

# Cutting the cord: virtual machines for real instrumental analysis not just at the instrument

Rosina M. Georgiadis<sup>1</sup> · Kristina Streu<sup>1</sup> · Norman C. Lee<sup>1</sup>

© Springer-Verlag GmbH Germany, part of Springer Nature 2018

This article describes how we virtualized instrumentation software in an advanced undergraduate chemistry laboratory course in fall 2016 as part of a pilot program at Boston University (BU).

For the first time, cloud-based software was integrated into the curriculum. This integration transformed the classroom and the laboratory experience and empowered students by allowing real learning about instrumental methods and data analysis to happen away from the instrument itself (i.e., “cutting the cord” between student analyst and instrument; see Fig. 1). This also eliminated a long-standing obstacle to teaching and learning in instrumental lab courses: the asynchronous round-robin scheduling of equipment [1]. It also created new self- and peer-learning opportunities.

We believe our approach can be adapted and scaled to any laboratory setting inside and outside of the academy in which analytical and bioanalytical instrumentation is used.

## How we did it: an introduction to virtual machines

Instrumentation software was virtualized using what are known as virtual machines (VMs) or virtual desktop infrastructure (VDI), also sometimes referred to as server-based or cloud computing.<sup>1</sup>

<sup>1</sup> Although technically not the same, we use terms VM and VDI interchangeably here to refer to what the user experiences, on their device, as a virtual desktop running with its own OS and software. That is, a computer within a computer. VDI utilizes server hardware to run desktop operating systems and application software inside a VM. This means that virtualized instrument software can be accessed on any device with an internet connection anywhere, anytime. Because the software is not installed on the user's device, its performance is independent of the user's computer resources (clock speed, memory, etc.). Each student has concurrent access to the same capabilities in the lab, in the classroom, or at home.

✉ Rosina M. Georgiadis  
rgeorgia@bu.edu

<sup>1</sup> Department of Chemistry, Boston University, Boston, MA 02215, USA

The user experience on a VM is *not* a simulation but rather an authentic experience that is identical to using the lab bench computer attached to the physical instrument. Unlike software simulators that mimic how an instrument works, the VM software serves to support and extend the utility of the specific laboratory instrument used in the course. VM software runs in what is called offline mode, in which the instrument is not controllable (that is, no data collection is possible) but all key software features such as the ability to create methods and analyze data are maintained.

What is transformative for our course is that all students and instructors have simultaneous access to fully functional VM instrumentation software from their own laptop or any mobile device that has internet access. This means that significant training can happen away from the instrument and that equipment use is highly efficient. For example, while the instrument is blocked off and being used for data collection by some students, others can work on related tasks using their own VMs rather than waiting for serial or round-robin access to instrument software.

## Where and how we did it: our course curriculum and resources

In our pilot, which we called the BU VM Project, we “cut the cord” and created instrument software VMs for seven different types of instruments used by 20 students and two instructors (lecture and laboratory). These VMs were fully integrated and used almost every week in our one-semester laboratory course on instrumental methods of analysis. The course has 24 h of dedicated pre-lab instruction in a computer classroom and 72 h of laboratory.

Our course is designed for undergraduate chemistry majors or other science students with two full years of chemistry (general and organic). Students are typically juniors or seniors with limited previous instrumentation experience, who enrol to “gain more experience and

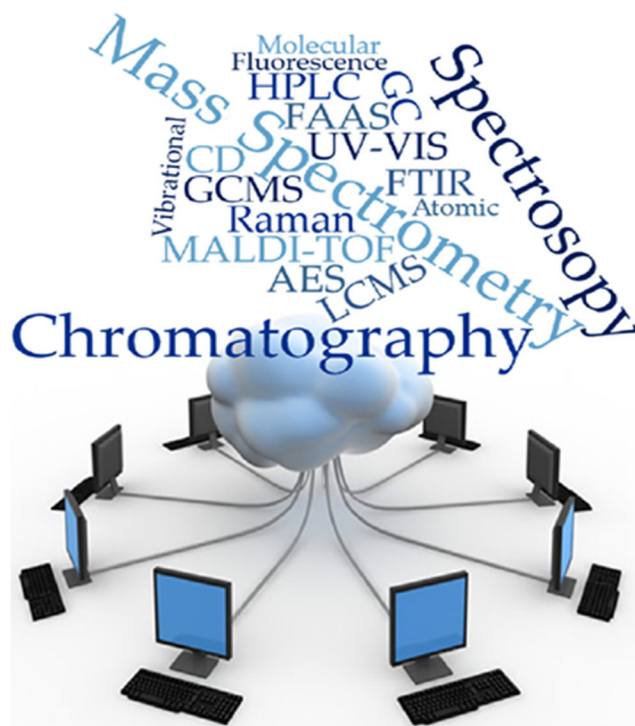
understanding of instrumentation” for their undergraduate research or postgraduate careers. The physical instruments (HPLC, GCMS, LCMS, AAS, AES, UV-VIS, and IR) are located in instructional laboratories in the Chemistry Department or within the Chemical Instrumentation Center at Boston University, a core facility that supports instruction and research.

