Virtualization in the Academic Computing Infrastructure: The Rise of Boutique Computing

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

Abstract

Boutique computing is the ability to offer specialized computing capabilities within a mainstream computing infrastructure. The concept emphasizes customization and flexibility without having to abandon the benefits associated with standardization and economies of scale. The author has been pursuing this vision of boutique computing over the past several years to support the academic programs of the Department of Mathematics and Computer Science at that University of Wisconsin-Superior. In an academic environment where application proliferation, demand for new technologies, and improved instructional methods abound, the approach draws upon a number of virtualization technologies to dynamically recast computer hardware into different roles and returning it to its original state with minimal intervention and management oversight. This paper presents the concept of boutique computing and applicable technologies along with examples, deployment issues, administrative concerns, and design considerations.

1 Introduction

The author defines boutique computing as the ability to offer specialized computing capabilities within a mainstream computing infrastructure. The concept emphasizes customization and flexibility without having to abandon the benefits associated with standardization and economies of scale. The approach draws upon a number of virtualization technologies that allow existing hardware resources to be dynamically recast into different roles and then returned to their original state with minimal intervention and management oversight. It is a technique suited for multiuser environments that rely on computer systems to fulfill their mission.

The author has been pursuing this vision of boutique computing over the past several years to support the academic programs of the Department of Mathematics and Computer Science at that University of Wisconsin-Superior. A number of technologies are currently deployed such as platform virtualization, virtual desktop infrastructure, and user state virtualization to maximize the hardware and software investments that serve over 200 students and ten faculty and staff members per semester in order to achieve the educational goals and academic mission of the department.

This paper presents the concept of boutique computing and applicable technologies along with examples, deployment issues, administrative concerns, and design considerations. Discussion begins with a brief overview of the circumstances and conditions leading to the need for boutique computing and what the technology would look like in an ideal world. Next, the paper details the different virtualization technologies used to implement boutique computing in the department. Finally, the paper closes with commentary on the approach and inherent challenges of developing the technical expertise needed to host such a system.

2 Background and Circumstances

The department manages and maintains a dedicated computing infrastructure primarily used to advance the department's teaching mission. Computing resources are used extensively in classes across the department's academic programs ranging from entry level coursework that fulfills the University's mathematics general education requirement to advanced electives that meet curricular objectives. Students, faculty, and academic staff enjoy a wide array of applications suited for general productivity, program development, analytic modeling, device emulation, network security, multimedia generation, and web authoring.

Two advanced computing laboratories are the most visible aspect of the department's computing infrastructure. Each advanced computing lab contains 25 computers with one set aside as a teacher workstation that drives the rooms projection and multimedia system. All workstations share a common configuration based on a core image containing Windows 7 x64 Enterprise as the operating system along with a productivity suite, integrated development tools, and a variety of utility applications. Specialty applications are licensed and installed on all computers within a given lab to ensure consistency and allow full utilization of lab workstations.

The department also maintains a “sandbox” for special projects, experimentation, prototyping, and demonstration that includes a variety of legacy computers, one-of-a-kind systems, components, networking gear, and mobile racks. The sandbox provides a setting where students and faculty can explore without tying up shared resources needed for instruction or disrupting the production systems.

The server and network resources are the cornerstone of the department’s computing infrastructure and provide essential services to host applications, manage computing assets, and deploy security policies. The physical system consists of two servers that host domain and internet protocol management services, a two node cluster along with an iSCSI storage array that provides a virtualization platform for 14 application servers, a dedicated server that hosts virtual desktops, and other specialty servers that provide shared storage and IPv6 connectivity.

