15-440 Homework #4
Kesden/Fall 2013

Virtual Machines

1. Please identify and describe three way that virtual machines ease the implementation and management of distributed systems

   They provide a common environment, even across heterogeneous hardware, easing roll-outs and upgrades. They enable heterogeneous environments which need not be the same as the underlying environment, or any portion thereof, easing, for example, backward compatibility and decoupling the upgrade and migration schedules of different projects. They provide knobs for managing resources, such as changing the number of hosts, migrating work, and limiting the resources consumed by an encapsulated user.

2. What are the differences and relative advantages and disadvantages of simulators, emulators, and virtualizers?

   Simulators provide excellent visibility and detail for debugging and architectural research. Emulators enable a compatible environment while also enabling optimization that isn't loyal to the original internals. Virtualizers take advantage of the ability to directly execute certain runs of non-privileged instructions on the native hardware, without the cost of simulation or emulation.

3. What is a hypervisor? What role do they play in the management of computing.

   Hypervisors are essentially a glue layer between the host operating system and the guest VMs. They manage the resources across the VMs to ensure that common physical resources are shared efficiently, effectively, and safely. In so doing, they make it possible to manage VMs on hardware and keep them playing nice with each other.

4. Consider computation, disk I/O, network I/O, human input such as via keyboards and mice, and GPU computation. Please rank these in terms of the penalty encountered through virtualization on same-architecture virtual machines (consider VMware or KVM, if you'd like). Then, please explain the underlying reasons that each type of function experiences the relative VM penalty that it does.

   GPU computation will encounter the greatest penalty. It is real and unique hardware and can't be efficiently simulated on a general-purpose processor. Making use of esoteric hardware requires specialized code. And, since there are many different GPU designs and GPU board implementations, it is hard to have one solution that is generally useful, which increases the cost-utility curve.

Disk and network I/O require the use of real hardware, which is protected by the kernel and requires supervisor mode. This can't be emulated. Additionally, they require that data be copied or mapped, which is a cost. It can't just be left outside the guest VM in the host. So, these are both pretty rough. The relative penalty here probably depends a lot upon the speed of the network and the relative efficiencies of accessing the caches.

CPU penalty is the smallest as the CPU is the easiest to virtualize and emulate protection issues are rare in pure computation.