In our course, the VM instrument software was used simultaneously by >20 students, but it could be scaled up to be used by hundreds or thousands of simultaneous users provided that the appropriate computational resources are available. For the BU VM project, we used two different approaches to virtualizing instrument software. These were developed at the same time, from conception to deployment, in a matter of weeks beginning in late summer 2016, with the assistance of IT technical support staff with specific VM/VDI knowledge. For specific details, see Appendix 1. Although there are other means of achieving virtualization,<sup>2</sup> we believe that the two methods used in this pilot are likely to be the most scalable to other laboratories and institutions.

## What we did: the impact of VMs

Using VMs was a game changer for our instrumental analysis course because students could access the VM software using their own computers while others used the instrument to collect data. This made it possible for all students in the lab section to perform the same experiment during a given laboratory period, even when there was only one physical instrument available. This capability was transformational for our curriculum because it circumvented the traditional round-robin random laboratory sequence and enabled self- and peer learning. As a result, VMs altered—and empowered—the

<sup>2</sup> An alternative model that should be mentioned is the use of open-source solutions such as VirtualBox. VirtualBox is completely free and can be used to create VMs of the necessary operating system (for example Windows XP or Windows 7) on student devices (Windows, Linux, Mac OS X laptops). The main disadvantage of this approach is the setup time and effort required from each user to create their own VM running locally on their device. While this could be very well suited to an advanced semester-long or year-long course that uses instrumentation, the time and technical support needed to ensure that each student sets up their device properly is impractical for large undergraduate courses. Professional IT staff would be needed to create and clone VMs and to help users install them on their personal devices. Software licensing would be tricky because there would be no expiration or control once installed on user devices. This option would not use cloud-based resources and would therefore depend on the user’s computer resources (clock speed, memory, etc.), although modern laptops would have no difficulty running most of the instrument software. However, the sheer variety of possible user devices and variability of user skill make this model daunting to all but the most technology-savvy users.



**Fig. 1** With virtualized laboratory instrumentation software, all students have access to instrumentation software on their own devices regardless of the operating system. This looks and feels identical to the software operating on the physical instrument and allows users to gain familiarity with setting-up methods, optional settings and configurations, help menus, and library functions, and can be used for data exploration and analysis. In our implementation, the software operates in offline mode and does not communicate with or control the physical instrument

way novice students interacted with their instrument: they no longer feared fumbling through its software or damaging the device itself, and they also gained 24/7 access to the software. These enhancements were important and measurable outcomes of the BU VM project.

For obvious reasons, the lab bench instrument computer is not the best place for students to learn how to explore their data, plan experiments, and create methods (software instructions, in our case) based on those plans. So we provided VMs which brought the lab computer to our students and created instructional scaffolding which leveraged both instrument resources and human resources (instructor expertise). Although a detailed discussion of specific teaching approaches that use VMs to enhance active learning is beyond the scope of this article and will be the subject of a later publication, we provide a synopsis and some examples of how we used VMs below.

Students first used VMs for group learning in pre-lab training sessions under the supervision of expert instructors, then again during the supervised laboratory session with their lab mates for off-instrument access to software and data, and finally at home for post-lab analysis.

