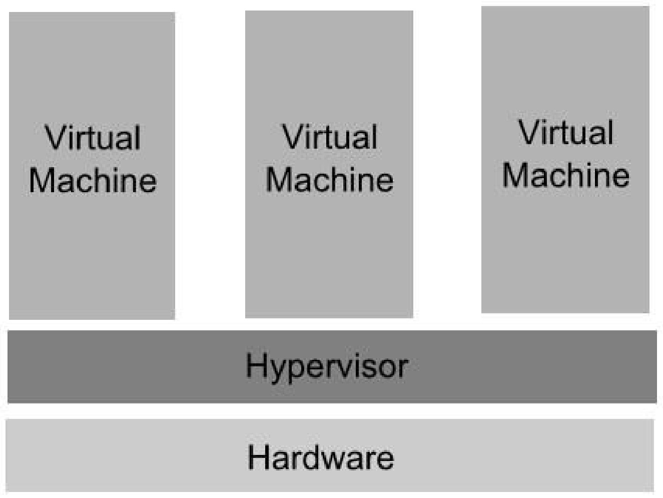
Bryan Burkhardt / xmv643
CS4663
28 Sep 2017
Assignment 1

1. **What is paravirtualization?**

   Paravirtualization is where the guest operating system is recompiled prior to being installed on a virtual machine. This technique allows the guest operating system to better utilize the host operating system hardware to allow for faster execution of performing operations. Paravirtualization also creates an interface for the guest operating system which is very similar but not identical to its host. Another feature of paravirtualization is that the virtual machine does not need to trap privileged instructions.

2. **What is type 1 virtualization and what does the architecture of type 1 virtualization look like?**

   Type 1 virtualization is when the hypervisor runs directly on the system hardware as opposed to type 2 where the hypervisor runs on the host operating system. It is believed that type 1 virtualization is more efficient and secure than type 2. Type 1 makes it feel as if the guest operating system is just another process and not a virtual machine.



Credit: See Source 1

3. **What are necessary conditions for an architecture to be virtualized (think about sensitive instructions, privileged instructions, etc.)What are the necessary hardware support on x86 platform for hardware virtualization? (e.g., CPU extensions, paging support, device virtualization support, etc. )**

   To accomplish this, VM's have two modes, a virtual user mode and virtual kernel mode; Both of these run in real user mode. Something these virtual modes allow the VM to do is what is known as a trap and emulate. This is where the VMM traps a privileged instruction, analyzes it, and executes the operation done by the guest. Once this is done, the VMM returns to the user mode and use of the VM continues. A common way to handle something like trap and emulate is by using nested page tables. With nested page tables, the guest maintains its page tables which it then translates from virtual to physical addresses. Once this is done, the VMM maintains the guest nested page tables and represents their current state to the CPU. Whenever there is a change to a page table on the CPU, the CPU informs the VMM so the VMM can update the nested page tables to reflect the changes.

**Sources:**

1. http://www.virtuatopia.com/index.php/An_Overview_of_the_Hyper-V_Architecture