Employing Virtualization in Library Computing: Use Cases and Lessons Learned

Arwen Hutt, Michael Stuart, Daniel Suchy, and Bradley D. Westbrook

This paper provides a broad overview of virtualization technology and describes several examples of its use at the University of California, San Diego Libraries. Libraries can leverage virtualization to address many long-standing library computing challenges, but careful planning is needed to determine if this technology is the right solution for a specific need. This paper outlines both technical and usability considerations, and concludes with a discussion of potential enterprise impacts on the library infrastructure.

Operating system virtualization, herein referred to simply as "virtualization," is a powerful and highly adaptable solution to several library technology challenges, such as managing computer labs, automating cataloging and other procedures, and demonstrating new library services. Virtualization has been used in one manner or another for decades,[1] but it is only within the last few years that this technology has made significant inroads into library environments. Virtualization technology is not without its drawbacks, however. Libraries need to assess their needs, as well as the resources required for virtualization, before embarking on large-scale implementations. This paper provides a broad overview of virtualization technology and explains its benefits and drawbacks by describing some of the ways virtualization has been used at the University of California, San Diego (UCSD) Libraries.[2]

Virtualization overview

Virtualization is used to partition the physical resources (processor, hard drive, network card, etc.) of one computer to run one or more instances of concurrent, but not necessarily identical, operating systems (OSs). Traditionally only one instance of an operating system, such as Microsoft Windows, can be used at any one time. When an operating system is virtualized—creating a virtual machine (VM)—the VM communicates through virtualization middleware to the hardware or host operating system. This middleware also provides a consistent set of virtual hardware drivers that are transparent to the end-user and to the physical hardware. This allows the virtual machine to be used in a variety of heterogeneous environments without the need to reconfigure or install new drivers. With the majority of hardware and compatibility requirements resolved, the computer becomes simply a physical presentation medium for a VM.

Two approaches to virtualization: host-based vs. hypervisor

Virtualization can be implemented using Type 1 or Type 2 hypervisor architectures. A Type 1 hypervisor (figure 1), commonly referred to as "host-based virtualization," requires an OS such as Microsoft Windows XP to host a "guest" operating system like Linux or even another version of Windows. In this configuration, the host OS treats the VM like any other application. Host-based virtualization products are often intended to be used by a single user on workstation-class hardware.

In the Type 2 hypervisor architecture (figure 2), commonly referred to as "hypervisor-based virtualization," the virtualization middleware interacts with the computer's physical resources without the need of a host operating system. Such systems are usually intended for use by multiple users with the VMs accessed over the network. Realizing the full benefits of this approach requires a considerable resource commitment for both enterprise-class server hardware and information technology (IT) staff.

Use cases

Archivists' Toolkit

The Archivists' Toolkit (AT) project is a collaboration of the UCSD Libraries, the New York University Libraries, and the Five Colleges Libraries (Amherst College, Hampshire College, Mt. Holyoke College, Smith College, and University of Massechusetts, Amherst) and is funded by the Andrew W. Mellon Foundation. The AT is an open-source archival data management system that provides broad, integrated support for the management of archives. It consists of a Java client that connects to a relational database back-end (MySQL, MSsql, or Oracle). The database can be implemented on a networked server or a single workstation. Since its initial release in December 2006, the AT has sparked a great deal of interest and rapid uptake of the application within the archival community. This growing interest has, in turn, created an increased demand for demonstrations of the product, workshops and training, and simpler methods for distributing the application. (Of the use cases described here, the two for the AT

Arwen Hutt (ahutt@ucsd.edu) is Metadata Specialist, **Michael Stuart** (mstuart@ucsd.edu) is Information Technology Analyst, **Daniel Suchy** (dsuchy@ucsd.edu) is Public Services Technology Analyst, and **Bradley D. Westbrook** (bradw@library.ucsd.edu) is Metadata Librarian and Digital Archivist, University of California, San Diego Libraries.

distribution and laptop classroom are exploratory, whereas the rest are in production.)