More specifically, in the classroom during the pre-lab lecture, students first *explored* the software in an active way with instructor-guided questions that pointed out details of the software. For example, for spectrometric instruments, students were led to discover the interplay of instrumental parameters such as data collection interval, integration time, and scan rate; they also considered what the default settings were and what happened if they were changed. Instructors provided students with sample data and targeted questions or puzzles to guide their data exploration exercises (which included overlaying graphical data for comparison, performing peak-finding analysis, carrying out background subtraction analysis, and using library search algorithms). VM training (with good and bad data, as well as successful and unsuccessful data analysis) was interwoven with short lectures or problem sets to emphasize theoretical underpinnings and principles.

Students arrived in lab empowered to decide for themselves how data should be collected (full or limited scan, single point reads, etc.) and how to judge whether their data were adequate for the analysis goal. Rather than struggling with unfamiliar software or blindly following “cookbook” instructions, students helped each other to take charge. When not using the physical instrument, students used their personal VMs to develop expert-like practices; for example they were able to develop or modify their own experimental method (data acquisition parameter settings within the software instructions) first before getting on the instrument. Or, after they had some data, they could decide if they wanted to edit their method. Students thus developed the capability to assess their own experimental progress and to make their own decisions as to whether to return to the physical instrument to collect additional data with different method settings. During the laboratory, students worked together in small groups and often engaged in spontaneous peer teaching.

After lab, students could continue to perform all of the same operations, and they now had the ability to do this at home and throughout the time they were writing their independently authored lab reports.

As our work with VMs is still very new, we do not yet have robust data on specific educational gains, but we were able to compare and contrast the impact of VMs on student laboratory behavior and active learning gains. This was possible because approximately 60% of the instruments used for our course had available VMs while the remaining 40% of the instruments did not, and this required us to use more traditional training approaches in the latter cases. We found that the VM-enabled approaches were marked by pronounced differences in active learning activities during the classroom training sessions, and that they also appeared to correlate with better learning outcomes and distinct patterns of student behavior in the laboratory.

More specifically, for the experiments with available instrument software VMs, the classroom training time was spent having each student operate their own independent VM, so they were actively engaged throughout the period, with the professor and laboratory instructor circulating around the room. The VM software training sessions provided students with the time and space required to become familiar and comfortable with the instrumentation software. Instructor-guided exploration of the software provided the students with an opportunity to learn on their own and from their peers in what are called “think-pair-share” activities. Students were novices and made mistakes, but through trial and error, peer-to-peer discussion, and instructor feedback and guidance, they gained more confidence and retained more information. For experiments for which VMs were unavailable, we also used classroom time to prepare the students to operate the unfamiliar instrument software using more traditional methods such as presenting a slideshow of instrument software screenshots or short screen capture videos of an experienced professor or TA (teaching assistant) using the software. In this case, however, students were passive recipients of information and did not have the opportunity to explore the software on their own prior to arriving in the lab.

We noted different student attitudes and recall about their experience with instrumentation in the laboratory experiment setting depending on whether VMs were available. The most marked difference for non-VM-prepared students was that they tended to wait for the TA to help them operate the instrument, holding back from trying to navigate the software, often citing apprehension of breaking the instrument as the reason. Without VM preparation, students passively observed as the TA used the instrument software for them or guided the students through point and click directions in the software. Of course, when the TA was monopolized by one group, they could not guide other students working on various other portions of the experiment. When asked about their experience with the experiment, we found that students complained more, expressed frustration, and were more likely to focus on the limitations of the software.

In contrast, student groups with VM preparation typically did not hesitate to navigate the instrument software, and they performed the experiment autonomously. If they sought help, their questions pertained to a deeper understanding of the experiment. Indeed, we found that students often utilized the VMs on their own initiative during the laboratory period to explore the software, seeking alternative or more efficient ways to collect or analyze data. Often, there was sufficient time to re-run samples or edit instrument methods, thus avoiding the cookbook laboratory experience. In surveys and reflections on their experiences when VMs were

available, most student comments pertained to the experiment itself rather than the software, and were more positive in general.

In conclusion, one of the educational gains that distinguishes the VM approach is the purely experiential learning that stems from students having a chance to practice and experience low-stakes failure in a supportive environment, leading to more confidence and positive learning outcomes in the laboratory.

When appropriately integrated into the curriculum, we believe that VMs can make a meaningful and long-lasting educational impact and enhance how we teach instrumental methods to advanced students and professionals in the future.

## What you can do: scalability of VMs

Virtualization of software (cloud-based computing) is inherently scalable, so it is no surprise to see more and more cloud computing in daily life. The education sector is no different, and the potential uses of instrumentation software VMs extend beyond coursework to training and research.

