

## **MOVING TOWARD A SERVER-BASED VIRTUAL MACHINE HOSTING ENVIRONMENT IN SUPPORT OF UNIVERSITY INFORMATION TECHNOLOGY PROGRAMMING COURSES**

*George Stefanek, PhD, Purdue University North Central, stefanek@pnc.edu*  
*Nipul Patel, Purdue University North Central, npatel@pnc.edu*

### **ABSTRACT**

*University information technology programming courses require the use of an integrated development environment such as Microsoft Visual Studio and often other additional software such as a database system or middleware in order to complete programming assignments. Students prefer to have the flexibility of performing programming tasks on their own PCs at times convenient to them instead of using university lab computers that have the required software. This requires students to download and install the various pieces of software needed for the course onto their own computers. Students can experience complex installation problems, compatibility issues with university computers, and compatibility issues when integrating multiple pieces of software together on team projects. To avoid these type of problems students need to have a software development environment where all the software components and operating systems are exactly at the same version level as the university lab releases. This will provide a stable and compatible platform for programming, and allow students to focus their full efforts on course programming assignments. This paper proposes that the most effective environment to solve these problems is to use virtual machines hosted on university servers that have an approved and tested software configuration to support specific course requirements.*

**Keywords:** Virtualization, Virtual Machines, Educational Technology, University IT, Information Technology

### **INTRODUCTION**

Information Technology (IT) programming courses typically require an integrated development environment (IDE) such as Microsoft Visual Studio, Eclipse or Netbeans and may also require additional software such as a database system (e.g., Oracle, Microsoft SQL Server, Microsoft Access or MySQL), spreadsheet software, middleware or other software components. Programming assignments may require students to write code which interfaces to databases, spreadsheets, external files, access external class libraries, access services either locally or as cloud services, or use middleware to create forms. Team projects require student collaboration and integration of individual programming components into one integrated system. To support these activities, these complex programming environments are setup on university lab computers. Students, however, usually prefer to setup their own computers with the required software so that they can work on their assignments and projects at home and at any time they want rather than on university lab computers which have restricted hours. University labs have computers with standardized images that contain multiple IDEs and other common software needed to support a variety of courses. These lab computers have software locked at a specified version level and have been tested to work correctly for the specified configurations. Students, however, can experience numerous problems when trying to configure their own programming environments and create incompatibilities with university configured software environments. In order to resolve these problems and enable students to focus on their programming assignments and have a more enjoyable and effective learning experience, this paper proposes the use of virtual machines hosted on university servers to support university information technology programming courses.

### **PROBLEMS ENCOUNTERED IN SETTING UP PROGRAMMING ENVIRONMENTS**

During installation of software packages required by specific programming courses, students can experience various installation problems, insufficient hardware resources, and compatibility problems when installing the software on their own computers. Software packages sometimes require the installation of additional software components at a specific version level which can lead to greater installation complexity or a failed installation. Post installation software updates (patches) may need to be applied. Also, periodic software updates that may occur during the

duration of the university course may sometimes break the installed software on the student's personal computer or require installation of other updates to other components on the student's computer. All these issues can lead to problems successfully installing the software, running the software through the duration of the course, or contribute to potential integration problems of code pieces developed across multiple computers on team projects. When students setup their own development environments, there are often mismatches with the version of the software installed on the university lab computers. When students develop code on their own computers and other code on the university lab computers, then this can lead to incompatibilities which cause errors during compilation of the code. These type of problems, even though they can be a learning experience, distract students from their class assignments and projects which are the main focus of a programming course.

