Teaching and learning now involves the use of computational tools via software applications in virtually every academic field. Twenty years ago, the primary delivery system of software and hardware was the campus computer lab. Gradually, as student computer ownership became common, many software applications began to migrate to the student’s own computer. Yet the general-use computer labs continued as the norm.

General-use computing labs on campuses command a significant percentage of academic IT budgets, both in materials and in labor expenditures. Given the current economic climate, this cost is driving intensive scrutiny of the role of and need for these labs. Student ownership is limited with respect to costly high-end software applications—those requiring high-end hardware or major technical support. For these applications, specialized campus labs will still be needed. But the general-function labs providing less-demanding applications come into question. The installation and upkeep costs, in combination with the current configuration of the labs (e.g., refreshing relatively powerful desktop computers on a three-to-four-year cycle), take dollars and human energy away from other academic pursuits.

Student ownership of computers does not eliminate all of the difficulties found in maintaining labs. Negotiating the hardware, license, and support issues with respect to the student-owned computers is costly. In addition, the heterogeneity of hardware and also of software environments (e.g., which Java Toolkit version is installed?) frequently leads to obstacles for instructors and students in the effective use of learning tools.

Cloud Computing

In cloud computing, the computer that provides the computational service comes to the user over the Internet. The applications are not installed on the user’s computer. This restructures many of the limitations experienced with the traditional computing lab and the student ownership models. The user needs only a modest-sized computer, while the faster processing speed and larger memory capacity required for advanced software can be provided via the cloud. Cloud computing delivers applications to any operating platform, so the religious wars between Windows, Macintosh, and Linux can be avoided. Many support issues simply go away because of the very great control over uniformity of the computing resource in the cloud.

Public clouds have begun receiving much notice. North Carolina State University (NC State), along with many other higher education institutions, has started to move student e-mail services into a public cloud environment. Private clouds have received less attention, but they are playing an increasingly prominent role in teaching and learning. The Virtual Computing Lab (VCL) is an example of private cloud computing at NC State. The VCL (http://vcl.ncsu.edu) has been in production for six years and serves 30,000-plus users and over 100,000 reservations per semester. It is an Apache Open Source incubator project. Images that integrate software applications, operating systems, and other utility and network-based access tools are loaded on blades—high-end personal computers clustered in a centralized system. The upkeep of the automated blade clusters takes less than one-tenth of the lab personnel’s time and effort that would be required to install and maintain hundreds of computing lab machines. Based on a dynamic architecture that allocates hardware resources on demand, the VCL serves course-related student computing requirements during academic semesters; in off-semester times, resources are switched to the high-performance computing (HPC) facilities needed for research by faculty and graduate students.

The VCL was generated out of a mounting crisis in several arenas beyond the rising costs of outfitting and maintaining campus computing labs. As the number of applications installed grew above fifty, the personnel time required to solve software and system conflicts led to unacceptable delays in the availability of software for courses. Central IT too often played a gatekeeper role in the face of creative faculty and students who wanted to expand—not contract—the diversity of computing tools in teach-
The Long Tail

Unprecedentedly lowering the cost and expanding the access of computational provisioning were clearly driving forces in the development and adoption of the VCL, but the focus here is on one of the unanticipated breakthroughs it enables in pedagogy: the “long tail of software.” This long tail mirrors, in many ways, the well-known “long tail” phenomenon of the Internet: the ability to profitably sell niche-market items, many of which are long off the charts, in relatively small quantities of each.²

Like its retail predecessor, the “long tail of software” phenomenon has emerged as a result of the very recent technological advances of cloud computing.¹ The essentially unlimited number of images that can be stored enables a correspondingly large number of applications to be available. The cost is minimal: approximately $2 per year for storing an image. Since each version of an application is stored in a separate image, multiple versions remain available to users via the VCL as long as they are desired. These capabilities enable the long tail of software. In contrast, multiple versions of the same application seldom can be retained on computer lab machines. Supporting fifty or more applications on one lab machine increases system-related problems (e.g., “dll wars”), placing limitations on the support of large numbers of courses and instructors.

What is the pedagogical ramification of the long tail of software? Software vendors develop and refine applications at a remarkable rate. Instructors often face software updates out of synch with their course preparations for the academic year. Assignments and learning materials based on an older version have to be changed if the institution’s lab machines are updated at the start or in the middle of a semester. The foundational academic construct—the diversity of methodological approaches, theoretical concepts, scholarly perspectives, teaching practices, and so on—now requires an equivalent flexibility in access to diverse software applications, including multiple versions. Conflicting needs of faculty—for example, whether to move quickly or slowly in upgrading software—should not privilege one group over the other, nor should they be arbitrarily resolved by vendor release dates or IT staff resources.

This long tail phenomenon became apparent at NC State when viewing the software usage in the VCL archives over a number of years. The use of older versions of applications has been considerably higher than anticipated. This effect can be seen in the use of two demanding applications in substantial use at NC State. The first is Maple, a heavily used mathematics application currently available in the VCL in three successive versions (as of February 2010). The newest version, Maple 13, was added in early summer 2009, and although the majority of Maple use is of this version, earlier versions still are used one-third of the time.

This continuing use of earlier versions is covered by the license for the current version, and so faculty and students are free to use whichever version they want, while costing the university only pennies in image storage and no additional license fees. The other example is MATLAB, another mathematical program. Five different versions of MATLAB are still in use. The cost for hardware, software, and personnel is the same whether or not access to all versions is allowed.

In Support of Pedagogy

The emergence of the long tail of software speaks to an enhancement of productivity and flexibility in higher education. As in the retail long tail landscape, a VCL-type cloud removes limitations on providing software. This frees institutions to realize the customization essential to education. Although we can envision cases in which instructors should be moving to newer software, we do not believe that these moves should be imposed by the IT staff.

Will commercial public clouds be willing and able to provide the diversity of software choices needed by higher education? The VCL private educational cloud is designed to do this—and at very low cost. The technological capacity may be present in commercial clouds but might not fit their bottom line.

The long tail of software also causes some licensing complexities that interact with software publishers’ concerns about cloud computing in general.³ These issues need to be resolved for the benefit of both the publishers, who want to increase their market, and the education community, which needs to utilize the great diversity of software available with the fewest impediments to pedagogy.

Affordable and convenient access to both cutting-edge and established learning tools is essential to the modern mission of higher education. Cloud computing’s unexpected impact on pedagogy is the long tail of software, enhancing the flexibility and scope of software applications in teaching and learning.

Notes


Sarah Stein (sstein@ncsu.edu) is Associate Professor in the Department of Communication and VCL Coordinator of Academic Outreach & Collaboration at North Carolina State University. Henry Schaffer (hes@ncsu.edu) is Professor Emeritus and Coordinator of Special IT Projects & Faculty Collaboration at North Carolina State University.

© 2010 Sarah Stein and Henry Schaffer. The text of this article is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License (<http://creativecommons.org/licenses/by-nc-nd/3.0/>).