CS 677 Spring 2019: Homework 2

Solutions

March 12, 2019

1. **Why is it important to respect cache affinities of processes and threads in multiprocessor scheduling?**

   In multiprocessor architecture, not all caches are shared across different processors. If a process runs on different processors every time it is scheduled, it will cause a lot of cache misses and lead to performance degradation while fetching code and data from memory.

2. **Explain in a few sentences as to why distributed scheduling will not provide much benefits at both light and heavy utilization levels.**

   At light utilization, distributed scheduling does not help much because it's better to run the new process locally. There are CPU cycles on the local machine that the process can use.

   At heavy utilization, it's not beneficial to try and move processes to other machines because it's likely that the other machines also have high load. In this case, it becomes better to run the process on the local machine.

   Distributed scheduling is only beneficial when the probability of the local host being busy is high, and the probability of finding another machine that's idle is equally as high.

3. **Why does type 1 hypervisor not need a host operating system when booting up? Explain in 2-3 sentences.**

   Type 1 hypervisor runs on bare metal. The hypervisor itself acts as the operating system, performing all functions of the operating system such as network management, CPU management, scheduling, etc. It's a combination of OS functionality and virtualization.

4. **Does a Type 1 hypervisor using paravirtualization need special hardwarde support from the CPU like normal type 1 hypervisors? Why or why not?**

   No. For normal type 1 hypervisor, the hardware support is needed to trap sensitive operations by guest OS. For type 1 hypervisor using paravirtualiztion, all sensitive operations are already replaced with hypercalls. Hardware support is no longer needed because now there are no sensitive operations to be trapped.

5. **Does Docker use a form of hardware-level or OS-level virtualization? Explain your answer in 2-3 sentences.**

   OS-level virtualization.

Docker creates containers which are multiple user-space instances running in isolation. Instead of providing simulated hardware, docker provide applications with emulated OS-level interfaces. The container does not run an entire operating system but relies on the underlying OS for performing OS functions.

6. **Why is OS virtualization more lightweight than hardware virtualization?**

   OS virtualization does not run an entire OS inside the VM, but rather relies on the underlying OS for OS functions. The kernel is shared across all containers in OS virtualization as opposed to different kernels in hardware virtualization.

7. **What is the primary difference between process and code migration?**

   Code migration only transfers the code from one machine to another and then the code is executed from the beginning on the other machine. Process migration is used to transfer a process that is already running. Therefore, both the memory state and the resources associated with the sender need to be migrated to the receiver.

8. **Why does VM migration not cause active network socket connections of processes to break even though the IP address of the physical machine does not move with the VM?**

   The VM has a logical Network Interface Card (NIC) that's independent of the physical NIC. The IP address can also be moved along with the VM.