Other problems encountered in teaching environments in support of information technology programming courses is the lack of adequate computer resources. Each programming course will have its own set of software and operating system requirements. Some courses only require a development environment such as Microsoft Visual Studio with C++, C# or Visual Basic. Other courses may require Java and Netbeans or that the software being developed integrate with a database system or use middleware to create forms. Some courses may specify that the development environment be installed on a Linux platform rather than Microsoft Windows. The combinations of software and system requirements is very wide and difficult to support by the university using only their limited lab computer resources. Sometimes there are not enough computers with the necessary software configurations with the specified operating system (OS) to support course requirements. Some courses may require more than one computer to be used. University computer labs open to the general student population usually have limited hours and typically have lab load images that contain various IDEs, database systems, word processor, spreadsheet program, and miscellaneous software that run on Microsoft Windows or Linux. However, they don't typically have both Linux and Windows dual boot configurations. Networking, Linux, and operating system courses may require modification of the operating system and a network to practice configurations and perform assignments. Students sometimes opt to setup their own virtual machines and then install the software components within the virtual machine so that they don't disturb their original software environment on their own computers. This can further create even more complexity and problems if not properly configured.

### FREQUENCY OF STUDENT RELATED SOFTWARE CONFIGURATION ISSUES

This study has compiled some preliminary statistics during the span of 4 semesters and 10 programming courses from a population of 196 students that show that from 9.8% to 29% of students in the various programming and information technology courses have had setup issues with operating systems, and the installation of IDEs and other software components. Also, 9.1% to 13.7% of students have had compatibility issues when integrating software from various team members working while working on team projects where the team members used a mix of their own PCs and university computers to perform coding, the design of graphical forms, and interfacing to databases. The IDEs for various courses included Microsoft Visual C++ Express for a C++ programming course, Microsoft Visual Web Developer Express with Visual Basic.NET and Microsoft SQL Server for an ASP.NET programming course, Java with Netbeans, Derby and MySQL for a Java programming course, and Linux for a Linux course which required writing scripts. Even though the number of issues can vary widely from semester to semester, on the average, the percentage of problems was high enough that it caused concern that it could have an impact on the quality of students work.

**Table 1.** Frequency of Issues across all Programming Courses

Issue	Students Affected / Total Students	Percentage
Installation / Configuration	31 / 196	15.8%
Incompatibility / Integration	16 / 196	8.2%
Total for All Issues Combined	47 / 196	24.0%

**Table 2: Frequency of Installation / Configuration Issues by Specific Course Types**

Type of Course	Students Affected / Total Students	Percentage
Java Programming	9 / 31	29.0%
C++ Programming	11 / 92	12.0 %

---

ASP.NET Programming	5 / 51	9.8%
Linux Scripting	6 / 22	27.3%

**Table 3:** Frequency of Incompatibility / Integration Issues by Specific Course Type

Type of Course	Students Affected / Total Students	Percentage
Java Programming	3 / 31	9.7%
C++ Programming	4 / 92	4.4%
ASP.NET Programming	7 / 51	13.7%
Linux Scripting	2 / 22	9.1%

### VIRTUAL MACHINE TECHNOLOGY BACKGROUND

To attempt to solve the installation, configuration and integration problems that students had experienced, virtual machines configured with all the required software components for a particular course hosted on university servers were configured. This solution has been initially implemented in Java programming, database design, and Unix courses.

#### What are Virtual Machines

A virtual machine (VM) is a software computer that like a physical computer runs an operating system and can execute its applications [9]. The VM is completely separate and isolated from the host computer and typically runs some version of Microsoft Windows or Linux. This isolation from the hosting operating system allows for multiple VMs to run simultaneously on a single computer. Some examples of where virtual machines are used are: 1) running legacy software on an older version of an OS without the need to port the software to every newer revision of the operating system, 2) testing software products prior to release on multiple operating systems and multiple versions of the operating systems without the need for individual computer hardware for each OS, and 3) isolating applications and processes from the hosting operating system for security reasons.

#### Virtualization Technology

Virtualization is software technology that allows a single server to host multiple virtual machines. There are four major types of virtualization technology as defined by Shields: hardware, operating system, application virtualization, and para-virtualization [6]. The technology used at the Purdue University North Central Information Services department to host virtual machines is VMWare Horizon (with View). This technology is a hardware virtualization type technology that emulates the physical computer hardware and runs an installed operating system that is compatible with the emulated hardware. A thin layer of software called a hypervisor separates the virtual machines from the host server and dynamically allocates computing resources to each virtual machine as needed. As each virtual machine encapsulates an entire computer, many applications and operating systems can run on a single host server at the same time [9].

