Bridging Hardware Barriers in Operating Systems Education through Hybrid Cloud Infrastructure

Peng Kang
Department of Computer Science
California State University, Sacramento
Sacramento, CA, USA
peng.kang@csus.edu

Palden Lama
Department of Computer Science
The University of Texas at San Antonio
San Antonio, TX, USA
palden.lama@utsa.edu

Abstract

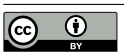
This paper presents a hybrid cloud approach to the delivery of lab environments of operating systems by integrating the NSF Chameleon Cloud (public) with the virtual server provided by the university (on-premises). Students gain access to preconfigured, high-performance virtual machines for implementing and testing their operating systems, eliminating hardware barriers while ensuring both scalability and reliability. In this model, Chameleon Cloud provides scalable resources, while the virtual servers provided by university ensures security and institutional control. The framework enhances accessibility, promotes equity in operating systems education, and prevents overprovisioning by accommodating CPU demand that varies between 60 and 120 cores over the semester for a class of 40 students.

ACM Reference Format:
Peng Kang and Palden Lama. 2026. Bridging Hardware Barriers in Operating Systems Education through Hybrid Cloud Infrastructure. In *Proceedings of the 57th ACM Technical Symposium on Computer Science Education V.2 (SIGCSE TS 2026), February 18–21, 2026, St. Louis, MO, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3770761.3777182

1 Introduction

Hands-on practice is essential for mastering operating system concepts such as scheduling, memory management, interprocess communication, and I/O management. At the University, the Operating Systems course has traditionally provided these experiences through hardware labs and, more recently, locally hosted virtual machines (VMs). While relying on the department's hardware labs was not scalable, using students' personal laptops to host VMs presented several challenges observed in recent semesters: Apple M-series laptops cannot run x86 VMs, older Windows laptops suffer from performance limitations, setup consumes valuable learning time, and assigning each student a dedicated VM leads to resource waste and unpredictable demand due to fluctuating enrollment.

To address these issues, we propose a hybrid cloud solution that combines the scalability of NSF Chameleon Cloud (public) with the security and reliability of the Virtual Server built by university (on-premises). This architecture provides students with seamless access to preconfigured, high-performance environments, allowing them to focus on operating systems implementation with a smoother and more consistent user experience. It prevents over-provisioning by accommodating CPU demand that varies between 60 and 120 cores over the semester for a class of 40 students.



SIGCSE TS 2026, St. Louis, MO, USA

ACM ISBN 979-8-4007-2255-4/2026/02
https://doi.org/10.1145/3770761.3777182

2 Related Work

Cloud-hosted virtual laboratories have been shown to reduce setup overhead, improve accessibility, and provide reproducible environments for computing education. Previous work highlighted benefits such as faster start-up times, persistent storage, and more reliable delivery of system-level content [1]. Saligrama et al. demonstrated teaching cloud infrastructure and scalable application deployment, and our work offers a complementary case study for engaging students in this process [3]. Chameleon Cloud has become a widely used academic testbed, supporting both research and teaching with ready-to-use appliances and streamlined VM workflows [2].

3 Course Description

Our undergraduate operating system course has historically required students to implement a real operating system on physical PC hardware. While this approach provided authentic hands-on experience, it also demanded dedicated lab space and made hardware maintenance and upgrades difficult to manage. Our current course for computer science and computer engineering students is structured to balance theoretical foundations with practical systems implementation. To achieve this balance, the curriculum incorporates hardware-level concepts, such as interrupts, timers, keyboard input, VGA output, and serial port communication, alongside simplified operating systems. To support this design, the System Programmer's Educational Development Environment (SPEDE), is deployed within a virtual machine, providing a complete toolchain that simulates physical PC hardware for operating systems development as shown in Fig. 1. SPEDE allows students to engage in realistic systems programming without requiring direct access to physical lab machines. The course is structured around seven projects that build sequentially to deepen understanding. Students begin with P0, learning the SPEDE environment and debugging tools, then proceed through P1 (device drivers) and P2 (timer interrupts and scheduling) to establish core process control. They then implement P3 & P4 (process management and system calls), P5 (inter-process communication), P6 (file systems), and P7 (virtual memory with user process).

4 Challenges

A direct one-to-one allocation of virtual machines for each student through Chameleon Cloud or server provided by university would

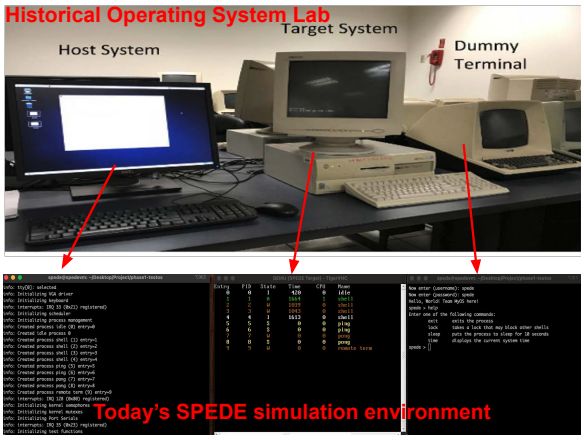


Figure 1: Historical operating systems labs versus current SPEDE simulation environment hosted on virtual machines. The host system handles debugging and OS image downloads, while the dummy terminal tests serial communication.

