# Streamlining Workstation Deployment and Configuration in an Academic Computing Environment

Shaun M. Lynch, Ph.D.
Department of Mathematics and Computer Science
University of Wisconsin-Superior
Superior, WI 54880
slynch@uwsuper.edu

## Abstract

Academic departments that consolidate several disciplines expect their dedicated computing labs to perform seamlessly despite the tasks placed upon them. Computers used for instruction in a multiuser environment must span a wide range of activities found in technology-related and analytic fields. Shared workstations often contain a whole host of applications that include productivity suites, development tools, desktop virtualization, utility applications, and antiquated legacy software. The Department of Mathematics and Computer Science at the University of Wisconsin-Superior streamlined workstation deployment and configuration by moving a majority of system configuration activities into the deployment phase and increasing the deployment rate. This was achieved by aligning the process to departmental operations and customizing a software application to automate repetitive activities. This paper presents the design and technologies used toward this goal along with evaluating tradeoffs between upfront and ongoing costs. The paper closes with thoughts and observations on the effort and gains realized.

# 1 Introduction

Academic departments that consolidate several disciplines expect their dedicated computing labs to perform seamlessly despite the tasks placed on them. Computers used for instruction in a multiuser environment must span a wide range of activities found in technology-related and analytic fields. Shared workstations often contain a whole host of applications that include productivity suites, development tools, desktop virtualization, a multitude of utility applications, and even antiquated legacy software used for niche applications. Offloading the process of deploying software used for classroom or lab activities to students and their personally owned computers is often infeasible due to inconsistent or underpowered hardware, licensing restrictions, and limited student technical ability to install and configure software. Shared computing labs provide students and instructors with a consistent user experience that is secure and reliable.

The Department of Mathematics and Computer Science at the University of Wisconsin-Superior recently replaced the computers in its two advanced computing laboratories, hardware lab, and learning lab. At the same time however, the department was at the crossroads of a personnel change due to turnover in the computer lab assistant position. This prompted a search for a new deployment platform to replace the existing system no longer considered capable of performing the task. The combine loss of expertise and availability of new tools offered ample motivation to automate image deployment and workstation configuration. The plan moving forward was to streamline the entire process, improve consistency, and reduce the manual intervention necessary to make the systems production ready.

This paper presents the design, technologies, and methods used to achieve this goal. Discussion begins with a brief background and history of the department's computing infrastructure and the circumstances leading to the decision to streamline the deployment and configuration process. Next, the author looks specifically at the challenges associated with workstation deployment and configuration along with identifying the tools considered for the task and the factors leading to the decision to use the deployment application selected. Special emphasis is placed on deployment strategies, imaging technologies, and the ancillary systems needed to configure workstations across shared computing resources. Finally, steps taken to customize the technology suited for a rapid deployment system are discussed along with efforts to tailor the process to take advantage of the unique conditions of an academic setting. The paper closes with commentary on the approach and inherent managerial challenges associated with workstation deployment and configuration in an academic computing environment necessary to suite a wide array a needs for a diverse set of users.

# 2 Background and History

The Department of Mathematics and Computer Science began hosting its own computing lab around 2002 as a dedicated computer facility consisting of 20 workstations and handful of servers. Called the Development Lab, nearly every aspect of the lab's operations was

the result of manual effort to deploy, configure, and operate the systems necessary to provide the required functionality. The lab often had two lab managers working between 10-20 hours a week to ensure the lab remained in working order. Each semester brought a flurry of activity to install new operating systems and software applications, configure individual workstations, and join computers to the Active Directory environment.

Beginning in 2008, conditions in the Development Lab began to deteriorate. Two factors led to this situation: First, the principal architect of the lab pursued another employment opportunity resulting in a loss of leadership and vision for the lab. Second, workstation computers were gradually replaced in batches due to budget constraints creating a hodgepodge of models and hardware with little consistency across systems. Several talented students serving as lab managers were able to keep the lab operating during this period despite the challenging conditions, however, a consistent deployment strategy failed emerged to systematically configure the lab computers.

In 2011, the Department of Mathematics and Computer Science made a long overdue move to the new academic building, Swenson Hall. The transition provided the department with improved teaching facilities plus an opportunity reenvision its technology infrastructure and computing lab spaces to better serve the department's programs in mathematics, computer science, and information technology. As a result, the number of workstations increased by 250% to 50 computer workstations divided equally into two advanced computing labs along with a greatly expanded server infrastructure to support the shared computing facilities.

The change to the new facility was so extensive that many of the old practices from the Development Lab no longer applied after the transition and were simply abandoned. Initially, this created a deployment conundrum since a totally new approach was needed in short order. Without a budget to purchase licenses for the campus computer imaging system used by the IT department, the administrators of the department's computing infrastructure settled on the open source application called FOG (Free Open-source Ghosting) to capture and deploy computer images for the two advanced computing labs.

