which will significantly limit bandwidth. At the same time, performance/QoS provision on cluster like palmetto is not well guaranteed since it lacks the isolation of different allocations. If some task consumes most of CPU resource at one node, or has significant memory leakage, the other tasks running on that node will become very slow. For a cloud used for networking teaching or experiment purpose, this difference will likely render different results for the same experimentation.

Based on the above considerations, we have deployed our cloud based virtual lab using VCL, as shown in Figure 1. The cloud in the middle represents the Clemson School of Computing network, or our VCL cloud service in the core. Inside this cloud, the different ellipses signify various VM images that can be instanced for teaching purposes. We hope some of them can serve as software defined router, and some MiniNet based VM and contain simulated network in it, and therefore link together as a large scale of simulated network. We developed some customized images that we will detail in Section 4. Students can load those images provided by instructors for specific homework, and save their own images as needed.

2 Related work

Due to the flexibility of cloud computing payment scheme, small and media size institutions are likely to commercially benefit from adoption of cloud based solution rather than purchasing physical machines.[16]

[15] further argued that the budget in educational institutions is usually limited, and experimented VCL based cloud computing solution in K-12. From their experience, VCL can provide computational resources for teaching and learning both cost-effectively and with the flexibility that education requires. They analyzed the benefits like software license cost savings and extension of machine life time. Also identified in that paper are some factors that may hinder the application of this new technology, such as worries of breaching Children's Internet Protection Act and human factors of being reluctant to change. Comparing with the field of use in [15], our application of VCL to higher education, specifically computer network and system courses, do not have those concerns. The computer science major students has the broadband access to Internet, are willing to learn this new technology.

As developed in NCSU, VCL has been used in the NCSU campus for a long time. [14] gives an introduction how the VCL grows, and the usage of VCL across the US at that time. From the description, we have identified several interesting points. First, comparing with our usage, they have the support from HPC(High-Performance Computing) group at the NCSU and the IBM BladerCenter hardware. To further scale up the VCL cloud service at Clemson, we also need to cooperate with the CCIT(Computing and Information Technology) at Clemson to provide better QoS. Second, similarly as a "pilot" trial, we believe the testbed we constructed has achieved the goal, and has a similar growing potential later as the NCSU platform.

[17] provided their solution of a virtual laboratory platform for Hands-On Networking Courses. Developed by themselves from scratch using XEN, their solution has the advantage

of better control on the software written by themselves, and multi-location support as they claimed. However, as stated in the later part of the paper, reliability is a bigger issue for a virtual lab system. This is actually one important factor we considered when we chose how to construct our virtual laboratory. With the verification and testing usage at several universities and other institutions, we believed VCL can provide better reliability and reusability to our lab, and it's proven in our trial.

3 Deployment of VCL platform

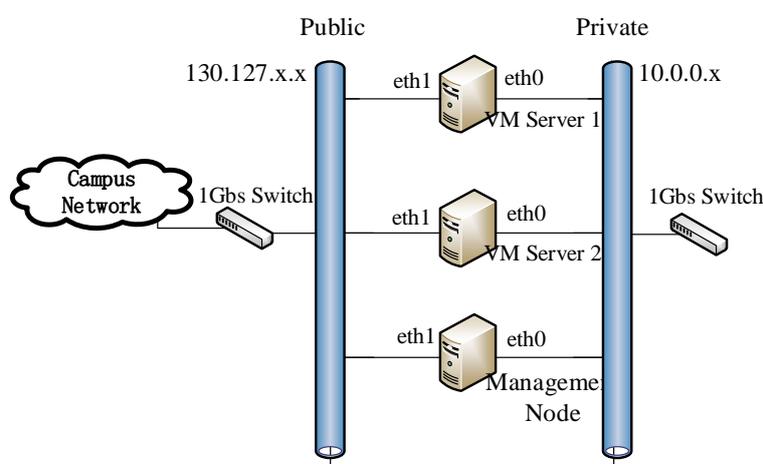


Figure 2: Topology of VCL Deployment

We deployed the network with the topology showed in Figure 2. VM Server 1 and VM Server 2 are two servers on which all VMs will operate. Each of them has 16GB memory and 4 cores. Due to the hardware limitation and homework need, we limited the resource of each VM to 500MB memory. To make the system more failure-resilient, we deployed the VCL management node in a third server as a VM. Therefore, we can easily recover it if there was any software/hardware failure on that machine. All the VCL related modules including the MySQL database run on that machine. We also set up Network File System (NFS) among those machines to share the VM images. Another machines was set up as an image back-up repository.