While still in the early stages of our pilot in the instrumental analysis laboratory, we received inquiries and interest in implementing specific instrument VMs in other teaching laboratories, such as biochemistry and physical chemistry, and in some research laboratories which use instrumentation, both within BU and at other institutions. These inquiries were the initial impetus for this article. We have come to realize that, even in the absence of on-site physical instrumentation, VMs may be a very effective way for instructors and students at resource-limited institutions (two-year colleges or even some high schools, for example) to teach and learn valuable skills. If available, student-prepared samples could be transported or shipped to another institution that has analytical instrumentation where data could be collected, and the results could then be viewed and analyzed using VMs.

Instrument software VMs are also well suited to in-class or online courses for graduate and professional training. Users of core laboratory facilities would also benefit from VM implementation not only for off-site user access to data analysis tools but also for off-instrument training sessions and other multi-user training scenarios. At Boston University, for example, we have begun exploring the use of VM-enabled access to instrumentation software to train teaching fellows and graduate students who teach, use, or manage instruments.

Access to software and data off-instrument with VMs may also be very useful for laboratories around the world where

scientists continue to use legacy operating environments<sup>3</sup> for many or all of their instrumentation computers.

Appendix 2 contains technical details of several VM scalability models.

## What's next: plans and outlook

We plan to continue using VMs in the advanced instrumental methods course at Boston University. Given the success of the BU VM project, students and instructors alike have found it difficult to conceive of the course being taught any other way.

Discussion has begun to incorporate VMs into other analytical, bioanalytical chemistry, biology, and medical laboratories as well as into engineering laboratories (biological, chemical, materials, etc.) where instrumentation is used.

We believe that virtualized instrumentation software should be an integral part of the instructional resources of all laboratory courses, and ought to be developed and exploited in ways that improve student learning, skills, and creativity in solving chemistry problems with instrumentation. Instrument manufacturers and vendors who include software VMs with their instruments will distinguish themselves as leaders in instrument training, pedagogy, and accessibility.

**Acknowledgements** We thank the College of Arts and Science IT Department for providing VMs on their server, and we especially thank Brian Anderson and his staff, particularly Jacob Vulkovich, for technical support. We thank Agilent Technologies (Santa Clara) for their support, and especially thank Alexi Zubiria and Shawn Anderson for their technical support. We thank Steve Gagliardi for his willingness to initiate the pilot program in late summer 2016, and Jim Lynch for his continued support. RMG acknowledges a six-week Center for Teaching and Learning Faculty Fellowship (CTL FF) Award in the early summer of 2016, which afforded time and space to reflect on teaching and engage with other faculty. The VM project is the outgrowth of a longtime goal to improve student experiences in advanced laboratory courses, and we are grateful to the students and instructional staff of CH303 (Instrumental Methods of Analysis) since its inception as a new course in 2013. Finally, we thank Binyomin Abrams and Morton Hoffman, faculty colleagues at Boston University, for helpful discussions, and Brian Anderson and Andrei Lapets for technical discussions.

<sup>3</sup> Many laboratories continue to use Windows XP OS even though Microsoft ended support in 2014 because migration to another OS typically requires costly hardware upgrades, time-consuming transfers of settings, and user retraining. When instrumentation still works well, there is no compelling reason to upgrade to a new OS and the new application software that goes along with it. Virtualization extends the utility of older hardware because data is easily accessible on user laptops running the latest operating systems.



## Appendix 1. How we cut the cord: two ways

The BU VM project used two different methods of deploying VMs. In one method, VDIs were created by IT staff and deployed on university servers using VMware; we refer to these as BU VMs. In the second method, VDIs were created by Agilent Technologies for Agilent instruments especially for this pilot and deployed on Amazon Cloud (AWIS); we refer to these as Agilent VMs.

For BU VMs, BU College of Arts and Sciences (CAS) IT staff set up the VMs at the beginning of the fall 2016 semester using software licenses provided by instructional staff. Access was limited to registered students and instructional staff with valid Boston University login credentials. The VMs were available 24/7 throughout the semester except for short announced maintenance blocks of a few hours.

