System Administration

Storage Systems
Agenda

- Storage Devices
- Partitioning
- LVM
- File Systems
STORAGE DEVICES
Single Disk
RAID?
[Guild] [Alinna]: so we don’t get blamed for fucking up
[Guild] [Alinna]: haha
[Guild] [Eldith]: LOL
[Guild] [Eldith]: n1.
[Guild] [Alinna]: and plus i got tests this week
[Guild] [Alinna]: so im not sure
[Raid] [Exclat]: we want him to go for tank after tank and never rush the healer/ranged grous
RAID

- Redundant Array of Independent Disks
- Software vs. Hardware
- RAID 0, 1, 3, 5, 6
Software RAID

- Parity done by CPU
- FakeRAID
- Linux md
- LVM
- ZFS, btrfs
  - Later
Hardware RAID

- RAID controller card
- Dedicated hardware box
Direct Attached Storage

- SAS interface
Storage Area Network

- Fiber Channel
- iSCSI
- ATA-over-Ethernet
Fiber Channel
Network Attached Storage

- NFS
- CIFS (think Windows File Sharing)
SAN vs. NAS

SAN
FiberChannel, iSCSI, or AoE

NAS
SMB, NFS, AFS
PARTITIONING
1 File System / Disk?
2 TB maybe... 2TB x 12?
2TB x 128 then?
Partitioning in Linux

- fdisk
  - No support for GPT
- Parted
  - GParted
Fdisk

root@vmhost:~ # fdisk /dev/vmhost/test

Command (m for help): p

Disk /dev/vmhost/test: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0005848b

Device Boot Start   End    Blocks  Id  System

Command (m for help): -
Add Partition

```
Command (m for help): n
Partition type:
  p  primary (0 primary, 0 extended, 4 free)
  e  extended
Select (default p): p
Partition number (1-4, default 1): 
Using default value 1
First sector (2048-41943039, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-41943039, default 41943039): +10G
```

```
Command (m for help): p
Disk /dev/vmhost/test: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x18d53d11

Device Boot Start  End   Blocks  Id  System
/dev/vmhost/test1 2048 20973567 10485760 83  Linux
```

Command (m for help): _
Delete Partition

Command (m for help): d
Selected partition 1
Save & Exit

```
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
WARNING: Re-reading the partition table failed with error 22: Invalid argument.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
root@vmhost:~# _
```

```
Disk /dev/vmhost/test: 21.5 GB, 21
255 heads, 63 sectors/track, 2610
Units = sectors of 1 * 512 = 512 b
Sector size (logical/physical): 51
I/O size (minimum/optimal): 512 by
```

Device Boot Start

```
Command (m for help): _
```
Parted

```
root@vmhost:~# parted /dev/vmhost/test
GNU Parted 2.3
Using /dev/dm-4
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) p
Model: Linux device-mapper (linear) (dm)
Disk /dev/dm-4: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start  End   Size  Type       File system  Flags
(parted) _
```

---

```
root@vmhost:~# parted /dev/vmhost/test
GNU Parted 2.3
Using /dev/dm-4
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) p
Model: Linux device-mapper (linear) (dm)
Disk /dev/dm-4: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start  End   Size  Type       File system  Flags
(parted) _
```
Add Partition

(parted) mkpart
Partition type?  primary/extended?  primary
File system type?  [ext2]?  ext2
Start?  1mib
End?  10gib
(parted) p
Model: Linux device-mapper (linear) (dm)
Disk  /dev/dm-4:  21.5GB
Sector size (logical/physical):  512B/512B
Partition Table:  msdos

<table>
<thead>
<tr>
<th>Number</th>
<th>Start</th>
<th>End</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1049kB</td>
<td>10.7GB</td>
<td>10.7GB</td>
</tr>
</tbody>
</table>