#### Advantages and Disadvantages Of Using Server-based Virtual Machines

Each virtual machine is setup with a specified operating system and all the required software needed to perform the work for a particular programming course. The virtual machine is tested by the Information Services department prior to release to assure that the software works properly. Separate virtual machines are created for each student in a course from a single pre-configured virtual machine that has all the required software installed on it. Professors can create assignments knowing that the assignments can be accomplished reliably within the VM environment and the code will function consistently for all students. Hosting the VMs on university servers reduces the need to install a virtual machine on every lab computer and allows for the virtual machines to be accessed remotely by students using an internet connection. This same environment can be used for distance education courses as well as on-site programming courses. Software development that utilizes reading and writing to databases would allow for individual development and for compatible integration of all software components developed by multiple students for team projects. The learning experience is enhanced for the student primarily by eliminating compatibility issues so the student can focus on their specific homework and project assignments.

A single virtual machine may include multiple software packages that support multiple courses or be tailored for each individual programming course. A standardized environment with identical software at the same software version would be used by all students in a course preventing incompatibility issues. The need for dedicated computer hardware resources in the computer laboratory would be reduced. Security would be enhanced by using Information Services configured security and updates [1]. Since VMs are separate from the host's (server's) operating system, if one virtual machine crashes, it doesn't affect the server or any other virtual machines running on it. VMs could be transferred from one compatible server to another by simply copying the VMs to other servers [9]. Once these virtual machines are configured for various courses, they can be re-used for the same courses over multiple semesters before needing to be updated. This can lead to maintenance cost reduction [1].

The disadvantages of using a server-based virtual machine hosting environment include: 1) the necessity to purchase enough virtual machine licenses to support multiple course student loads (this requires planning ahead of time before deployment), 2) the IT department has to configure, test and support all the required VMs which could lead to greater IT staff commitments in the short run, and 3) any configuration issues effect all students in the course and require immediate response by the IT department.

### **USING SERVER-BASED VIRTUAL MACHINES FOR PROGRAMMING COURSES**

Virtual machines have been used in teaching environments to teach classes in networking, network security, system administration [4], operating systems, database systems and administration, and distance education [7, 8], and other teaching areas. Ogunyemi [5], Bower [2], and Hacker [3] discuss experiences with virtualization technology in education. Some of these teaching environments use virtual machines installed on individual computers while others use server-based virtual machine hosting.

#### **Approach**

This paper proposes that the best approach to create a stable software development environment for student programming assignments and projects in information technology programming courses to use virtual machines configured and tested for each specific programming course and hosted on university servers. Each specific course will have its own virtual machine configured with the required software components and operating system. After configuration of the virtual machine, university Information Services will need to make copies of the VM for each student in every programming course. Once the virtual machines are copied onto the university servers, then students can access the VM using a browser or by downloading a small client application that connects to the VM hosting server using the internet. This sets up an environment where every student in a class has access to identical software development environments from any location where there is an internet connection. No time is spent setting up individual computers and configuring the required software for a course. Students can immediately begin doing their assignments and/or projects under the required programming environment.

#### **Server-based Virtual Machine Hosting Environment Configuration**

After configuring a virtual machine for the course, the Information Services department gets a list of students in the class, and then creates copies of the VM for each student in that class. University student accounts are used by the students to connect via the internet to the university's virtual server resources. Each of these virtual machines is hosted on servers run by the university Information Services. Each virtual machine would typically be allotted 4-5 GB of space to support the operating system and installed software. For 20 students in a programming class 80 GB of space would need to be allocated on a cluster of servers. The cluster software balanced the load onto individual processors automatically. VMWare Horizon View is used as the VM hosting software. The university would have to have adequate resources in terms of servers and disk space in order to support multiple classes. Planning for this type of support needs to be done at least one semester ahead of time in order to make sure resources are adequate to support the need. Planning should include: 1) evaluating the software requirements for a course, 2) evaluating the hardware requirements to support a course, 3) creating a plan to support the requirements including getting all the resources and software licensing to support the requirements, 4) creating a virtual machine for the course and loading and configuring all the required software onto the virtual machine, 6) testing the virtual machine, 7)

reviewing the results of testing, 8) making any modifications and retesting, 9) getting a list of all the students in the course and creating individual VMs for each student, and 10) deploying all the VMs onto the university servers.