Reference images were prepared manually for each lab on virtual machines hosted in VMware Workstation. The image included a default installation of the operating system plus a full complement of software applications common to both labs along with specific programs designated for each individual lab. Images were captured using FOG and prepared for network deployment. Lab computers were then manually configured for PXE (Preboot eXecution Environment) boot to install the designated image. Once the image was installed and the computer joined to the domain, each workstation received a final visit by lab personnel to manually install remaining drivers, and configure hardware and desktop settings.

Like many organizations, a decision was made not to upgrade Microsoft Windows 7 to version 8 due to vexing interface changes that left many users confused and frustrated. This meant maintaining the current image as long as possible and making incremental changes until a better alternative was available. Many of the post-deployment activities, including

updating existing software, installing new programs, removing programs, and applying manual settings were accomplished by students serving as computer lab assistants.

As time went on, the author found it increasing difficult to attract students with the skills and interest necessary to perform as computer lab assistants. Budget cuts exacerbated this situation further reducing the pool of students available. This led to increased reliance on automation and scripting tools to manage the configuration of the lab computers. The author first adopted PowerShell 2.0 to offset personnel shortages and offload some of the more routine tasks to free lab assistant time for more complex issues. Although there were definite limitations during the early versions of PowerShell, the ability to centrally manage and automate processes greatly extended the available resources.

In the years leading up to 2016, computers in two additional labs were included into the mix of systems needing ongoing configuration and management. First, the Hardware Lab was specifically designed for developing microcontroller-based applications and devices. The lab contained two computer workstations that contained software programs needed to design these systems and interface with external peripherals such as a microscope with digital camera, digital oscilloscope, and logic analyzer. Second, the Learning Lab was created to provide students and faculty with advanced technology needed for custom projects. The lab contained two workstation-class computers equipped with high-speed solid-state storage and high-end graphic cards. Unlike the computers in the advanced computing labs, these systems were configured manually.

# 3 Rethinking the Process

Several circumstances precipitated the need to rethink and streamline the deployment and configuration process used to manage the department's increasing number of workstations. First, the tools and technologies used for the initial deployment were no longer sufficient for the task. Although FOG worked sufficiently well as a mechanism to capture and deploy images, the application could at times be difficult to use as a deployment platform. It also lacks essential pre- and post-deployment tools that addressed advanced deployment techniques beginning with Windows 8 and becoming standard practice in Windows 10.

Second, the procedure needed to create a reference image was manually intensive, time consuming, prone to errors, and difficult to debug. Every reference image needed to be created from scratch regardless of the number of common elements between them. Installing and configuring the operating system, productivity applications, and software utilities along with applying updates required considerable attention to a vast number of details and settings making it difficult to replicate accurately. Furthermore, problems encountered during the procedure were often elusive and difficult to pin down. It was not uncommon to repeat the entire procedure multiple times to debug errors and resolve deployment issues. Every new image needed only exacerbated the challenges involved.

Third, insufficient tools were available to manage post-imaging tasks. The department uses a number of applications that are not suited for imaging. For instance, applications that create virtual network devices like VMware Workstation or Npcap often do not survive a

system initialization using the *sysprep* command. Other applications that have per instance licensing requirements such as IBM SPSS or Adobe Creative Cloud and rely on specific hardware or software settings on the physical computer. These applications can only be installed on each machine after the image has been applied.

Fourth, scheduled computer replacements and an expanding number of computers in departmental labs accelerated the need to prepare computers in bulk. A surge of new computers can quickly overwhelm the resources of a lightly staffed operation. The new systems had significantly different hardware and there was no way to migrate the old image. In addition, increasing demand for new programs and dissimilar software arrangements across labs only complicated matters. The new computers acquired by the department simply needed an image just to be functional.

Fifth, an extremely talented student serving as the computer lab assistant graduated creating an expertise vacuum. It is rare to find a student willing to serve as computer lab assistants that has a real interest and zest for working with microcomputer systems that holds an equally advanced skill set. The department was fortunate to have one such student that was able to resolve difficult configuration and deployment problems independently. The student was employed as the system evolved so the individual was quite familiar with the nuances of the infrastructure and what it took to maintain the system and keep it operational. After working as a lab assistant for three years however, the student graduated leaving a significant knowledge gap that successors would have a difficult time closing.

Many of these issues and circumstances where known well in advance and preparations were being made to address them. The challenge lay in timing and ability to rally the resources needed along with the willingness to enact a major change to a fundamental process used to support the department's computing infrastructure.

