Revised: 2/9/2020

Server Deployment

Michael C. Hackett

Computer Science Department

Community College of Philadelphia

Lecture Topics

- Server Hardware
- Server Virtualization
- Server Storage
  - SCSI Configuration
  - RAID Configuration
  - ZFS Configuration
  - BTRFS Configuration
- Server Installation
  - Resolving Problems During and After Installation
- System Rescue

Server Hardware

- While Linux can be installed on workstations, it can also be installed on specialized server hardware.

- Servers can come in many shapes and sizes, but are usually in the form of **rackmount servers**, which are stored in a vertical **server rack**.

Server Hardware

Rackmount Server
(also called a Blade Server)

Server Racks

Server Hardware

- In addition to servers, racks may also contain…
- Storage Area Network (SAN) devices
  - Provides large amount of storage space for servers
- Uninterruptable Power Supply (UPS) devices
  - Provides battery backup power in the even of a power failure

Server Hardware

Typical SAN device

Typical UPS device

Server Hardware

- The height of rackmount devices are measured in **rack units** (or **U** for short).

- **1U** = 1.75 inches
- **2U** = 3.5 inches (2 * 1U)
- **3U** = 5.25 inches (3 * 1U)
- and so on…

Server Hardware

1U

2U

3U

4U

Server Virtualization

- **Virtualization** is the process of running more than one operating system at the same time on a single computer.

- Software that allows the computer to host multiple, simultaneous operating systems is called a **hypervisor**.
  - Oracle VirtualBox is an example of a hypervisor.

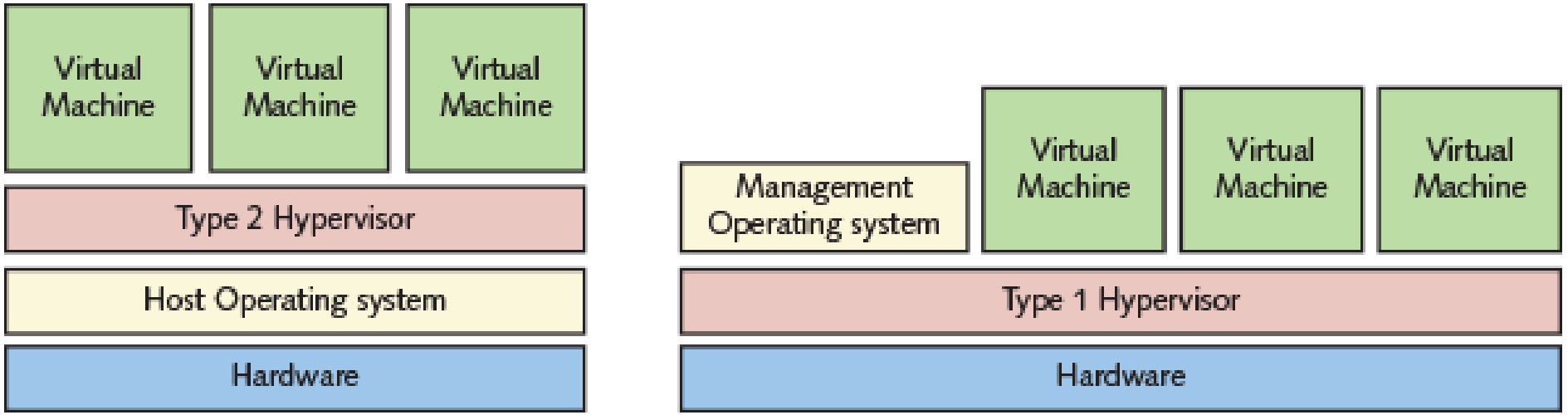
Server Virtualization

- More specifically, VirtualBox is an example of a Type 2 Hypervisor.

- A **Type 2 Hypervisor** runs on an existing (*host*) operating system and the other operating systems (*virtual machines*) access the computer's hardware via the hypervisor.

Server Virtualization

- A **Type 1 Hypervisor** interacts with the hardware directly and has its own operating system that manages its configuration and its virtual machines.



Server Virtualization

- You are already familiar with the fact that virtual machines utilize virtual hard disk files to store data.

- Virtual hard disk files can use thin provisioning or thick provisioning
  - **Thick provisioning** – The virtual hard disk file is set to a fixed size
  - **Thin provisioning** – The virtual hard disk file is dynamically allocated space as it is needed (What we have been using in our virtual machines)

SCSI Configuration

- Small Computer System Interface (SCSI, “scuzzy”) connects multiple devices for easy scalability

- Parallel SCSI
  - Older technology; still (but rarely) seen on older systems
  - SCSI devices are linked together in a chain using a single cable

SCSI Configuration

- Serial Attached SCSI (SAS, “sass”)
  - Replacement for Parallel SCSI
- A SCSI controller card can support up to 65,535 SAS hard drives
- SAS devices must be connected to the controller
  - The controller can be manually configured in the SAS controller’s BIOS

RAID Configuration

- Though disks can be broken up into partitions, if the drive has a hardware failure then all data on the drive may be lost.