The Purdue University North Central campus has a cluster of 5 servers as part of the VM hosting environment. Each server has a dual Xeon processor with a total of 165 GB of total memory and can support at least two programming courses at a time with its VM hosting environment. The Purdue University North Central campus is a commuter campus with a student population of 3,400 full and part-time students. Most students live within 45 minutes of the campus, but some students in outlying areas may drive twice as long. The desire for more programming and other course resources to be made available online including the availability of these resources for distance education is increasing and the infrastructure to support these type of courses with complex programming environments can be met relatively inexpensively using server-hosted virtual machines.

### CONCLUSIONS

Purdue University North Central information technology courses require the use of various software such as multiple IDEs (e.g., Microsoft Visual Studio, Netbeans, Eclipse, Visual Web Developer, and others), multiple database systems (e.g., Microsoft SQL Server, Oracle, MySQL, and others), and multiple operating systems (e.g., Microsoft Windows, Linux, and Android). Data has been collected for 10 of the courses offered over a period of 4 semesters that showed a variety of problems encountered by students in the installation and use of the required software for these courses (see section "FREQUENCY OF STUDENT RELATED SOFTWARE CONFIGURATION ISSUES" for statistics). Some of the specific problems that students experienced across multiple courses prior to the use of server hosted VMs included: 1) download of IDE software failures, 2) installation of IDE questions, 3) inadequate personal computer resources to install or operate their own virtual machine or other required software, 4) failure of Visual Studio to compile C++ code after installation, 5) IDEs stopped working in the middle of the semester due to failure to apply updates to the IDE and operating system, 6) problems with ASP.NET when accessing university virtual drives, 7) trouble installing the Java Derby database, 8) team collaboration failures between Netbeans versions, 9) failure of some ASP.NET forms to run due to incompatibility issues when porting the forms from a newer version of ASP.NET to an older version on university lab computers, 10) incompatibilities when running two versions of ASP.NET on the same computer, and 11) no availability of computers with Linux on university lab computers.

These various problems were numerous enough to move toward the use of server hosted virtual machines to support programming courses such as Object Oriented Programming using Java, database systems, and Unix in order to alleviate these problems. The Java Object Oriented Programming course had one of the highest percentage rates of 29% for installation / configuration issues so it was chosen as one of the first programming courses to implement the use of university server hosted VMs. Using server-based VMs has eliminated the need for students to download and configure their own programming environments or download and setup their own VM software.

Early results from using university server hosted VMs have been favorable where configuration problems have been eliminated from 29% to 0% in a Java programming course and provided a relatively stable development environment for the course. It has allowed students to conveniently access the VM environment using their home internet connection or from any other internet connected location by downloading and installing a VMware Horizon client software component onto their computer that connects to the student's virtual machine on the university servers. Students have responded favorably to the VM environment and seem to prefer it over other options. Overall, the experience so far has been favorable and it is planned to expand this type of environment into more programming courses. Future data on the experiences by students will be collected by formal detailed surveys.

Some of the managerial issues for instructors of using a VM environment for programming courses are: 1) at the start of the semester a VM machine needs to be setup for a specific course and tested with instructor collaboration, and 2) a list of students in the class needs to be submitted to the IS (Information Systems) department in order to setup the appropriate amount of VMs for the class. The instructor will usually be required to participate in the setup of the master VM by IS staff since the faculty member is the most familiar with the specific IDEs and any other software that will need to be installed on the VM for the course. The setup of a master VM for the course will be setup once and then reused from semester to semester until the software needs to be changed or upgraded.