AT workshops

The Society of American Archivists sponsors a two-day AT workshop occurring on multiple dates at several locations. In addition, the AT team provides one- and two-day workshops to different institutional audiences. AT workshops are designed to give participants a hands-on experience using the AT application. Accomplishing this effectively requires, at the minimum, supplying all participants with identical but separate databases so that participants can complete the same learning exercises simultaneously and independently without concern for working in each other's space. In addition, an ideal configuration would reduce the workload of the instructors, freeing them from having to set up the AT instructional database onsite for each workshop.

For these workshops we needed to do the following:

- provide identical but separate databases and database content for all workshop attendees
- create an easily reproducible installation and setup for workshops by preparing and populating the AT instructional database in advance

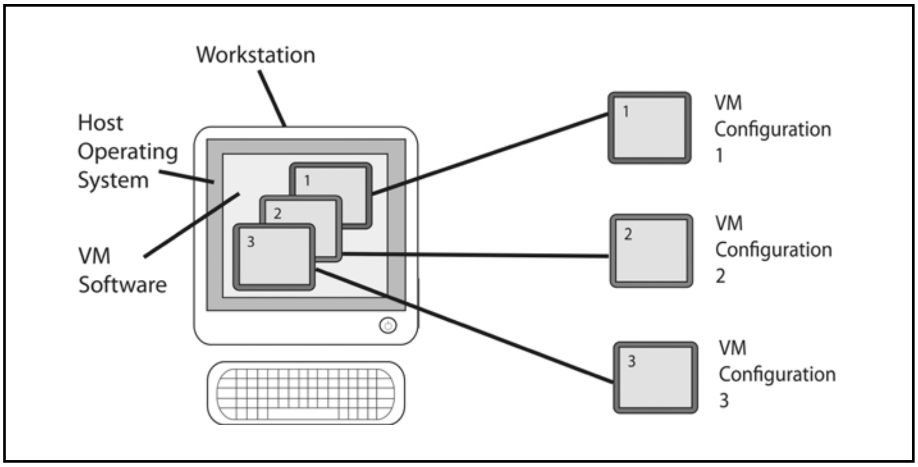

Figure 1. A Type 1 hypervisor (host-based) implementation

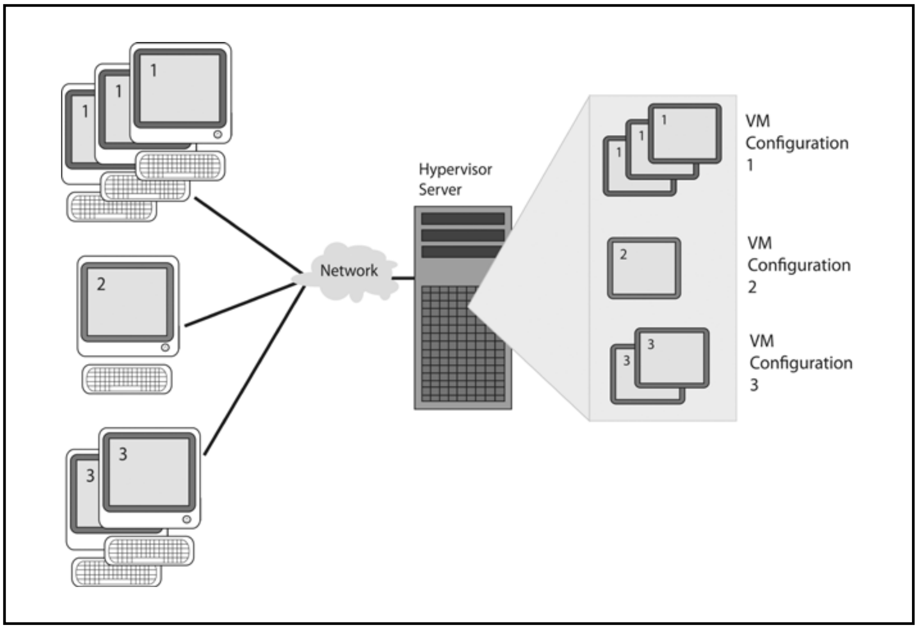

Figure 2. A Type 2 hypervisor-based implementation

Virtualization allows the AT workshop instructors to predefine the workstation configuration, including the installation and population of the AT databases, prior to arriving at the workshop site. To accomplish this we developed a workshop VM configuration with MySQL and the AT client installed within a Linux Ubuntu OS. The workshop instructors then built the AT VM with the data they require for the workshop. The AT client and database are loaded on a DVD or flash drive and shipped to the classroom managers at the workshop sites, who then need only to install a copy of the VM and the freely available VMPlayer software (necessary to launch the AT VM) onto each workstation in the classroom. The AT VM, once built, can be used many times both for multiple workstations in a classroom as well as for multiple workshops at different times and locations.

This implementation has worked very well, saving both time and effort for the instructors and classroom support staff by reducing the time and communication necessary for deploying and reconfiguring the VM. It also reduces the chances that there will be an unexpected conflict between the application and the host workstation's configuration. But the method is not perfect. More than anything else, licensing costs motivated us to choose Linux as the operating system instead of a proprietary OS such as Windows. This reduces the cost of using the VM, but it also requires workshop participants to use an OS with which they are often unfamiliar. For some participants, unfamiliarity with Linux can make the workshop more difficult than it would be if a more ubiquitous OS was used.

AT demonstrations

In a similar vein, members of the AT team are often called upon to demonstrate the application at various professional conferences and other venues. These demonstrations require the setup and population of a demonstration database with content for illustrating all of the application's functions.