2.1 The Need for Boutique Computing

The concept of boutique computing arose from a need to create customized computing solutions within a standardized computing infrastructure. Any number of factors can precipitate the need for a custom solution in an academic environment; however, they generally fall into one of two categories. First, an application falls outside the normal deployment model. For example,

- An application may have a restrictive license agreement that limits deployment
- An application may be constrained to a certain number of instances
- An instructor would like to use an operating system other than the one installed
- An individual would like to install a program for a special project or assignment
- A new version of an application is released outside the normal installation period
- A limited group of users need access to a particular application
- An application has special installation requirements or is ill-behaved in some way

Second, one or more users need special access to computer resources beyond what is supported. For instance,

- A user needs to install one or more applications
- A user needs to customize the computer to fit their needs
- A user needs full access to computer settings requiring administrative privileges
- A user needs a system that can be easily reset, replaced, or disposed of

However, there is another reason that drives the need for boutique computing—simplified system administration. There is no limit to the number of capabilities and features that can be added to an academic computing infrastructure. Application proliferation, demand for new technologies, and improved instructional methods conspire to overwhelm the personnel responsible for managing the system and ensuring that it remains operational.

In this context, boutique computing is a management philosophy that helps distinguish core functionality from specialized functionality. By identifying the core functions of the systems, administrators can structure the services needed to maintain the system and enact them in an ongoing manner. Specialized functionality can then be encapsulated and deployed as custom solutions eliminating the need to maintain them while not in use.

2.2 Boutique Computing in an Ideal World

What would boutique computing look like in an ideal world? From a user's perspective, the custom functionality boutique computing offers should be 1) readily apparent and easy to access, 2) consistent regardless of the mode in which is deployed, and 3) connect seamlessly to individual files and setting and shared network resources. Mechanisms that would facilitate this experience might include:

- Recognizable icons or dynamic tiles that identify custom solutions
- File management functions linked to the individual's profile
- Similar tools in which to navigate for network resources
- Minimal transition times from one custom solution to the next

From a system administrator's point of view, boutique computing would 1) define boundaries between core and specialized functionality of the system, 2) allow specialized functionality to be packaged in self-contained modules, and 3) be easily deployed across multiple systems and workstations. Characteristics that enhance manageability include:

- The ability to copy custom modules from a shared folder
- The ability to remove, reset, or replace configurations quickly
- The ability to uninstall or remove a configuration cleanly
- The ability to delegate responsibility for the custom solution to others

Combining these two perspectives would clearly be a challenge, however, the concept of boutique computing promotes a truly universal machine that transforms depending on the task it is assigned.

3 Enabling Boutique Computing

Virtualization is the key technology that enables boutique computing in an academic environment that deploys and maintains shared resources. Virtualization separates the logical applications from the physical implementation creating new degrees of freedom allowing hardware resources to be recast in multiple roles. In addition, virtualization provides system administrators with new tools such as snapshots, thin provisioning, and migration to effectively and efficiently manage the computing infrastructure.

This section presents the different virtualization technologies used by the department to implement boutique computing within its existing infrastructure. A brief description of each technology is provided followed by how the technology was deployed. Emphasis is placed on design and performance issues along with observed caveats. Examples are provided as available from actual classroom setups, sandbox configurations, and virtual desktop deployments to demonstrate progress toward the vision.

3.1 Platform Virtualization

Platform virtualization is a technology used to create and host virtual machines that imitate the physical hardware of a computer. Two types of platform virtualization technologies are available based on the type of hypervisor used. Type I hypervisors serve as the base operating system of a computer and interact directly with the hardware using its own suite of device drivers. Type I hypervisors are normally used in an enterprise settings to create a virtualization platform to host application servers. Microsoft Hyper-V is the preferred virtualization platform for hosting the department's application servers and virtual desktops.

Type II hypervisors on the other hand, run within the host's operating system and interact with the underlying hardware through the operating system's suite of drivers. Applications that incorporate Type II hypervisors are typically installed on individual workstations to create an environment suitable for building and running virtual machines. VMware Workstation is the supported virtualization application installed on the shared workstations in the department's computing infrastructure. Particular attention is given to this platform virtualization technology since it offers the greatest potential to transform a homogeneous pool of shared workstations into a heterogeneous computing resource that can be configured to host any number of operating systems and applications.

Implementing platform virtualization on the workstation revolves around two basic scenarios. The first scenario is organized around a self-service model where prebuilt virtual machines are available for download to an individual's profile. The author currently maintains a virtual machine library that includes desktop and server installs of a variety of Linux distributions, Windows, and even MS-DOS. Of course, a user could build their own custom virtual machine using any number of operating systems or applications available on department file shares or internet.