(parted)
## Change Units

```bash
(parted) p
Model: Linux device-mapper (linear) (dm)
Disk /dev/dm-4: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

<table>
<thead>
<tr>
<th>Number</th>
<th>Start</th>
<th>End</th>
<th>Size</th>
<th>Type</th>
<th>File system</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1049kB</td>
<td>10.7GB</td>
<td>10.7GB</td>
<td>primary</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```bash
(parted) unit
Unit? [compact]? gib
```

```bash
(parted) p
Model: Linux device-mapper (linear) (dm)
Disk /dev/dm-4: 20.0GiB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

<table>
<thead>
<tr>
<th>Number</th>
<th>Start</th>
<th>End</th>
<th>Size</th>
<th>Type</th>
<th>File system</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00GiB</td>
<td>10.0GiB</td>
<td>10.0GiB</td>
<td>primary</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Delete Partition

(parted) p
Model: Linux device-mapper (linear) (dm)
Disk /dev/dm-4: 20.0GiB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

<table>
<thead>
<tr>
<th>Number</th>
<th>Start</th>
<th>End</th>
<th>Size</th>
<th>Type</th>
<th>File system</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00GiB</td>
<td>10.0GiB</td>
<td>10.0GiB</td>
<td>primary</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(parted) rm
Partition number? 1

(parted) p
Model: Linux device-mapper (linear) (dm)
Disk /dev/dm-4: 20.0GiB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start  End  Size  Type  File system  Flags
(parted)
No need to save

- Any action you do is permanent
- Parted will try to update system partition table
Script support

- parted can also take commands from command line:
  - parted /dev/sda mkpart pri ext2 1Mib 10Gib
Resize (Expand)

1. Edit partition table
   ◦ Delete and create with same start position
2. Reload partition table
   ◦ Reboot if needed
3. Expand filesystem
Resize (Shrink)

1. Shrink filesystem
   ◦ Slightly smaller than final
2. Edit partition table
   ◦ Delete and create with same start position
3. Reload partition table
   ◦ Reboot if needed
4. Expand filesystem to fit partition
No Partition Moving
LOGICAL VOLUME MANAGER
What is LVM?

- A system to manage storage devices
- Volume == Disk
Why use LVM?

- Storage pooling
- Online resizing
- Resize any way
- Snapshots
Concepts

- Physical Volume
  - A disk or partition
- Volume Group
  - A group of PVs
- Logical Volume
  - A virtual disk/partition
- Physical Extent
  - Data blocks of a PV
Typical limits for Linux LVM v1:

- Range of PE size (p): 8KB... 512MB
- Range of PV size (v): 512MB... 2TB
- Range of PEs (N): 1... 65534
- Range of PVs (X): 1... 256
- Range of VGs (Z): 1... 99
Using a partition for LVM

- Best to have a partition table

1. Create partition with LVM type
   - Fdisk: use “t” to change type to “8e”
   - Parted: toggle “lvm” flag

2. pvcreate /dev/XXX
lvm> pvs
PV  VG  Fmt  Attr  PSIZE  PFree
/dev/sda6  vmhost  lvm2  a--  521.19g  183.19g

lvm> pvdisplay /dev/sda6
--- Physical volume ---
PV Name   /dev/sda6
VG Name   vmhost
PV Size   521.19 GiB / not usable 5.00 MiB
Allocatable  yes
PE Size   16.00 MiB
Total PE   33356
Free PE    11724
Allocated PE 21632
PV UUID   vRy78A-qBzb-OBnP-sUeP-J5Mm-KAIW-WDT8yR
Create a volume group

- `vgcreate <name> /dev/XXX ...`
<table>
<thead>
<tr>
<th>VG</th>
<th>#PV</th>
<th>#LV</th>
<th>Attr</th>
<th>VSize</th>
<th>VFree</th>
</tr>
</thead>
<tbody>
<tr>
<td>vmhost</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>wz---n-</td>
<td>521.19g</td>