One of the constraints posed by the demonstration scenario is the importance of using a local database instance rather than a networked instance, since network connections can be unreliable or outright unavailable (network connectivity being an issue we've all faced at conferences). Another constraint is that portions of the demonstrations need some level of preparation (for example, knowing what search terms will return a nonempty result set), which must be customized for the unique content of a database. A final constraint is that, because portions of the demonstration (import and data merging) alter the state of the database, changes to the database must be easily reversible, or else new examples must be created before the database can be reused.

Building on our experience of using virtualization to implement multiple copies of an AT installation, we evaluated the possibility of using the same technology for simplifying the setup necessary for demonstrating the AT. As with the workshops, the use of a VM for AT demonstrations allows for easy distribution of a prepopulated database, which can be used by multiple team members at disparate geographic locations and on different host OSs. This significantly reduces the cost of creating (and recreating) demonstration databases. In addition, demonstration scripts can be shared between team members, creating additional time savings as well as facilitating team participation in the development and refinement of the demonstration. Perhaps most important is the ability to roll back the VM to a specific state or snapshot of the database. This means the database can be quickly returned to its original state after being altered during a demonstration. Overall, despite our initial anxiety about depending on the VM for presentations to large audiences, this solution has proven very useful, reliable, and cost-effective.

AT distribution

Implementing the AT requires installing both the toolkit client and a database application such as MySQL, instantiating an AT database, and establishing the connection between database and client. For many potential customers of the AT, the requirements for database creation and management can be a significant barrier due to inexperience with how such processes work and a lack of readily available IT resources. Many of these customers simply desire a plug-and-play version of the application that they can install and use without requiring technical assistance.

It is possible to satisfy this need for a plug-and-play AT by constructing a VM containing a fully installed and ready-to-use AT application and database instance. This significantly reduces the number and difficulty of steps involved in setting up a functional AT instance. The customer would only need to transfer the VM from a DVD or other source to their computer, download and install the VM reader, and then launch the AT VM. They would then be able to begin using the AT immediately. This removes the need for the user to perform database creation and management; arguably the most technically challenging portion of the setup process. Users would still have the option of configuring the application (default values, lookup lists, etc.) in accord with the practices of their repository.