The second scenario is based on a kiosk model that allows users to select a virtual machine from a pool of locally installed virtual machines. This concept is currently being evaluated by the author and is a work in progress, but includes provisions for a menu to select hosted virtual machines and automatic virtual machine management. The virtual machines would be deployed by administrative personnel that contain a package of applications suitable for specific activities. Virtual machines could be configured to run in isolation or joined to the local domain with access to shared network resources.

Platform virtualization on the workstation offers a wide range of options for academic departments wishing to implement a boutique computing infrastructure. There a number of advantages associated with this approach that includes:

- Systematic deployment of virtual machines across a pool of shared workstations

- The ability to utilize the full computing power of the workstation
- The ability to incorporate tools included in platform virtualization software (snapshots, synthetic drivers, virtual switches, etc.) in the infrastructure
- The ability to increase computing capabilities and instructional options
- The ability to retrain centralized administrative and management functions

From an administrative perspective, platform virtualization on the workstation has the potential to significantly increase the demands placed on the computing infrastructure. Table 1 itemizes the resources used to create a virtual machine hosting various guest operating systems using VMware Workstation 10.[1]

Guest Operating System	Total Disk Space (GB)	Total Disk Write Ops (x1,000)	Total Disk Write Bytes (GB)	Total Disk Read Ops (x1,000)	Total Disk Read Bytes (GB)
Ubuntu 12.04 LTS Server	1.9	98.2	2.49	16.3	0.29
Ubuntu 12.04 LTS Desktop	4.4	124.0	4.23	39.5	0.86
Windows 2012 R2 Core	8.4	108.5	6.03	31.0	1.29
Windows 2012 R2 GUI	12.6	211.3	9.38	33.9	1.38
Windows 8.1 x64 Enterprise	13.6	195.2	11.00	115.1	3.60

Table 1: Estimated resources use for various virtual machine configurations

Shared storage systems are particularly susceptible to performance degradation if sufficient resources are not provisioned. To begin with, it is conceivable that an individual's profile could grow excessively if a user does not actively police the number of virtual machines installed. This situation could lead to the premature consumption of available free space if left unchecked. Fortunately, storage quotas and data deduplication allow administrators to monitor and maximize available storage resources.

Next, running virtual machines from an individual user profile will dramatically increase the number of input-output requests a networked storage device must process. Building a virtual machine and installing updates are extremely disk intensive activities and can overwhelm both throughput and input-output requests leading to significant performance degradation and delays. Insufficient network capacity can compound storage problems by introducing congestion thus impacting virtual machine performance on the workstation.

Finally, administrators must exercise extra vigilance to monitor domain services and internet protocol management systems when deploying virtual machines across a number of shared workstations. Plans should be made to manage active directory objects assigned to virtual machines within the host domain. In addition, a pool of addresses must be provisioned for virtual machines connecting directly to the infrastructure's network. Computing infrastructures that reside within private IPv4 networks have the potential to exhaust assigned address pools quickly as virtual machines proliferate.

[1] Results were generated using Microsoft Performance Monitor during a typical install of each operating system. Samples were taken every second over the duration of the installation from initial boot to presentation of the first login screen. Values listed are estimates for comparison purposes only and are not intended as performance benchmarks.

3.2 Application Virtualization

Application virtualization is a technology that hosts a program on a remote server that is normally installed on a client workstation. From the user's perspective however, using the application appears as if it were running on the client's physical computer. This technology is particularly useful for deployments targeting specific groups of users or for applications that have a limited number of licenses that cannot be fully deployed to the entire pool of shared workstations.

During Fall Semester 2013, the author launched a pilot program to deploy course-specific applications needed by instructors and students enrolled in two classes. A collection of virtual applications was created using Microsoft's Remote Desktop Service architecture that included separate web access, connection broker, and session host servers. The shared applications were installed on a remote desktop session host and published to a security group containing approximately 20 class participants. Authorized users could connect to the service through the remote desktop website and select the program to run.

