Trevor Osborne

Eduardo Pantoja

CSCI 305 - Concepts of Programming Languages

Programming Language Service or Construct Paper

November 17, 2021

Virtual Machines

Introduction

The programming industry has changed drastically over the last 50 years. Where some elements of the past have been lost to time, some have remained while slowly improving over time. Virtual machines are one of these elements that have existed close to the beginning of the programming industry. Originally seen solely as a way to split one single operating system onto multiple machines for multiple users or to bootstrap languages onto new hardware with subpar performance, virtual machines have significantly improved and modernized. Virtual machines are now seen running behind several programming languages and software on the average consumer's PC.

Why We Chose It

Virtual machines (VM) are a topic that the both of us in our group have heard come up several times. Whether this be in the programming field running behind a specific language, a virtualization of the Windows operating system on a MacOS device, or emulating outdated software on your PC, this term comes up a lot. However, neither of us have really sat back to answer the question of what a virtual machine is. In

common media, virtual machines are shown what they do, but not necessarily how they do it and why they have become such a commonplace today. This is why we chose to go into more detail and learn exactly what virtual machines are.

What is a virtual machine?

A virtual machine is a virtualization/emulation of a computer system environment. Its application programming interface (API) includes everything needed to correctly execute programs that run inside of it. Every virtual machine includes an instruction set architecture (ISA) designed to run the programs within it. These instruction sets may be identical to an existing hardware's ISA, or completely built from the ground up for this specific VM (Scott, 2015).

Virtual machines are generally broken into two different types, these are system and process virtual machines. The most complex of the two, being system virtual machines, emulate all of the hardware facilities needed to run all features inside of an operating system (OS). These include both privileged and unprivileged instructions, memory-mapped I/O, virtual memory, and interrupt facilities. This emulated operating system functions identically to an authentic hardware run version of this system. This includes the ability to install external software compatible with that specific operating system. A process virtual machine is a lot simpler in comparison to a system virtual machine where it exclusively provides the environment needed to run a single process which, once finished, terminates the VM (Scott, 2015).

System virtual machines are generally managed by a virtual machine monitor (VMM), also known as a hypervisor. The hypervisor multiplexes a single physical

machine among a collection of other operating systems which run inside of their own unique virtual machine. This allows the ability to circumvent the creation of an operating system that supports multiple users. This has directly led into both cloud computing, by allowing a hosting center to share physical machines among a large number of isolated users, and simulated operating systems for development when compatibility issues are encountered (Scott, 2015).

Process virtual machines have had a great impact on the distribution of software when developed in specific languages. Process virtual machines can function as a quick solution to running software on architecture types that may not naturally support it. Some languages, like Java, include their own integrated virtual machine. In the case of Java, the Java Virtual Machine (JVM) exists for an easier software distribution when it comes to compatibility (Scott, 2015).

What Issue is Addressed

Virtual machines are primarily used to solve two problems based on the two types of virtual machines that exist. In the instance of a specific process needing a peculiar architecture set, a process virtual machine can fill this void. When a company wants to minimize costs associated with hardware when distributing a cloud based service, system virtual machines can greatly aid in this. Additionally, system virtual machines provide a layer of great compatibility when simulating other operating systems on one's hardware. Developers can utilize this for testing their applications on other systems without specifically having to acquire and invest in that hardware. Performance

is generally worse than the actual hardware that would be used, however, it does provide a great sample and first impression of what to expect.

What are the Challenges?

While the use of virtual machines are handy for many companies that use them, there are many challenges that are faced when implementing this technology. The most common difficulties that are faced are sprawl, network congestion, hardware limitations, and security risks (Bigelow, 2019; TechAdvisory, 2019).

Virtual machine sprawl occurs when companies lack the required practices to manage virtual machine creation. Virtual machines make it very easy to create new servers, so when workloads are virtualized, it's common that more physical servers will need to be purchased to accommodate the increase of workloads. The lack of management of these systems causes many of these processes to go unnoticed and suck up resources rapidly, especially ones that are not being used anymore. Thus, leading in company resources to go to waste in purchasing more servers rather than properly managing the used vs. unused virtual machines. This can also lead to issues with backing up resources as well.

Network congestion occurs when the server's Network Interface Card (NIC) is overloaded with virtual machine workloads resulting in network errors or crashes. Many servers only have a single NIC port, so when one virtual machine is running it likely will not become an issue. However, as more and more virtual machines are used on a single server, it doesn't take long to limit the NIC's capabilities (Bigelow, 2019).

One major problem that users face is that virtual machines tend to be less efficient. Using emulated software means that the virtual machine has to request access to the hardware from the host machine and once permission is granted the information has to go back to the emulation. This adds time to the process which hinders efficiency. Problems may also occur if a single host is running multiple virtual machines. If the computer has sufficient computational power, then this is not an issue. However, if the computer is old, slow, or simply lacks the power, then the virtual machines will feel sluggish and be hindered by the host's hardware limitations. Another issue that virtual machines face with hardware limitations is that they can be infected with issues from the host machine. For example, if the operating system of the host machine is hindered with bugs/problems, then any and all virtual machines run on that host will face the same problems that the host faces (Buyya et al., 2013).

