Brian Ackley

CSE 565

Spring 2014

Final

* 1. A hypervisor is some type of software or hardware that lets a computers operating system to run a virtual machine. The hypervisor acts as a supervisor by acting as ring 0 and acts as it is the operating system for the system and gains the same privileges as the original operating system does. The hypervisor is different than the actual supervisor, as the supervisor is the actual kernel of the original operating system which controls and regulates routines that go on through the system.
  2. Native ISA execution of the virtual machine is the virtualization of the virtual machine on that of the hypervisor, the virtual machine acts as it has the privileges of the native kernel. This allows the virtual machine to run using that of the computer’s hardware without any issues and minimal slow down of the system. The emulation of the ISA is taking the ISA that the virtual machine is sending and then converting it to an ISA that the native machine can understand. It’s doing this all while running the virtual machine and can be hard for the system to keep up without losing performance. The emulation requires more work than that of having it run native off the kernel.
  3. The hypervisor must be able to boot the system, since it is booting up an operating system or virtual machine of some sort. It must be able to regulate access to multiple users that could be on the machine and distinguish between them. The hypervisor must also be able to have access as the supervisor does and be able to distinguish any instructions that the supervisor would be able to handle.
  4. Bare iron refers to the hypervisor running directly on the hardware to control it and run processes. This is also known as type 1 hypervisor or native. The hypervisor runs between the hardware and the standard operating system of the machine.
     1. Hyper-V is an example of a program that runs bare iron. Hyper-V is installed onto the physical computer that is being used and is run right on top of the hardware and below the kernel operating system. This allows for it to control the hardware resources and to manage guest operating systems. A type of hypervisor that is not like that of bare iron is one that would not be one that would be hosted, it would run on a second layer while the operating system runs on a third layer all above the hardware.
     2. Virtual machines will have some overhead some is needed for the machines to power on and run but sometimes overhead can deplete resources as well. Sometimes they will try to take in resources when they don’t even need them anymore.
  5. The host operating system kernel executes a non virtualized environment in ring 0 where both the system kernel and the virtualized machine are both at. The execution of the operating system would be in ring 1 which is where the instructions would be handled. Since both have ring 0 privileges the guest instructions are then ran in ring 1 and then the hypervisor which is in ring 0 has the privileges to take over.
  6. The GPU module would be one of the most important because when it comes to virtualization the GPU has multiple cores on it much more then the processor so it can be able to compute much more at a better rate. Being able to utilize the performance from the GPU the virtual machine will have much more memory resources to use and this will allow the machine to run more at a more steady state.

2.1 An exception is an event which occurs during execution of a program which disrupts the normal flow of the programs instructions. An interrupt is a signal from a device or software that causes the main program that operates the computer to stop and think about what it should do next. An interrupt is masked when it has been disabled or when the CPU has been instructed to ignore it. A non-masked interrupt cannot be ignored and is used only for critical hardware errors.

2.2 Reentract code is a routine that can be used by multiple programs simultaneously. It does not modify itself in any way as well. This means that this can be used among the operating system and other software throughout the system and multithreading. This is important because one copy of the code or routine can be shared among any number of users and processes.

2.2.1 function callback

If j is in use then wait

When j is avaialabe lock mutex

run j

return

unlock mutex

2.2.2 function(callback j)

lock mutex;

j();

unlock mutex;

This code basically makes a function to call an item but while its grabbing the memory locations that it needs its locking access so that no other thread can access it until it is done. And after it is done it then unlocks that exclusion for the next instruction to access.

2.3 If an interrupt happens under a hypervisor the hypervisor will have the access to the kernel or that of the supervisor to take care of the interrupt if it’s on the same physical machine. If the hypervisor is like that of type 2 and is not on the physical memory then there’s going to need to be some type of conversion between the ISA of the hypervisor between that of the supervisor or main kernel.

3.1 U =P[ i] + P[j];

for (x = 0; x =< U.size(); x++)

{

w[i] = x[i] =< x[j];

w[j] = x[i] > x[j];

}

W[i] = i + 1;

W[j] = j + 1;

3.2 The numerical values in the chart are irrelevant because the fact is that the numbers are based off of the one thing that is uncontrollable to the data transfer and that’s the length or distance it has to travel considering that the speed is at the constant speed of light.

3.2.1 The best was at 78.65 million requests, the worst was at 1.97 million requests and the average was at 18.07 million requests.

3.2.2 It will use as many minor cycles as it can take but there would be so many request that no matter what there will always be some that need to idle.

3.3 the best carbon emissions on the table 2 is 1200kg so with average emissions for a typical vehicle at 336g/km that would be .336kg/km so it would need to travel a distance of 3,571.43km to equal the average carbon emissions of 1200kg.