- Servers typically want to use a fault tolerant configuration to prevent downtime in the event a drive fails.

- A common implementation of this is using a **Redundant Array of Independent Disks (RAID)**

RAID Configuration

- There are seven different RAID configurations
  - RAID 0 through RAID 6

- In a **RAID 0** configuration, two or more disks work together to store and retrieve data but it does not provide fault tolerance.
  - RAID 0 can be implemented with *spanning* or *striping*.

RAID Configuration

- In a RAID 0 configuration that uses *spanning*:
  - Two or more disks are recognized as a single volume
  - Useful if you need a large amount of space in a single volume
  - Also known as *Just a Bunch of Disks* (JBOD)

- In a RAID 0 configuration that uses *striping*:
  - Data is broken up spread across two or more drives
  - Useful if you need to speed up disk access (writing and reading data)

- In both RAID 0 configurations, there is **no** fault tolerance.
  - If one disk fails, all data is lost.

RAID Configuration

- In a **RAID 1** configuration, the same data is written to two or more disks simultaneously.
  - Also called *mirroring*
- If one disk fails, the other disks still have the identical data
  - The broken disk can be replaced without loss of data
  - This makes RAID 1 fault tolerant
- Drawback is cost
  - Twice the amount of hard disk space needed must be purchased.

RAID Configuration

- **RAID 2** is no longer used.
  - It was similar to RAID 0 and allowed error checking.
  - All hard disks today have error checking.

- In a **RAID 3** configuration, striping is used along a parity bit that indicates the location of data spread across the drives.

RAID Configuration

- RAID 3 requires at least three disks.
  - Two for data storage and one to store parity information
- If a drive fails, the parity information can be used to recover the data that was stored on the failed drive.
- If the drive with the parity information fails, then that particular drive will need to be restored from a backup.

RAID Configuration

- **RAID 4** is a variation of RAID 3
  - Offers faster access speeds as it does not need to access all disks at once

- **RAID 5**, the most common RAID configuration, uses striping with parity.
  - Instead of a dedicated disk with parity information, parity information is spread across all disks in the array.
  - If any disk fails, the lost data can be regenerated from the other disks in the array.
    - Though, if two or more fail at the same time then data will need to be restored from a backup.

RAID Configuration

- **RAID 6** is a variation of RAID 5
  - Allows up to two disks to fail at the same time
  - Thus requires a minimum of four disks

- RAID configurations are managed through
  - A SCSI/SAS/SATA hard disk controller card (**hardware RAID**)
  - In the computer's system BIOS (**firmware RAID**)
  - Software installed on the operating system (**software RAID**)

RAID Configuration

- For hardware RAID, it is configured through the BIOS of the controller card or using software provided by the manufacturer.
- Firmware RAID is typically configured through the system BIOS
  - Hardware RAID and Firmware RAID are functionally the same, only how they are configured differs.
- On Linux systems, the **mdadm** utility is used for **m**ultiple **d**evice **adm**inistration
  - A software RAID

ZFS Configuration

- ZFS (Zettabyte File System) is a high-performance filesystem
- Designed for large-scale Linux systems
- RAID-like ZFS volumes can span thousands of local and network storage devices
- Checks and repairs errors automatically

ZFS Configuration

- Storage devices are grouped together into **pools** that are managed by ZFS
- The filesystems that are created and managed from a ZFS pool is a **ZFS volume**
- The **`zpool`** utility program is used to create and manage ZFS pools and the volumes contained in them

ZFS Configuration

- **RAID-Z** is an implementation of RAID 5 in ZFS

- While **`zpool`** is used to manage ZFS pools, the **`zfs`** command is used to manage specific features like quotas and compression options.

BTRFS Configuration

- The B-tree File System (BTRFS) is similar to ZFS
- Can be used to create volumes that span multiple storage devices
- Not as fast or as robust as ZFS
  - Intended to eventually replace ext4

- When using BTRFS, it is managed using the **`btrfs`** command and can be checked for errors using the **`btrfsck`** command

Server Installation

- Most Linux servers do not have a GUI, especially those that are rackmount servers.
  - They are usually managed remotely through a terminal/CLI
  - This is why it is so important for a Linux administrator to be comfortable using the terminal
- Any Linux distribution can be used as a server, provided the necessary server software packages are installed
  - Some distributions provide a “server edition” that allows greater control over what is initially installed on the system

Server Installation

- These “server editions” or “server distributions” typically allow the following to be configured during installation:
  - Host name, IP information/settings
  - Enabling/Disabling Automatic Updates
  - Server Package/Software Selections
  - Service Configuration
  - Boot Loader Configuration

Server Installation

- Problems during installation are usually caused by faulty hardware or incorrect device drivers.