Batch processing catalog records

The rapid growth of electronic resources is significantly changing the nature of library cataloging. Not only are types of library materials changing and multiplying, the amount of e-resources being acquired increases each year. Electronic book and music packages often contain tens of thousands of items, each requiring some level of cataloging. Because of these challenges, staff are increasingly cataloging resources with specialized programs, scripts, and macros that allow for semiautomated record creation and editing. Such tools make it possible to work on large sets of resources—work that would not be financially possible to perform manually item by item. However, the specialized configuration of the workstation required for using these automated procedures makes it very difficult to use the workstation for other purposes at the same time. In fact, user interaction with the workstation while the process is running can cause a job to terminate prior to completion. In either scenario, productivity is compromised.

Virtualization offers an excellent remedy to this problem. A virtual machine configured for semiautomated batch processing allows for unused resources on the workstation to process the batch requests in an isolated environment while, at the same time and on the same machine, the user is able to work on other tasks. In cases where the user's machine is not an ideal candidate for virtualization, the VM can be hosted via a hypervisor-based solution, and the user can access the VM with familiar remote access tools such as Remote Desktop in Windows XP.

Secure sandbox

In addition to challenges posed by increasingly large quantities of acquisitions, the UCSD Libraries is also encountering an increasing variety of library material types. Most notable is the variety and uniqueness of digital media acquired by the library, such as specialized programs to process and view research data sets, new media formats and viewers, and application installers. Cataloging some of these materials requires that media be loaded and that applications be installed and run to inspect and validate content. But running or opening these materials, which are sometimes from unknown sources, poses a security risk to both the user's workstation and to the larger pool of library resources accessible via the network. Many installers require a user to have administrative privileges, which can pose a threat to network security.

The virtual machine allows for a user to have administrative privileges within the VM, but not outside of the VM. The user can be provided with the privileges needed for installing and validating content without modifying their privileges on the host machine. In addition, the VM can be isolated by configuring its network connection so that any potential security risks are limited to the VM instance and do not extend to either the host machine or the network.

Laptop classroom

Instructors at the UCSD Libraries need a laptop classroom that meets the usual requirements for this type of service (mobility, dependability, etc.) but also allows for the variety of computing environments and applications in use throughout our several library locations. In a least-common-denominator scenario, computers are configured to meet a general standard (usually Microsoft Windows with a standard browser and office suite) and allow minimal customization. While this solution has its advantages and is easy to configure and maintain from the IT perspective, it leaves much to be desired for an instructor who needs to use a variety of tools in the classroom, often on demand. The goal in this case is not to settle for a single generic build but instead look for a solution that accommodats three needs:

- The ability to switch quickly between different customized OS configurations
- The ability to add and remove applications on demand in a classroom setting
- The ability to restore a computer modified during class to its original state

Of course, regardless of the approach taken, the laptops still needed to retain a high level of system security, application stability, and regular hardware maintenance.

After a thorough review of the different technologies and tools already in use in the libraries, we determined that virtualization might also serve to meet the requirements of our laptop classroom. The need to support multiple users and multiple VMs makes this scenario an ideal candidate for hypervisor-based virtualization. We decided to use VDI (Virtual Desktop Infrastructure), a commercially available hypervisor product from VMware. VMware is one of the largest providers of virtualization software, and we were already familiar with several iterations of its host-based VM services.

The core of our project plan consists of a base VM to be created and managed by our IT department. To support a wide variety of applications and instruction styles, instructors could create a customized VM specific to their library's instruction needs with only nominal assistance from IT staff. The custom VM would then be made available on demand to the laptops from a central server (as depicted in figure 2 above). In this manner, instructors could "own" and maintain a personal instructional computing environment, while the classroom manager could still ensure the laptop classroom as a whole maintained the necessary secure software environment required by IT. As an added benefit, once these VMs are established, they could be accessed and used in a variety of diverse locations.

Considerations for implementation

Before implementing any virtualization solution, in-depth analysis and testing is needed to determine which type of solution, if any, is appropriate for a specific use case in a specific environment. This analysis should include three major areas of focus: user experience, application performance in the virtualized environment, and effect on the enterprise infrastructure. In this section of this paper, we review considerations that, in hindsight, we would have found to be extremely valuable in the UCSD Libraries' various implementations of virtualization.