Other than an initial dialog box requesting permission to use local resources, the remote applications appeared to run as if installed locally. File paths of *Open* and *Save As* dialog boxes automatically defaulted to the user's profile providing a seamless experience. The only differences noticed by the author were missing file associations and the inability to launch the remote application when files were viewed locally. However, it appears that further configuration refinements can resolve these issues.

From the author's perspective, application virtualization greatly reduces the effort needed to deploy and administer programs within the computing infrastructure. Specifically, centralized command and control of the application provides:

- The ability to install the program once on a shared server resource versus having to install the program on multiple workstations in a shared pool
- The ability to place the application online or take it offline as needed
- The ability to control user access and the number of instances available
- Simplification of common administrative tasks that include patching, modifying configurations, and installing updated versions

However, there are drawbacks that should be considered before deploying application virtualization that include:

- Involved, one-time setups that entail configuring and deploying multiple servers
- Increased reliance and demands on server and networking infrastructure
- The need to manage software licensing associated with Microsoft's Remote Desktop Services in addition to the license agreements of hosted applications

Overall, the author received positive feedback from the instructors and a decision was made to extend the pilot to another class beginning Spring Semester 2014.

3.3 Desktop Virtualization

Desktop virtualization technology provides a user experience normally associated with a traditional client system without the operating system actually being installed on the client's physical computer. Instead, the operating system is installed on a virtual machine that resides on a remote server and accessed via network using a browser or remote desktop application on the client system.

The author's first attempt to deploy desktop virtualization began as an experiment for a department colleague teaching a small class of students needing individualized computers during Fall Semester 2012. A Dell Precision T3500 running an instance of VMware ESXi was used as the virtualization platform and class participants were assigned to one of four virtual desktops cloned from a Windows 7 virtual machine template. The virtual desktops were accessed directly using Microsoft's Remote Desktop Connection from computers in either of the two advanced computing labs.

Although robust enough to handle the tasks performed by the students and instructor, the hardware used for the virtualization platform proved to be significantly underpowered. Installing applications, running Windows Update, powering up, and shutting down led to considerable delays as the number of input-output requests saturated the local hard drive. Nonetheless, recognizing the potential for this type of deployment stimulated the purchase of a Dell PowerEdge 420 server explicitly configured to host virtual desktops for at least 25 users as the 2012-13 academic year came to a close.

The author originally planned to continue using VMware's product suite; however, plans changed as difficulties arose trying to acquire ESXi version updates and Horizon View through VMware's academic licensing program VMAP. Instead, attention turned to Microsoft's recently updated Remote Desktop Services (RDS) released with Windows Server 2012. The required features were readily available since the department had just upgraded its server infrastructure to Windows Server 2012.

Fortunately, application and desktop virtualization using Microsoft RDS share a common configuration thus simplifying the steps necessary to deploy a desktop virtualization infrastructure (VDI). Adding the new virtual desktop server to the RDS infrastructure entailed installing Windows Server 2012, adding the Hyper-V role, and running the RDS wizard in Server Manager to configure the virtualization host. From there, virtual desktop collections could be created and deployed to various users and security groups.

The setup proved timely in that another colleague teaching an assembly language class wanted to demonstrate concepts using an evaluation version of microcontroller emulator. Unfortunately, the software license did not have provisions for academic lab installs; however, individuals could register with the company and receive a license to use the application. To resolve this dilemma, the author created a pool of 18 virtual desktops that could be individually assigned to class participants for use over the semester. Each user had full ownership of the virtual desktop and could install software as needed within the terms of each application's license agreement.

Preparing and deploying the virtual desktops was not without its share of difficulties. To begin with, customizing the template answer file proved challenging since a number of entries were needed to setup and activate the virtual desktops as they are deployed. Next, several undocumented caveats in the template preparation phase caused the deployment process to unexpectedly fail. Finally, roaming profile incompatibilities between Windows versions used for the virtual desktops and the advanced computing lab workstations led to interface problems when users first signed in to the virtual desktops. In the end however, all the difficulties were resolved leading to a successful deployment.

