CMPSCI 677 Operating Systems **Spring 2019**

Lecture 6: February 8

Lecturer: Prashant Shenoy *Scribe:* ***Xingda Chen***

6.1 How to Virtualize Disk/IO

A machine has a disk. Virtual machine wants to read/write file from a disk. Multiple VM would want to share/store space on the same disk. Hypervisor creates an abstraction of a virtual disk. Multiple virtual disk maps to the real disk. Virtual disk is created usually as a large file?this large file is presented as a disk to the virtual machine.

Q: how much overhead does the virtual disk adds to the system?
A: Virtual disk adds overhead to the system. Having a real disk, the OS directly read/ write blocks to the disk. In the case of virtual disk, the hypervisor is reading/writing files as it is like a real disk. In order to do so, request has to be send to the file system. The file system will then invoke the driver and the driver read/write to the corresponding parts of the block. The overhead in this case non-trivial.

Q: In non-virtualized environment, does OS need to pay the overhead of address space virtualization?
A: OS takes a disk and gives an illusion as files and folders. When read/write to the file, the OS translate the requests to actual blocks to the corresponding location. Storage space management is done by the OS, there is no overhead beyond this. One can also create virtual network interface card. Logical interfaces are constructed and mapped to the physical NICs. It is the same technique used by the hypervisor.

6.2 OS Level Virtualization

Use native OS interface to emulate another OS interface to mimic the system call using the native system. Lightweight virtual machines have no hypervisor. There exist containers that are light weight VMs which are constructed using OS level virtualization. This is a virtual machine that allows user to take resources from physical machine and construct containers. Each container can run one or more applications. All containers are managed through native operating systems. Difference between OS and hardware level virtualization: hardware level virtualization has the abstraction of the hardware, which is emulated by the virtualization layer, first have to run the OS in the VM and then the application runs in there. In OS level virtualization, when there are containers, OS doesn't run in VMs. The OS is the same as what it runs on the actual machine, the containers runs on the native OS. Applications run in OS level virtualization are "sand-boxed", which means that don't see other running applications and they only see the resources allocated in their only containers.

Benefits of putting applications into containers: 1: Isolation properties, better CPU usage allocation
2: Security, limiting which files an application can see
3: Good way to distribute software

Q: Does a malicious application in a container corrupt the host OS?
A: Yes, a malicious application can corrupt the host OS. In hardware level virtualization this is not the case: the OS is running inside the VM, if one VM is down the other VM continues to run. OS virtualization has less security than hardware virtualization–every VM gets its own OS.

Linux containers is lighter weight comparing to hypervisors, it is fast provisioning due to a simpler structure. The container can emulation different OS interfaces. There is only one scheduler which resides in the OS. The scheduler in the OS ensures each container only gets its assigned CPU time.

6.3 OS Mechanisms for LXC

OS mechanisms are for resource isolation and management: namespaces and Cgroups. Namespaces mechanism is process based resource isolation and Cgroups (stands for container groups) allow to specific limit and priorities. Other Linux built-in features:

chroot: changes the root directory to a user specified directory.

Namespace: A subset of actual resource that limits what a container can see. The resources are divided into name spaces. One namespace contains some certain processes. Take processes in the system and divide them into groups, which become part of the container. It is a way to limit what a process can see?two processes that is not inside a namespace cannot see each other. The mechanism to make a namespace is call Cgroup. Example: controlling CPU allocation?set an upper bounce on how much CPU is allocated into each namespace. Can also do it through share- base scheduling, which allocate a weight to a container and CPU time is allocated to the containers in proportion to the weight. In the situation when the container is not utilizing the assigned CPU time, hard allocation will just waste the CPU time and fair resource distribution will redistribution the cycle accordingly, it is also called work concerning scheduler.

6.4 NIC Virtualization

The IP assigned to the virtual machine is independent of the underlying machine. It has its own logical interface to emulate ethernet cards. It uses the underlying OS's ethernet card.

PlanetLab: Virtualized architecture used for research by students in different locations.