User experience

Traditionally, system engineers have developed systems and tuned performance according to engineering metrics (e.g., megabytes per second and network latency). While such metrics remain valuable to most assessments of a computer application, performance assessments are being increasingly defined by usability and user experience factors. In an academic computing environment, especially in areas such as library computer labs, these newer kinds of performance measures are important indicators of how effectively an application performs and, indirectly, of how well resources are being used.

Virtualization can be implemented in a way that allows library users to have access to both the virtualized and host OSs or to multiple virtualized OSs. Since virtualization essentially creates layers within the workstation, multiple OS layers (either host or virtualized) can cause the users to become confused as to which OS they are interacting with at a given moment. In that kind of implementation, the user can lose his or her way among the host and guest OSs as well as become disoriented by differing features of the virtualized OSs. For example, the user may choose to save a file to the desktop, but may not be aware that the file will be saved to the desktop of the virtualized OS and not the host OS. External device support can also be problematic for the end user, particularly with regard to common devices such as flash drives. The user needs to be aware of which operating system is in use, since it is usually the only one with which an external device is configured to work.

Authentication to a system is another example of how the relationship between the host and guest OS can cause confusion. The introduction of a second OS implicitly creates a second level of authentication and authorization that must be configured separately from that of the host OS. User privileges may differ between the host and guest OS for a particular VM configuration. For instance, a user might need to remember two logins or at least enter the same login credentials twice. These unexpected differences between the host and guest OS produce negative effects on a user's experience. This can be a critical factor in a time-sensitive environment such as a computer lab, where the instructor needs to devote class time to teaching and not to preparing the computers for use and navigating students through applications.

Interface latency and responsiveness

Latency (meaning here the responsiveness or "sluggishness" of the software application or the OS) in any interface can be a problem for usability. Developers devote a significant amount of time to improving operating systems and application interfaces to specifically address this issue. However, users will often be unable to recognize when an application is running a virtualized OS and will thus expect virtualized applications to perform with the same responsiveness as applications that are not-virtualized. In our experience, some VM implementations exhibit noticeable interface latency because of inherent limitations of the virtualization software. Perhaps the most notable and restrictive limitation is the lack of advanced 3D video rendering capability. This is due to the lack of support for hardware-accelerated graphics, thus adding an extra layer of communication between the application and the video card and slowing down performance. In most hardware-accelerated 3D applications (e.g., Google Earth Pro or Second Life), this latency is such a problem that the application becomes unusable in a virtualized environment. Recent developments have begun to address and, in some cases, overcome these limitations.[3]

In every virtualization solution there is overhead for the virtualization software to do its job and delegate resources. In our experience, this has been found to cause an approximately 10–20 percent performance penalty. Most applications will run well with little or moderate changes to configuration when virtualized, but the overhead should not be overlooked or assumed to be inconsequential. It is also valuable to point out that the combination of applications in a VM, as well as VMs running together on the same host, can create further performance issues.

Traditional bottlenecks

The bottlenecks faced in traditional library computing systems also remain in almost every virtualization implementation. General application performance is usually limited by the specifications of one or more of the following components: processor, memory, storage, and network hardware. In most cases, assuming adequate hardware resources are available, performance issues can be easily addressed by reconfiguring the resources for the VM. For example, a VM whose application is memory-bound (i.e., performance is limited by the memory available to the VM), can be resolved by adjusting the amount of memory allocated to the VM.

A critical component of planning a successful virtualization deployment includes a thorough analysis of user workflow and the ways in which the VM will be utilized. Although the types of user workflows may vary widely, analysis and testing serve to predict and possibly avoid potential bottlenecks in system performance.

Enterprise impact

When assessing the effect virtualization will have on your library infrastructure, it is important to have an accurate understanding of the resources and capabilities that will form the foundation for the virtualized infrastructure. It is a misconception that it is necessary to purchase state-of-the-art hardware to implement virtualization. Not only are organizations realizing how to utilize existing hardware better with virtualization for specific projects, they are discovering that the technology can be extended to the rest of the organization and be successfully integrated into their IT management practices. Virtualization does, however, impose certain performance requirements for large-scale deployments that will be used in a 24/7 production environment. In such scenarios, organizations should first compare the level of performance offered by their current hardware resources with the performance of new hardware. The most compelling reasons to buy new servers include the economies of scale that can be obtained by running more VMs on fewer, more robust servers, as well as the enhanced performance supplied by newer, more virtualization-aware hardware. In addition, virtualization allows for resources to be used more efficiently, resulting in lower power consumption and cooling costs.