# 4 Alternative Evaluation and Selection

There is a wide range of deployment and configuration technologies available; therefore, it is important to consider the particular platform and services currently being used as well as scope and integration. The department's computing platform is primarily built around the Microsoft line of operating systems and services due to the availability of products through the ELMS/Microsoft Imagine academic licensing program. Therefore, it is natural to look for options specifically tailored for the ecosystem that can take advantage of the current services already used within the infrastructure.

The department's computing infrastructure is relatively modest and consists of roughly two dozen servers and 50 plus workstations putting it on par with a small to midsized business. Microsoft Active Directory and Directory Services (ADDS) provides authentication and directory services for the site and Group Policy is used to manage the site's policy settings. In addition, Windows Server Update Services (WSUS) provides a local mechanism for controlling and distributing software updates and patches to Windows-based systems.

Although there are a number of enterprise-grade configuration management applications available such as Puppet and Chef that work across a number of different platforms, the decision focused on whether or not to use Microsoft's fully integrated enterprise configuration management application System Center Configuration Manager (SCCM), or their individual applications and services such as Microsoft Deployment Toolkit (MDT) and Windows Deployment Services (WDS) that are tailored for specific deployment tasks. The evaluation of each option began during the summer of 2014 when the author created an isolated sandbox on a separate virtual host that contained the essential components necessary to prototype the two systems.

After experimenting with the two alternatives, the advantages and disadvantages of each solution became clearer. The advantage of using SCCM is that all the components necessary for deploying and configuring systems are integrated in one system that has many capabilities. The application combines a number of constituent services (MDT, WSUS, WDS, MSSQL, etc.) under one framework that can be scaled to match the needs of the organization. Client systems interact with SCCM through a specialized client-side application that monitors the state of the computer. Although setting up the servers is somewhat complicated, it is doable provided clearly articulated instructions are available. The primary downside of SCCM is the complexity introduced by integrating so many components. An administrator must possess deep knowledge of the various subsystems to operate and utilize SCCM effectively. This means a steep learning curve and a formidable barrier to entry for individuals not fully acquainted with the technologies involved.

Alternatively, the MDT/WDS application-service combo provides an incremental, piece-wise deployment solution. Although MDT does not include all the capabilities SCCM provides, it is still quite functional as a deployment tool and can be customized. The learning curve is fairly modest for basic implementations but can increase quickly should any customization to the application be necessary. MDT uses task sequences to execute pre- and post-imaging activities thus spanning the full deployment process. Unfortunately, there is no real provision to manage the configuration of a client system after completing the deployment phase so other technologies must be utilized.

The crux of the decision to use a fully integrated or piece-wise solution lay in the process itself and whether it could be manipulated to favor one alternative over another. If the process continued as it had emphasizing infrequent deployments and long-term configuration management then SCCM would be the clear choice moving forward. However, substantially increasing the deployment rate would likely eliminate most of the configuration management activities needed between deployments. In this scenario, the MDT/WDS solution would be the preferred due its limited scope and manageable learning curve. Overall, the decision hinge on the ability to reduce and manage the upfront costs associated with deployment versus the ongoing costs of configuration management.

Based on the characteristics of the two alternatives, the author opted for a piece-wise solution using the MDT/WDS combo instead of the fully integrated SCCM application. Several factors led to the decision: First, implementing MDT/WDS is much simpler and does not require as many computational resources. Second, a piece-wise solution addresses the specific problem and keeps the scope manageable. Third, the learning curve is more

gradual for MDT/WDS. Fourth, a shorter deployment schedule based on semester or academic year boundaries aligns with the author's work schedule. Fifth, MDT can be customized to provide additional functionality as needed. Sixth, technologies already in place would be more than sufficient to cover a reduced configuration management workload. Seventh, the solution could always be scaled up and incorporated into a fully integrated option such as SCCM.

# 5 Implementing a Rapid Deployment Process

Implementing the rapid deployment model using the MDT/WDS alternative occurred in two phases. The initial phase was out of sheer necessity to get the department's advanced computing laboratories ready for the academic year. Replacement computers arrived and were installed during the summer 2016, however, the new deployment system was not yet operational leading to delays preparing the workstations for use in the computer labs. A considerable amount of effort was expended fine-tuning the image building process. At the time, Windows 10 was still fairly new with few resources available that addressed the more detailed and nuanced deployment issues. In addition, a lot of time was spent investigating various deployment strategies and aligning them with this particular implementation. Questions such as:

- What applications should go on the image and which should be installed at the time of deployment?
- How should legacy applications that require manual installations or have unsigned drivers be dealt with?
- What steps need to be taken to ensure the image remains consistent once deployed in a general access lab setting?