Since we want the students be able to access the VMs using Virtual Network Computing (VNC), we made some revision to the VCL source code. The problem we met in setting up the VNC is that there are two password related to VNC, i.e. the password for ssh to the image, and the password for the VNC software to connect to the image. The VCL only supports revision of the former, and leads to failure of VNC connections due to password mismatch error. So, we changed the VCL source code to unify the two password, and therefore enabled the VNC connection feature in VCL.

Due to the limitation of hardware resource as to the number of students, we let the students form 10 groups to cooperatively work on projects. Each group can reserve one VM

at a time. So there will be at most 10 VMs running at the same time. In the VCL setting, we map 5 VMs to each server. However, due to the amount of machines needed by labs, and instructor/TA's trial reservations, we extended that to 10 VMs per machine at the later part of the semester. VCL is responsible to the load balancing as it will select a machine that has more resource available and run an image in a VM on that machine.

4 Customized images for networking class learning

Based on the needs of different courses and sometimes the different needs of homeworks in one class, we have designed the following customized images. For CPSC 851, we designed images with NS2, MiniNet, and VNC support. We generated KVM(Kernel-based Virtual Machine) `qcow2` images of those Operating Systems(OSs), and imported ¹ those images to VCL by the executable program `vcl.d`.

NS-2 Simulation Scenario: The open source ns2 simulator is the defacto simulation tool used by the Internet research community. It is a discrete-event simulator that faithfully reproduces a wide set of TCP/IP protocols and applications. The simulator used in the class will be patched with several modules for teaching purposes.

Network Experiment Scenario: Each group will be given access to a simple three node system. This is illustrated in Figure 1 by the ellipse showing nodes interconnected by a router. The students will learn basic network administration and networking skills. The group's account on the three virtual nodes will allow sudo access to basic administration commands that are necessary to setup the network environment and to monitor system configuration and performance. The students will develop or utilize existing open source network performance and security software on the testbed.

MiniNet Scenario: As described in [8], MiniNet-HiFi is a of container-based emulation tool that has been introduced in an effort to reproduce network experiments running real code on an emulated network using VM technology. Building on the knowledge obtained from the previous scenarios, we use MiniNet to introduce Software Defined Network and OpenFlow.

Hadoop Scenario: Hadoop [7] is a distributed computing framework including three important components, i.e. HDFS, MapReduce and YARN. We has created a KVM image with Hadoop installed for the VCL platform, and planning to design some experiments for students to learn Hadoop using our VMs.

4.1 Case study

We also did the following experiments to verify the property of VCL in isolating the network usage. We first started one VM and created a high speed `iperf` from that VM to our shared server VM (which is not on the two VCL VM servers). Then, we repeat but have 4 `iperf` experiments going from 4 different student VMs. We plot the evolutions of the two scenario's throughputs in Figure 3. As we can see, the result getting from the two experiments are very similar. From our experience, the VCL based VMs can provide students with pretty consistent networking performance under our VM and machine settings.