The managerial issues related to the VM environment for Information Systems staff are: 1) the IS staff needs to setup the master VM for a course, 2) once the master VM is setup and tested, the IS staff needs to make copies of it within the VM hosting environment for every student in the class list submitted by the instructor, 3) the IS staff needs to support the environment during the duration of the semester which may require periodic issues to be addressed, 4) the IS staff needs to plan and implement the necessary hardware and software upgrades to support future classes and class loads that will require VMs, and 5) some of the IS staff needs to be trained on how to work with the VM hosting environment. Training for IS staff can be done using VMWare's Horizon (with View) on-line training course that includes hands-on lab exercises. Future work will include a detailed survey of IS department staff regarding the actual impact on resources, management and technical support requirements and specific problems and fixes to the VM environment.

The organizational benefits to the university are an anticipated reduced need of physical PCs in labs throughout the university. Since multiple VMs can run on a single server, there is an implied savings on PC hardware resources for the university. There is less of a projected need to increase lab space and increased lab staffing requirements as the student population grows. Other organizational benefits to the university are since students have 24 hour access to VMs via the internet it reduces the need for students to physically come into labs to do assignments which can reduce the hours when labs are open. On the other hand, as the VM hosting environment grows, the hardware and software resources to support it will need to grow as will increased training of IS staff to support the environment. Proposed further study will gather data to determine if there is an actual decrease in physical resources and staff as the use of server hosted VMs grows.

Some of the limitations of using a VM hosting environment are that when there is a problem with the environment it can affect all or multiple students that are using it in the class. Performance of the VM for students with a slow internet connection may be poor and thereby frustrating to use. An under configured VM hosting environment with not enough servers / memory or inappropriately configured VM hosting software can contribute to sluggish performance of the VM. So far, the technical issues with the VMs were: 1) students that had slow internet connections (dial-up) had sluggish response time when using the VM hosting environment, and 2) there were occasional problems starting the VM. The IS department responded quickly to resolve the second problem early in the semester. This problem disappeared in the second half of the semester as the VM hosting environment became appropriately configured.

Future evaluation and development will include: (1) expanding the use of VMs to additional programming courses, (2) getting additional, more formal, detailed feedback from students, faculty, and the IS department using surveys of the issues, effectiveness and experiences using and supporting VMs, (3) analyzing the results from the surveys and proposing responses to any issues, (4) looking at the costs of configuring and supporting the VM environment, (5) analyzing the savings from using this environment, and (6) determining the exact training, expertise and skill sets required to support the environment.

## REFERENCES

1. Anderson, B., Joines, A., & Daniels, T. (2009). Xen worlds: Leveraging virtualization in distance education. *Proceeding of ITiCSE 2009*, 293-253.
2. Bower, T. (2010). Experiences with virtualization technology in education. *Journal of Computing Sciences in Colleges*, 25(5), 311-318.
3. Hacker, T. (2011). AC 2011-1231: Exploring the use of virtual machines and virtual clusters for high performance computing education. American Society for Engineering Education. Available: [www.asee.org/public/conferences/1/papers/1231/download](http://www.asee.org/public/conferences/1/papers/1231/download)
4. Marshall, B., Cardon, P., Callender, C., Robertson, and P, Patel, N. (2011). Using VMware vCenter to teach system administration in a lab. *Issues in Information Systems*, XII(2), 153-161.
5. Ogunyemi, A., & Johnston, K. (2010). The use of virtual machines to support hands-on learning experiences in undergraduate systems-oriented courses. *Proceedings of the 4th International Conference on Dynamic Informatics*, Monash University, South Africa.
6. Shields, G. (2008). The shortcut guide to selecting the right virtualization solution. Realtime Publishers.

7. Stackpole, B. (2008). The evolution of a virtualized laboratory environment. *Proceedings of SIGITE 2008*, 243-248.
8. Stackpole, B., Koppe, J., Haskell, T., Guay, L., & Pan. Y. (2008). Decentralized virtualization in systems administration education. *Proceedings of SIGITE 2008*, 249-253.
9. VMware. (2015). What is a virtual machine: VMWare. Available: <http://www.vmware.com/virtualization-basics/how-virtualization-works.html>