Based on the author's experience, desktop virtualization offers the most promise for academic environments implementing boutique computing. Once the initial investment is made to setup the infrastructure and debug the process, creating and deploying custom solutions becomes nearly automated and has the potential to replace the manual activities associated with traditional image creation, deployment, and maintenance.

In addition, moving the desktop from the physical workstation to a network accessible server removes the dependence upon location. This introduces a concept the author refers to as "Labs without walls" whereby specialized desktop configurations normally reserved for dedicated computer facilities could be accessed anywhere, anytime, on any type of device.[2] This not only has implications for technology savvy departments, but for any academic unit across a campus that relies on computing technology in their curriculum.

"Labs without walls" potentially marks a shift from decentralized academic computing facilities to centralized private or public cloud infrastructures. Due to virtualization, application expertise and administration can remain in the functional areas where it belongs while systems are deployed remotely. It should be noted that this framework does not do away with dedicated classroom and lab facilities, but does impact the type of technology required to equip the facility and equipment replacement lifecycles.

Discussion would not be complete without considering some of the drawbacks and caveats associated with desktop virtualization. To begin with, not all applications will run on a virtual desktop or at least on certain virtualization platforms (e.g. VMware Workstation will not run on a Windows operating system with Hyper-V enabled); while other application may be dependent on particular hardware (i.e. dongles, graphics cards, coprocessors, etc.) and require native access. In addition, virtual desktop infrastructure adds yet another licensing layer that must be managed along with application licenses.

Desktop virtualization is by no means a one size fits all technology, however, it does offer an effective means to create and deploy boutique computing solutions. Beginning Spring Semester 2014, 20 new virtual desktops were created for two different classes needing custom configurations.

[2] This service is also referred to generically as Desktop as a Service (DaaS).

3.4 User State Virtualization

The goal of user state virtualization is to provide a consistent experience and access to individual data across platforms regardless of the computing system selected by the client. The technology relies heavily on networking to connect client systems to centralized servers that store and manage user files and settings. Options are available to host user state virtualization solutions locally or through a cloud service.

The department maintains its own private active directory forest using Microsoft products for its domain infrastructure. User state virtualization is implemented using roaming profiles in conjunction with redirected folders and managed centrally using group policy. Two domain controllers provide domain services for the computing infrastructure and user files and settings are stored on a virtualized file server configured for high availability on a failover cluster.

Although roaming profiles has been enhanced to improve reliability and reduce network traffic over the years, the technology is nearly two decades old and showing its age. A recent exercise to deploy virtual desktops using Windows 8.1 in a shared workstation environment based on Windows 7 introduced a number of incompatibilities when the user profile service attempted to synchronize settings across platforms. Fortunately, most of the problems were able to be resolved—at least temporarily—until the latest incarnation of user state virtualization from Microsoft called User Environment-Virtualization (UE-V) can be evaluated.

Another effort to introduce Linux desktop systems as full-fledged domain members has been particularly challenging. Although Linux systems can authenticate users against active directory, there appears to be few tools available that enable Linux clients to utilize user state virtualization technologies. Unfortunately, current user state virtualization solutions simply are not capable of natively spanning heterogeneous deployments due to incompatibilities and design differences between operating system.

Lastly, shared storage and network performance are two important areas to consider when implementing user state virtualization in a boutique computing environment. It is common knowledge that shared storage systems are the most prone to becoming bottlenecks in highly virtualized systems. Decisions revolve around tradeoffs between storage capacity, input-output processing, and cost. For networking, the bandwidth between servers and distribution switches connecting the pool of workstations often becomes the limiting factor. In either case, system performance in an academic environment is generally defined by peak activity during class when all the participants perform the same tasks in unison and saturate storage and networking capabilities.

3.5 Native-Boot to Virtual Disk

Native-boot to virtual disk (a.k.a. boot to/from virtual hard disk) enables the launch of an operating system from a mutually exclusive set of operating systems specifically configured to a computer. This approach is limited to Microsoft's Windows operating system beginning with Windows 7 (Enterprise and Ultimate editions) and Windows 2008 R2, but proves useful in situations where one computer rotates through multiple roles that require particular configurations or where an operating system must be removed or replaced without impacting other configurations.