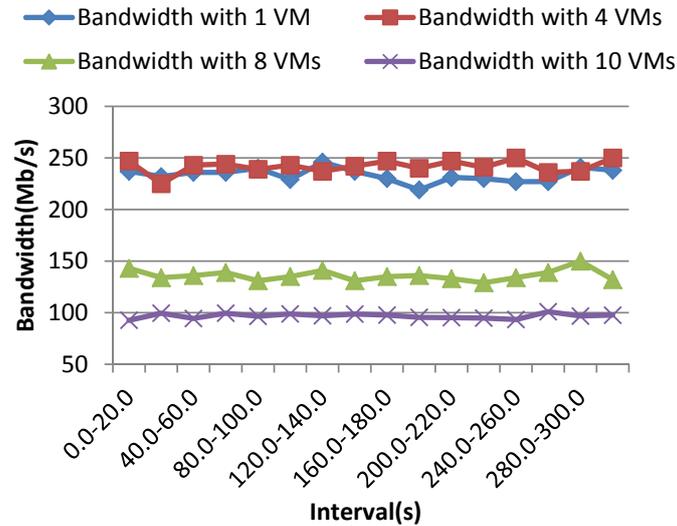


Figure 3: Comparison of bandwidth test under different VM loads

5 Evaluation and results

We used the statistics collection feature in the VCL web interface to keep track of the usage of our system. And, at the end the 2013 Fall semester, we collected the following usage statistics and figures. As shown by the statistics, we had 171 reservations and 24924 hours reserved in total. We can see the number of reservations to different images in Figure 4. As we can see, the image with VNC feature got more reservations, which indicates that the VNC support we added is useful/required in some labs. We can also see some failures of those images. That's because we had short period of server problem in the middle of term, which prevented the student from reserving. We hope to ameliorate this problem by updating our hardware in cooperation with CCIT at Clemson.

	Reservations	Unique Users	Hours Used	< 2 min load time	>= 2 min load time	Failures
CentOS 6.4 i386 Base:	39	9	6194	34	2	2 (5%)
CentOS 6.4 i386 Base_NS2:	29	8	8050	25	1	1 (3%)
CentOS 6.4 i386 Gnome and VNC:	54	9	4979	45	6	1 (1%)
CentOS 6.4 x86_64 Base:	31	3	3245	30	0	9 (29%)
CentOS 6.4 x86_64 Gnome and VNC:	27	4	2453	23	3	3 (11%)
Mininet 2 Base:	2	1	1	1	1	0

Figure 4: The usage statistics of different images provide in CPSC 851

6 Lessons learned

We summarize the lessons we learned from our trial of VCL usage as follows.

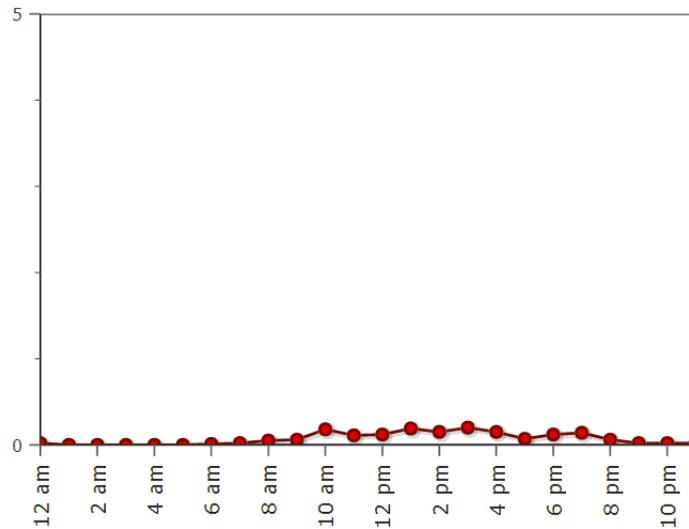


Figure 5: Hourly usage

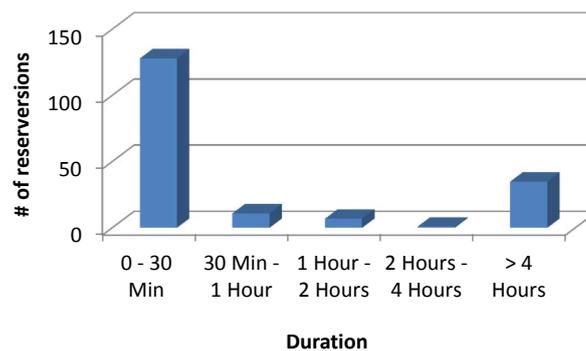


Figure 6: Duration of reservations

- Due to the limitation of hardware resource and trial property of this experiment, we limited our trial to 3 courses (CPSC 424, 851, 852). It's clear that all of our systems courses can make use of the capability (CPSC 360,362,424,851,852). If we can get support from CCIT at Clemson or further cooperate with IBM, we believe we can get more VCL usage at Clemson, and scale up the virtual lab.
- The students overwhelmingly want hands-on, realistic lab exercises. Further, in our opinion, these projects need to be individual. However, due to the limitation of resource, we can only make them work in group. This has partially impaired the learning advantage we aimed to achieve, since usually only one or two students ends up learning everything.
- The system needs to have the capabilities to accurately emulate the virtual networks. As we found our trial, VCL does not support SDN like in other alternatives we mentioned in the Section 1. We mark this as a feature that are highly desired in the future, as

students then can use SDN in multiple VM and a more realistic setting, as oppose to our current MiniNet virtualization approach.