Each successful login produces an identical, as installed, desktop since configuration settings are not saved. However, student data can be uploaded, stored, and conveniently shared on shared folders and storage drives. To access BU VMs, each student first downloaded and installed a VMware Horizon client on their own device (Mac or PC) or any computer connected to the BU network; off-campus access required use of a VPN (virtual private network). When deployed, a new window opened on the user's device that was a functional Windows desktop with all of the necessary instrument or analysis software pre-installed by IT staff and previously tested by instructional staff.

The BU VM desktop contained icons to launch various Windows 7 applications such as instrumentation software for spectroscopy and chromatography, freeware simulators for HPLC and GCMS, a multi-user group license for Origin (OriginLab, Northampton, MA, USA), and a variety of other instructor-provided materials, including raw data sets for in-class activities and training.

External Agilent VMs were created for each instrument by Agilent staff and deployed on Amazon Cloud for testing by the instructional staff prior to use by students. VMs were configured with internet access so that data could be uploaded and downloaded from the desktop. They were set to time out after 30 min of inactivity and had a predetermined availability of 14 days or as long as needed for the experiment.

To access Agilent VMs, each student simply launched the VM as their own instance from a browser. For a specific instrument (for example UV-Vis), each student activated a pair of hyperlinks in sequence, the first to start the AWIS instance and the second to log in with credentials. Successful login launched a browser window that contained a desktop with the necessary instrument software installed, often with sample data and other resources (such as video tutorials) included. Instructor-provided data or student-generated data were easily

imported for use. While all software settings could be manipulated, each successful login produced an identical "startup" VM (i.e., it appeared as originally installed), since configuration settings were not saved.

## Appendix 2. Approaches to scaling VMs for educational use in the laboratory and classroom

For institutions with IT resources for hosting VMs (e.g., servers running VMware or an equivalent product such as institutional cloud access) and IT staff who can work directly with faculty, all that is needed for implementation is a copy of software that was obtained with instrument installation. Most vendors agree that access to software offline for educational purposes does not violate licensing provided that the institution has a valid software license.

Institutions or users who do not own the physical instrument may be able to negotiate the use of offline software licenses for teaching purposes. For institutions without dedicated or available server infrastructure, commercial and public cloud resources are available. Importantly, we note that even nominally free cloud computing requires ongoing local institutional/organizational IT resources dedicated to creating, deploying, and overseeing cloud-hosted VMs.

Instrument vendors can help, and there are a number of possible ways for them to help make instrument VMs available for educational and classroom use. Rather than having the institution create images, for instance, vendors can create images with the software pre-installed and available for institutional download on commercial or free cloud services. This may be preferable from the perspective of some educational institutions, especially if they already have IT resources that can be dedicated to the deployment and maintenance of virtual machines.

Alternatively, vendors can create Amazon Machine Images (AMIs) containing the software applications, and make these available in the Amazon Web Services Marketplace. This may become a product/service offered by the vendor that is offered to multiple institutions and could have different pricing models such as subscriptions to AMIs and per-session AMI usage fees. Or, the school or user could pay standard cloud usage fees that could be coupled with AWS Educate, which provides some free and some non-free services to universities.

## Reference

1. Arnaud C. New approaches to undergraduate lab classes. 2017. <http://pubs.acs.org/doi/10.1021/cen-09516-scitech2>



**Rosina Georgiadis** is an associate professor of chemistry at Boston University. Her research in experimental physical and bioanalytical chemistry has been funded by NSF, including an NSF Career Award, and NIH. At BU, she developed an entirely new advanced laboratory course and has been actively developing resources to help students who use (and instructors who teach with) research-caliber analytical instrumentation.



**Norman C. Lee** is the director of the Chemical Instrumentation Center (CIC) at Boston University, where some of the VM instrumentation is located. He actively participates in instructional activities and research projects with instructors and faculty, and has over 15 years of experience with a broad range of analytical instrumentation. The CIC is a core facility which has received instrumentation awards from NSF and NIH in recent years.



**Kristina Streu** is a postdoctoral faculty fellow in the Chemistry Department at Boston University. Her research interests include developing computational approaches to identifying macrocyclic inhibitors that target protein-protein interactions, along with her research faculty mentor Adrian Whitty. Since 2016, she has taught an instrumental methods of analysis laboratory course with her teaching faculty mentor Rosina Georgiadis. She also teaches introductory chemistry courses and participates in outreach with the Boston University Women in Chemistry organization.