result in significant resource inefficiency and increases the maintenance burden. Student resource requirements vary throughout the semester. With an average enrollment of 40, planning begins at two CPU cores per student for P0. In P1 & P2, heavy use of remote desktops for hardware simulation raises demand to about three cores per student (target, debugging, and visualization), reaching a peak of 120 cores. By P3, students reduce reliance on remote desktops, lowering usage to about 1.5 cores each and stabilizing total demand near 60 cores. The final two projects are optional, further reducing overall requirements. However, shared hardware presents its own challenges. Provisioning large VMs from Chameleon is challenging, as VM leases may not always be available given that the Chameleon Cloud is shared by many institutions across the United States [2].

Authentication and system security present additional obstacles. Students frequently forget passwords or choose weak credentials, creating vulnerabilities. While the server provided by university integrates stronger authentication mechanisms (two-factor authentication) and offers good protection, its limited resources and lack of scalability restrict its usefulness. Each year, administrators must estimate student enrollment to pre-provision resources, which often leads to either overutilization at the start of the semester or underutilization later as demand decreases.

5 System Architecture

Our design consists of two components. First, two university servers (on-premise) act as gateway nodes for student authentication and identity verification. Each server is equipped with 20 CPU cores and 32 GB of memory, and the two are synchronized via a Network File System (NFS) to maintain identical configurations, allowing students to connect seamlessly to either server. Second, the Chameleon KVM infrastructure, supported by OpenStack, provides a scalable resource layer. Each standard KVM instance (m1.xlarge) is provisioned with 8 CPU cores and 16 GB of memory, ensuring sufficient performance for operating systems projects. Once students complete their code, the compiled OS images are submitted from the

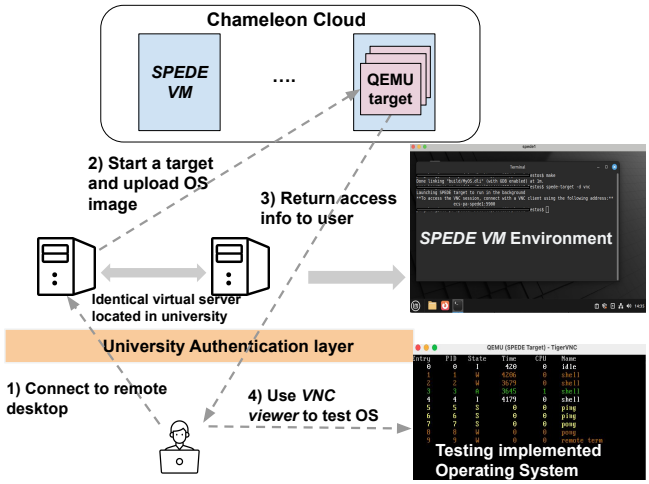


Figure 2: Hybrid Cloud Architecture for OS Labs

university servers using custom commands, which automatically deploy the images to a Chameleon VM. The testing process is managed by the VNC reviewer software, which provides feedback on verifying the target system. Importantly, Chameleon VMs are not accessible for direct student login or general-purpose use; they are reserved exclusively as controlled environments for automated operating system testing as shown in Fig. 2. All university servers and Chameleon VMs share the same configuration. When system workload is low, students can run their OS simulations directly on the university servers.

6 Future work & Conclusion

This paper addresses a critical challenge in operating systems education: ensuring that students can focus on mastering core concepts without being limited by hardware constraints or access barriers. Combining Chameleon Cloud and university infrastructures, we provide OS labs with improved scalability, reliability, and security. This hybrid approach also enhances accessibility and increases teaching efficiency. Student feedback will be collected to evaluate and improve the current work in the future semester. The University of Texas at San Antonio also plans to adopt this model in future offerings of its operating systems course or as a senior project.

References

[1] José Antonio González-Martínez, Miguel L. Bote-Lorenzo, Eduardo Gómez-Sánchez, and Rafael Cano-Parra. 2015. Cloud computing and education: A state-of-the-art survey. *Comput. Educ.* 80 (2015), 132–151.

[2] Kate Keahey, Jason Anderson, Zhuo Zhen, Pierre Riteau, Paul Ruth, Dan Stanzione, Mert Cevik, Jacob Colleran, Haryadi S. Gunawi, Cody Hammock, Joe Mambretti, Alexander Barnes, François Halbach, Alex Rocha, and Joe Stubbs. 2020. Lessons learned from the Chameleon testbed. In *Proceedings of the 2020 USENIX Conference on Usenix Annual Technical Conference (USENIX ATC'20)*. USENIX Association, USA, Article 15, 15 pages.

[3] Aditya Saligrama, Cody Ho, Benjamin Tripp, Michael Abbott, and Christos Kozyrakis. 2025. Teaching Cloud Infrastructure and Scalable Application Deployment in an Undergraduate Computer Science Program. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1* (Pittsburgh, PA, USA) *(SIGCSETS 2025)*. Association for Computing Machinery, New York, NY, USA, 1015–1021.