- The cost of deploying a VM system sufficient to support a Virtual Network lab environment to each student is high - both in terms of bare wire machines, disk space, and admin/operational costs. To benefit from scale, the system should be a capability provided by the University computing services.

7 Conclusions

In this paper, we explored the construction of a cloud based virtual lab for computer network courses at Clemson using VCL. We introduced how we set up the lab with limited private hardware resource, which can be an example for instructors from other universities. Also, we compare VCL with some other alternatives, like GENI, OpenStack, HPC cluster at Clemson University, and summarized their benefits and drawbacks for this particular purpose. To meet the teaching requirement of specific networking courses, we designed some customized images. From our trial experience, we learned some limitations of the current VCL for the purpose of networking course related cloud management. The most important part is that it does not support OpenFlow SDN comparing with some other alternatives. We also learned some lessons that is related to our trial stage hardware limitations, which the other pilot educators may also encounter. We describe those lessons in the hope that they can avoid those problems. As for the result of this trial, we believe it is very successful. Our VCL service was successfully set up during summer time, and running for the 2013 Fall semester (with only a short period down time due to IP problem in our lab). From the instructor's perspective, though some first time setting up efforts are required, the amortized cost is lower than setting up a bunch of new KVMs every time from scratch each semester. They also obtain more permission management options and image version control capacities. From the perspective of students, they are happy with the cloud based service, since they can now access the images with root permissions, and save their changed as needed.

8 Future work

In the future, we would like to continue our projects in the following four directions.

1. Cooperate with CCIT and Palmetto team at Clemson to provide larger-scale resource provision at Clemson.
2. Study how to revise VCL to support Software Defined Network(SDN), which is highly desired in our computer networking related courses.
3. Explore the features needed for using VCL to support Computer Security courses at Clemson. We hope to build a customized version of VCL that can have better support in system recovery, and a platform for the computer security instructors around US to share the images and resource definition files used for the teaching.

Acknowledgements

This project was partially supported by IBM, and we would like to thank for their support.

References

- [1] Flack. <http://www.protogeni.net/wiki/Flack>.
- [2] Flukes. <https://geni-orca.renci.org/trac/wiki/flukes>.
- [3] iRodsWeb. <http://witestlab.poly.edu/svn/listing.php?repname=irodsweb>.
- [4] LabWiki. <http://labwiki.mytestbed.net/>.
- [5] VCL(Virtual Computing Lab). <http://vcl.apache.org/index.html>.
- [6] GENI. <http://www.geni.net/>, January 2014.
- [7] Hadoop. <https://hadoop.apache.org>, 2014.
- [8] Mininet. <http://mininet.org>, 2014.
- [9] OpenStack. openstack.org, 2014.
- [10] SaltStack. www.saltstack.com, 2014.
- [11] Starting qemu with a tap interface. 2014.
- [12] GEC 19. GENI In Classroom. <http://groups.geni.net/geni/wiki/GEC19Agenda/GENIClassroom>.
- [13] Mike Kavis. Saltstack is gaining momentum. <http://www.virtualizationpractice.com/saltstack-is-gaining-momentum-22170/>, 2013.
- [14] H.E. Schaffer, S.F. Averitt, M.I. Hoit, A. Peeler, E.D. Sills, and M.A. Vouk. NCSU's Virtual Computing Lab: A Cloud Computing Solution. *Computer*, 42(7):94–97, July 2009.
- [15] Sarah Stein, Jennifer Ware, Johanne Laboy, and Henry E. Schaffer. Improving K-12 pedagogy via a Cloud designed for education. *International Journal of Information Management*, 33(1):235 – 241, 2013.
- [16] Nabil Ahmed Sultan. Reaching for the "cloud": How smes can manage. *Int. J. Inf. Manag.*, 31(3):272–278, June 2011.
- [17] Le Xu, Dijiang Huang, and Wei-Tek Tsai. V-lab: A cloud-based virtual laboratory platform for hands-on networking courses. In *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE '12*, pages 256–261, New York, NY, USA, 2012. ACM.