The author employs native-boot to virtual disk technology in the department's sandbox computer systems used for prototyping, demonstration, and experimentation. Specifically, computers used in the mock-up of the department's server and network infrastructure utilize native-boot to virtual disk to preserve configurations as mock-ups were built up and torn down. This approach demonstrated several advantages over other techniques—such as traditional multi-boot systems and server virtualization—that include the following:

- Native-boot to virtual disk maximizes a physical disk's contiguous free space. Only two partitions are required for native-boot to virtual disk—one for the Windows boot environment (200-350 MB) and a second for the virtual disks. The second partition can contain any number of virtual disks since each disk wholly contains the operating system. This is in contrast to a traditional multi-boot system that requires a separate partition for each operating system installed.
- A virtual disk created with the *dynamically expanding* option enables thin provisioning of physical storage space. This allows multiple virtual disks to be installed on a partition provided there is enough free space within the partition to encompass the logical size of the selected virtual disk at startup. Partitions allocated for an operating system in a traditional multi-boot system consume a fixed amount of space whether the operating system is active or not.
- A remote computer can be used to create the virtual disk and install the operating system. Once installed, the operating system can be generalized enabling deployment on any computer. The virtual disk then can be posted to a shared folder and copied to a host system. In a traditional multi-boot system, the operating system must be installed directly on a host partition.
- Native-boot virtual disks can be easily managed. Removing an operating system only requires deleting the virtual disk and removing the boot entry from the boot system store. Resetting an operating system entails overwriting an existing virtual disk with a fresh copy. Resetting an operating system on a traditional multi-boot system involves reinstalling the operating system from installation media.
- Native-boot to virtual disk interacts directly with the installed hardware and employs the specific drivers designed for the actual hardware components. This allows access to specific hardware capabilities and settings particular to the host computer. In comparison, an operating system installed on a virtual machine employs emulated or synthetic drivers that interact with hypervisor thus masking the native settings from the guest operating system.
- Only one configuration can be run at any given time. The boot menu allows a user to choose a particular configuration at start up and only this configuration will be loaded. Systems employing a hypervisor can be configured to launch a virtual machine at startup, however, additional effort must be made to ensure only one virtual machine is allowed to run at a time.

The advantage of using native-boot to virtual disk is that it requires few additional software or hardware resources to implement other than a local hard disk with enough storage capacity. In the department's sandbox systems, a 320 GB hard disk could potentially host four or more 60 GB virtual disks depending on its physical size. The only other resource needed to deploy native-boot to virtual disk was a shared folder to store template virtual disks.

4 Summary

Boutique computing increases the value technology brings to academic departments and programs by using virtualization to dynamically recast computer hardware into new roles and returning it to their original state with minimal intervention. Deploying custom solutions within a standardized computing framework benefits students, faculty, and instructional staff by providing the tools needed to enhance the learning experience. Accomplishing this within an existing, resource constrained environment maximizes investments made in hardware and software.

Virtualization plays an ever increasing role in computing infrastructures and is a key technology that enables boutique computing. The technology provides the mechanism in which to create truly universal machines that transform on demand to accomplish the task at hand. However, this technology is not free and requires considerable investments of time and resources to learn and master. This creates a paradox for academic departments that absolutely need their technology investments to perform, but lack the personnel or internal expertise to deploy such systems. Students seeking lab assistant positions face steep learning curves and often need extended training to help administer highly integrated and complicated systems.

In summary, this paper presented the concept of boutique computing and virtualization technologies that facilitate its implementation. To demonstrate progress toward the vision, concrete examples were provided from actual classroom setups, sandbox configurations, and virtual desktop deployments. Deployment issues, administrative concerns, and design considerations rounded out the discussion to provide insight into efforts needed transform a homogeneous infrastructure into a dynamic heterogeneous computing environment that advances the mission and goals of an academic department.