Answering questions like these and many others meant a lot of experimentation and trial and error to work out problems and produce a procedure that could be replicated. By November of 2016 however, the lab workstations were finally imaged and ready for use.

The next phase began in earnest during the winter break between Fall Semester 2016 and Spring Semester 2017 with the goal of streamlining and automating the image preparation process. Deploying an image is normally a four-step procedure that entails: 1) creating a reference system, 2) customizing the reference system, 3) capturing the reference image, and 4) deploying the reference image to the target computers. Normally, this four-step procedure is repeated for each image required and is configured using the MDT application *Deployment Workbench*.

Deployment Workbench controls the entire deployment process using one or more task sequences. A *task sequence* defines the all steps necessary to load, customize, and capture an image. Deployment Workbench also has the capability to identify target computers through a variety of means (i.e. serial number, MAC address, etc.) and associating that information with configuration settings in a database. Unfortunately, the scripts used to control the process were hard coded to work with only one reference image at a time.

A common technique used to overcome this limitation is to install the applications as a state restore activity (post-imaging) in the task sequence. Unfortunately, there are several programs the department uses that cannot be installed silently meaning that each workstation would have to be attended to complete the deployment. Alternately, a unique image could be created for each configuration. Building each image from scratch is time intensive and would entail repeating the same procedure over and over leading to duplicated effort. Therefore, neither of these traditional approaches solve this problem effectively.

In the meantime, the author was exploring an alternate approach that entailed creating a hierarchy of images by taking advantage of the multi-image feature of Windows Image files. Unlike traditional sector-based images, Windows Image files are file-based and store a single instance of any particular file thus offers a good option for storing differential images. Individual images can be appended to an image file and referenced by index or name. Unfortunately, MDT does not support this capability by default so the ability to specify an image to apply or name a captured image would require customization.

Approaching the problem in this manner offers a number of advantages. For instance, the process of building a reference image could be divided into stages. Each stage could be used as a base-image for any number of subsequent stages and debugged independently. Images could also build upon prior stages thus reducing duplicate effort by minimizing the number of times an application needed to be installed. And, specific images could be associated with one or more computers at the time of deployment allowing the deployment process to be consolidated and optimized for automation.

Implementing this capability required modifying the *apply-image* and *capture-image* portions of MDT's deployment process. Each change required: 1) declaring and initializing properties, 2) create or modify wizard panes to enter settings, and 3) modifying scripts to implement the functionality. In each case, the default settings could be used, entered manually using the wizard interface, or defined by setting the appropriate properties in the task sequence or database.

Once the changes were made to MDT, the deployment process was divided into two distinct phases. The first phase involved the construction of the reference image. A task sequence was created for each node in the image hierarchy. The *apply-image* and *capture-image* properties were assigned in each task sequence to prevent entry errors. A bare metal virtual machine served as the reference system. Booting the reference system automatically launches MDT and prompts the user to select the desired task sequence to execute. After the reference system was customized, the image was captured and appended to the reference image.

The second phase, called *production*, revolves around the actual deployment of images and software to individual computer systems. A new task sequence was created specifically for production that would query the identity of the computer and match it to a specific configuration using a database. Production begins by restarting the computer and initiating a PXE boot. The PXE boot environment will launch MDT, query the computer's identity, retrieve configuration settings, load drivers, apply the image, install software applications,

join the computer to the domain, and reboot the system. The procedure is fully automated and completely touch-free once the computer boots into the PXE environment.

# 6 Closing Thoughts and Observations

Deployment and workstation configuration is arguably not the most glamorous topic in information technology. Deployment and configuration management is an extremely time-intensive, detail-oriented exercise even after nearly four decades of personal computing. However, ensuring students and faculty have access to secure and functional workstations and receive a consistent experience is an essential part of administering a computing infrastructure in an academic department that absolutely relies on computing technology.

Moving system configuration into the deployment phase and increasing the deployment rate has had an immediate impact on operations. Prior to the conversion, the consistency across lab computers suffered from entropy and continually drifted from the standard. This situation provided computer lab assistants with a steady stream of work over the course of the academic year. Now, the bulk of the work occurs before the semester even begins and the computers rarely need any maintenance to keep them configured. Lab assistants rarely interact with individual workstations anymore except in cases where there is a hardware problem or something unusual is going on. A dramatic change from where things began.

This paper is all about managing the tradeoffs between upfront and ongoing costs and closely scrutinizing processes and technologies to wring out inefficiencies that sap already limited resources. This is achieved by considering the history and circumstances leading to the problem, articulating factors that necessitated a change, redesigning processes to align with departmental operations, and customizing a software application to enable the process. These steps lay the groundwork for automation to help streamline workstation deployment and configuration thus improving the delivery of technology services to students and faculty in an academic setting.