Also, the network is often one of the most overlooked factors when planning a virtualization project. While a local virtualized environment (i.e., a single computer) may not necessarily require a high performance network environment, any solution that calls for a hypervisor-based infrastructure requires considerable planning and scaling for bandwidth requirements. The current network hardware available in your infrastructure may not perform or scale adequately to meet the needs of this VM use. Again, this highlights the importance of thorough user workflow analyses and testing prior to implementation.

Depending on the scope of your virtualization project, deployment in your library can potentially be expensive and can have many indirect costs. While the initial investment in hardware is relatively easy to calculate, other factors, such as ongoing staff training and system administration overhead, are much more difficult to determine. In addition, virtualization adds an additional layer to oftentimes already complex software licensing terms. To deal with the increased use of virtualization, software vendors are devoting increasing attention to the intricacies of licensing their products for use in such environments. While virtualization can ameliorate some licensing constraints (as noted in the AT workshop use case), it can also conceal and promote licensing violations, such as multiple uses of a single-license applications or access to license-restricted materials. License review is a prudent and highly recommended component of implementing a virtualization solution. Finally, concerning virtualization software itself, it also should be noted that while commercial VM companies usually provide plentiful resources for aiding implementation, several worthy open-source options also exist. As with any open-source software, the total cost of operation (e.g., the costs of development, maintenance, and support) needs to be considered.

Conclusion

As our use cases illustrate, there are numerous potential applications and benefits of virtualization technology in the library environment. While we have illustrated a number of these, many more possibilities exist, and further opportunities for its application will be discovered as virtualization technology matures and is adapted by a growing number of libraries. As with any technology, there are many factors that must be taken into account to evaluate if and when virtualization is the right tool for the job. In short, successful implementation of virtualization requires thoughtful planning. When so implemented, virtualization can provide libraries with cost-effective solutions to long-standing problems.

References and notes

[1] Alessio Gaspar et al., "The Role of Virtualization in Computing Education," in *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education* (New York: ACM, 2008): 131–32; Paul Ghostine, "Desktop Virtualization: Streamlining the Future of University IT," *Information Today* 25, no. 2 (2008): 16; Robert P. Goldberg, "Formal Requirements for Virtualizable Third Generation Architectures," in *Communications of the ACM 17, no. 7* (New York: ACM, 1974): 412–21; and Karissa Miller and Mahmoud Pegah, "Virtualization: Virtually at the Desktop," in *Proceedings of the 35th Annual ACM SIGUCCS Conference on User Services* (New York: ACM, 2007): 255–60.

[2] For other, non–UCSD use cases of virtualization, see Joel C. Adams and W. D. Laverell, "Configuring a Multi-Course Lab for System-Level Projects," *SIGCSE Bulletin* 37, no. 1 (2005): 525–29; David Collins, "Using VMWare and Live CD's to Configure a Secure, Flexible, Easy to Manage Computer Lab Environment," *Journal of Computing for Small Colleges* 21, no. 4 (2006): 273–77; Rance D. Necaise, "Using VMware for Dual Operating Systems," *Journal of Computing in Small Colleges* 17, no. 2 (2001): 294–300; and Jason Nieh and Chris Vaill, "Experiences Teaching Operating Systems Using Virtual Platforms and Linux," *SIGCSE Bulletin* 37, no 1 (2005): 520–24.

[3] H. Andrés Lagar-Cavilla, "VMGL (formerly Xen-GL): OpenGL Hardware 3D Acceleration for Virtual Machines," www.cs.toronto.edu/~andreslc/xen-gl/ (accessed Oct. 21, 2008).