- A **segmentation fault** results when a program tries to access and unassigned area of the system memory
  - If this occurs, run the **memtest86** utility to check for errors with the computer's memory
  - Memtest86 is included with many distribution installers and can usually be selected at the installation's welcome screen
  - A segmentation fault may also result when the system's CPU is overclocked

Server Installation

- To check if there were any issues during installation, view the installation log files.
  - Location for Fedora systems: **`/var/log/anaconda`**
  - Location for Ubuntu systems: **`/var/log/installer`**

- To quickly find any issues:
  - Fedora systems: **`grep -i "(error|warn)" var/log/anaconda/*`**
  - Ubuntu systems: **`grep -i "(error|warn)" var/log/installer/*`**

Server Installation

- Some devices, like storage devices, have their device driver compiled into the Linux kernel
  - Referenced by a device file in **`/dev`**

- Other devices (graphics cards, network interface cards, etc.) have their device drivers loaded into the kernel as a **module**

Server Installation

- Module files ends with a .ko or .ko.xz extension
  - Located in **`/lib/modules/`*`kernelversion`*`/`** or **`/usr/lib/modules/`*`kernelversion`*`/`**

- Modules can be manually loaded into the kernel using the **`insmod`** or **`modprobe`** commands

- To display a list of modules currently loaded, use the **`lsmod`** command.

Server Installation

- For information about a particular module, use the **`modinfo`** command

- To manually remove a module from the kernel, use the **`rmmod`** command

- Some modules depend on others, so after installing a new module be sure to run **`depmod`** to update the system's database of module dependencies

Server Installation

- There are a number of commands that can be used to list hardware device information:

`lscpu`	Displays CPU information
`lsmem`	Displays RAM information
`lsdev`	Displays interrupts, IO ports, and DMA channels
`lspci`	Displays PCI device information
`lsusb`	Displays USB device information
`lsscsi`	Displays SCSI device information
`lsblk`	Displays block device information
`lshw`	Displays general hardware information for the entire system

Server Installation

- To view any issues or problems that occured at boot time, check the contents of the following log files (depends on the distribution):

  `/var/log/boot.log`
  `/var/log/messages`
  `/var/log/syslog`

- The **`dmesg`** command is used to view the kernel's log

System Rescue

- If a Linux system fails to boot, it is common practice to use a Live CD/DVD to perform system rescue.

- A Live OS will include tools like **`fdisk`**, **`fsck`**, and others that can be used to fix the following types of problems on a broken Linux system:
  - Boot loader problems
  - Filesystem problems
  - Configuration file problems
  - Driver problems

System Rescue

- In a Live OS environment, you can mount the problematic filesystem in the live environment

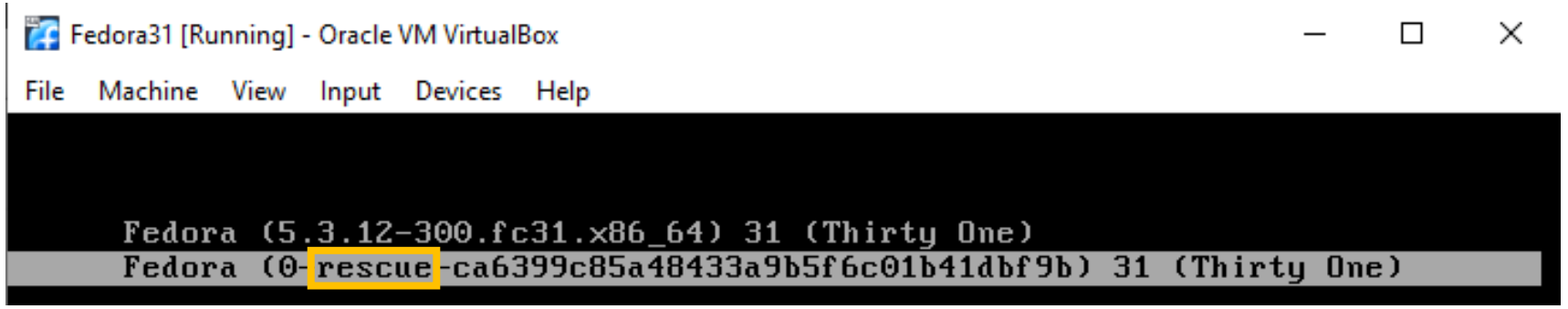
- The **`chroot`** command will **ch**ange the **root** of the current environment to installed system (after it is mounted).

  `chroot` ***`/mountpoint`***

- This will give you root access to the installed system.
  - Also useful for recovering a lost root password.

System Rescue

- Some distributions will offer a recovery option at boot.



- This will load a limited OS with utilities like **`fdisk`**, **`fsck`**, etc
- Will allow you to **`mount`** and **`chroot`** into the installed system