Security risks can exist in numerous ways within virtual machines. One that you may face is a trojaned virtual machine. If you were to install a pre-built virtual machine, there could be malware within the software. It's possible to run into isolation failures when running multiple virtual machines on the same host. Isolation failures are caused with issues from the operating system, and risks are increased when multiple virtual machines within different trust levels are running on the same OS. Additional risks occur with offline virtual machines. Many updates and patches are constantly pushed out to software to increase security and prevent breaches. Offline virtual machines do not get these security patches/updates, thus posing more of a risk to sensitive information (Prescient Solutions, 2018).

While it may appear that there are many issues and challenges that are faced with the implementation of virtual machines, many of these issues can be addressed with easy solutions.

How This Was Handled Historically

While there is not an exact timeline as to when these difficulties were solved, there are many simple solutions that could prevent the above challenges from occurring. Companies can avoid virtual machine sprawl by adding policies and procedures to better manage their use of virtual machines. Some steps to follow this path could involve asking themselves, “Do we need a new virtual machine?” or, “How long do we foresee the use of this virtual machine?” It is important to treat the creation of a new virtual machine as if they were buying and installing a new physical server. Companies can also look into management tools for their virtual machines so they can often review what is currently being used and if it needs to be actively used right at that moment.

To prevent network congestion from arising, organizations can consider upgrading to servers that contain additional NIC ports to avoid bottlenecks. Companies, however, may need to consider that upgrading the amount of NICs may also require a switch upgrade or distributing the workload across multiple switches. With that in mind, they should be able to successfully avoid network congestion.

Hardware limitations may be avoided by either upgrading the hardware of the host machine or preventing how much load the organization will put on one sole machine. This will vary based on the company's needs; would they rather upgrade the

hardware so that they can run multiple virtual machines on one system or can they get the same out of distributing the work across multiple hosts? If cost is a concern, organizations may consider only upgrading a select few hardware components that they use solely for extensive procedures, while using the limited hardware for smaller scale operations (Bigelow, 2019).

Organizations can increase security for their virtual machines by implementing monitoring and management tools, similar to how they likely use tools to monitor/manage their physical servers. Companies should also be weary of what third party applications they use for their environments as they could potentially contain malware. Many of the security monitoring tools that exist for virtual machines can whitelist software and even quarantine a specific instance to prevent further data loss or spread of a virus (Prescient Solutions, 2018).

Implementation/What Languages Offer this Service

Programming language virtualization was first used in 1966 with the introduction of the Basic Combined Programming Language (BCPL). The C programming language took a lot of inspiration from this architecture. Other languages that used the BCPL virtualization include Pascal and Smalltalk. After this architecture, virtualization went quiet for the next 30 years.

In 1996, the introduction of the Java platform revived virtual machine programming again. The popularity of Java became the leading choice for organizations, especially those who wanted to implement virtualization. This was known as the JVM architecture. While originally designed for Java, as time went on other

languages were able to implement JVM's capabilities such as Python, Ruby, Groovy, and Pascal. Support for multiple languages is one of the key elements of the Common Language Infrastructure (CLI), which is included in the .NET framework. As of right now, JVM and CLI are the two most popular choices for program-level virtualization.

Both JVM and CLI are stack based, meaning that operations are performed based on an execution stack. The byte code contains instructions that add operands to the stack, execute the instructions, then add the results to the stack. Any calls to specific methods and calls to objects/classes are also added to the stack. An advantage of this implementation includes easy interpretation with the use of a lexical analyzer. Hence, it's easily implemented with other architectures (Buyya et al., 2013).

Conclusion

Virtual machines have stood the test of time and have proven that they can provide usefulness to those that utilize them.They provide many great advantages such as emulating a particular architecture to save on cost for compatibility, testing, and many other purposes. While there are issues that need to be addressed before considering implementing virtual machines into your work environment, they can be easily fixed with proper management.

References

Bigelow, S. J. (2019). *5 common virtualization problems and how to solve them.* SearchServerVirtualization. Retrieved from https://searchservervirtualization.techtarget.com/feature/Solutions-to-the-five-most-common-problems-with-virtualization.

Buyya, R., Vecchiola, C., & Selvi, S. T. (2013). *Mastering cloud computing: Foundations and applications programming*. Morgan Kaufmann.

Prescient. (2018). *Address these risks to ensure your virtual machines are secure.* Prescient Solutions. Retrieved from https://www.prescientsolutions.com/blog/address-these-risks-to-ensure-your-virtual-machines-are-secure/.

Scott, M. L. (2015). *Programming language pragmatics (4th Edition)*. Morgan Kaufmann, an imprint of Elsevier.

What are the common challenges of virtualization? TechAdvisory.org. (2019). Retrieved from https://www.techadvisory.org/2019/05/what-are-the-common-challenges-of-virtualization/.