IT 3300 : Virtualization

What it is

Driving forces again

- Underutilized hardware
- Space
- Energy costs
- Administration costs

Categories of VZ

- Client (vmware desktop, virtualbox, parallels)
- Server (this is what we will focus on)
- Storage (SAN, NAS) (Storage class next semester)
- Network? (SDN)

Types of Virtualization

- Operating system Virtualization
- Hardware emulation (or Full virtualization)
- Paravirtualization

OS Virtualization

- Self-contained v environments make an application think it has access to whole host OS.
- Each has it’s own file system, process tree, network, and system libraries.
- Sometimes this is referred to as containers; (LXC, Docker, Jails?)
- i.e. each container thinks it is its’ own os
- On same physical machine (with shared underlying OS)

OS Virtualization Continued

- Files that are used by multiple containers aren’t duplicated. (The underlying OS will store them only once)
- Has greatest performance of any VZ solution.
- Also supports the highest VZ density (highest number of vz environments for a given piece of hardware)
- Great for isolating applications
- Licensing may be cheaper

OS Virtualization Continued

Disadvantages:

- Each container reflects the configs of underlying os
  - If the underlying OS is centos, all the containers are as well (same kernel, patches, type, etc...)
  - Many applications might have differing patch needs... cannot change this just for one app. (Might have to have a second physical server with other os and containers)
- All the containers rely on the base OS, if it dies, so do all the containers. (have they fixed this yet?)

OS Virtualization Continued
Use cases:
- datacenters: give each customer their own ‘server’ (web hosting)
- training: students needing own environment without breaking others?

**Hardware VZ**

- Virtualization software (hypervisor) (HV) emulates a hardware environment for guest systems.
  - VM’s interact with the emulated environment (Virtual Machine Monitor)
  - Vms are not installed on ‘physical’ machine, but on ‘virtual’
- Image can be migrated from one HV to another. (Even if HV is on other physical machine)

In other words... the x86 architecture doesn’t provide a convenient way for multiple OS’s to simultaneously run. So Hardware VZ (or full-emulation), vz software intercepts every call to system resources and manages those calls by passing them to the underlying hardware. This software is referred to as a Hypervisor!

**Hypervisors**

- A.k.a virtual machine manager
- The term hypervisor is a variant of supervisor, a traditional term for the kernel of an operating system: the hypervisor is the supervisor of the supervisor (wikipedia)

**Hypervisors Types**

- See this image


**Hardware VZ continued**

- Dissimilar OS’s can be run
- Great for server consolidation

**Hardware VZ continued**

Disadvantages:

- Slightly slower (than running on physical hardware) because everything has to be processed by VMM (Hypervisor), (Even with bare metal approach)
- Hypervisor must contain driver for hardware
  - so hardware support is limited to whatever the hypervisor understands and has drivers for.

**Hardware VZ Examples**

- VMWare Esxi, HyperV? (Bare Metal)
- VirtualBox, Qemu, VirtualBox, VMWare Workstation (hosted or type2)

**Paravirtualization**

The entire hardware environment is not emulated. Software coordinates all guest VM activities with the underlying hardware. (Multiplexes)

- Less overhead for performance (Remember that full emulation had a thin layer between)
Paravirtualization Continued

With new VZ enabled chips, we can paravirtualize (PVZ) windows machines.

In Full VZ the guest OS is *unaware* it is running virtualized. In Para, the guest OS is aware and has drivers that instead of issuing hardware commands that need to be binary translated and captured by the Hypervisor, simply issue commands that the Hypervisor understands.

So, PVZ will have to coordinate less because the Host OS’s are coordinating nicely with one another.

Paravirtualization Examples

XenServer, HyperV?

Storage VZ

- Exploding DATA!!
- Don’t want a bottleneck of putting all on 1 machine (hence: vz)
  - redundancy, failover, availability
- backup data!

How to store data

- Directly attached
  - local disk
- NAS
  - machine on your network specifically for storage
  - high amount of net traffic to device (problems?)
  - if NAS goes down... can’t get storage
- SAN (next page)

SAN

- Don’t use host network
  - Use own dedicated network (Fiberchannel or ISCSI)
- Easy to grow
  - Highly available, redundant

Failover

- Quickly starting up a vm when one has crashed
  - mission critical
  - losing money?
  - Don’t want a single point of failure
- Also called [clustering](#) or [high availability](#)

How?

- Mirror copy of vm somewhere else
- Easy to do
- Resurrect state of vm:
  - hardware representations
  - external connections (storage and ip add)
  - what settings vm is using
• Hypervisor monitors and manages failover

High Availability

• Failover is nice, but what if underlying hardware fails?
• Multiple hardware used, multiple hypervisors coordinate.
  ◦ have to move state of vm too.

Load Balancing

• Running multiple instances of same vm on separate hardware and dividing the workload between.
  ◦ Protects against SPOF
  ◦ Better use of machine resources
• Could even duplicate networks
• Must have virtualized storage.

Server Pooling

• Allows you to automatically get failover and load balancing.

How does VZ help with disaster recovery?