</tr>
</tbody>
</table>

--- Volume group ---

<table>
<thead>
<tr>
<th>VG Name</th>
<th>System ID</th>
<th>Format</th>
<th>Metadata Areas</th>
<th>Metadata Sequence No</th>
<th>VG Access</th>
<th>VG Status</th>
<th>MAX LV</th>
<th>Cur LV</th>
<th>Open LV</th>
<th>Max PV</th>
<th>Cur PV</th>
<th>Act PV</th>
<th>VG Size</th>
<th>PE Size</th>
<th>Total PE</th>
<th>Alloc PE / Size</th>
<th>Free PE / Size</th>
<th>VG UUID</th>
</tr>
</thead>
<tbody>
<tr>
<td>vmhost</td>
<td></td>
<td>lvm2</td>
<td>1</td>
<td>58</td>
<td>read/write</td>
<td>resizable</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>521.19 GiB</td>
<td>16.00 MiB</td>
<td>33356</td>
<td>21632 / 338.00 GiB</td>
<td>11724 / 183.19 GiB</td>
<td>YPE07E-0Hd3-6Jq6-92fU-gkTR-tSNJ-rBjHfJ</td>
</tr>
</tbody>
</table>
Add PVs to volume groups

- vgextend <name> /dev/XXX ...
Create a logical volume

- `lvcreate --name <name> --size <size> <volume group>`
- Size units:
  - B, K, M, G, T ...
- Path to new volume
  - `/dev/<volume group>/<name>`
More about LVM

FILE SYSTEMS
What is a File System?

- A file system (or filesystem) is an abstraction to store, retrieve and update a set of files.
- Learn more in OS course
Linux Disk File Systems

- Ext Family
  - Ext2
  - Ext3
  - Ext4
- ReiserFS
- XFS
- ZFS
- Btrfs
Ext Family

- Ext replaced MINIX file system
- Ext2 was major overhaul
- Ext3 adds
  - Journaling
  - Online expand
  - Htree directory index (was linked-list)
Ext Family cont.

- Ext4 adds
  - Larger file system (> 2TB)
  - Extents
  - Journal Checksum
  - Increase subdirectory limit
  - Delayed Allocation
  - Persistent Pre-allocation
  - ...
Ext Family cont.

- Default FS in many distros
ReiserFS (Reiser3)

- By Hans Reiser / NAMESYS
- Earlier than ext3
- Fast for huge directories and small files
- Features
  - Journaling
  - Online resize
  - Tail packing
XFS

- By SGI
- Opensourced and ported to Linux
- Fast for large filesystems and large files

Features
  - Journaling
  - Extents
  - ...
ZFS

- Filesystem + LVM
- By Sun
- Open-source but license not compatible with Linux
- Available on
  - Solaris
  - FreeBSD (older/slower than Solaris)
  - Linux using FUSE or “ZFS on Linux”
- Basis for Sun Open Storage
Btrfs

- Started by Oracle
- *Experimental*
- Comparable to ZFS
Recommendation

- Use ext4 or XFS
- See manpages for tunables
  - Features
  - Block size
  - RAID stripe size / stride width
  - Journaling level
How to create a filesystem

- `/sbin/mkfs.<type> /dev/XXX`
  - Type: ext2, ext3, ext4, xfs, reiserfs, ...
- ZFS and btrfs have special tools
Filesystem Utilities

- Ext Family
  - e2fsprogs
- ReiserFS
  - reiserfsprogs
- XFS
  - xfsprogs
How to mount a filesystem

- A filesystem is mounted at a “mount point”, a directory
  - Contents of that directory are hidden
- mount /dev/XXX /mnt

- mount
  - Shows currently mounted filesystems
How to unmount

• unmount /dev/XXX
• unmount /mnt
Automatic mounting

- Add entry in /etc/fstab

```
# /etc/fstab: static file system information.
#
# Use `blkid` to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system>  <mount point>  <type>     <options>     <dump>  <pass>
proc      /proc      proc defaults    0       0
#/ was on /dev/vda1 during installation
/dev/vda1  /        ext4  noatime,errors=remount-ro 0       1
/dev/vda2  /home    ext4  noatime     0       2
```

- Will mount during boot
root@vmhost:~# mkfs.ext4 /dev/vmhost/test2
mke2fs 1.42.5 (29-Jul-2012)
Filesystem label= 
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
655360 inodes, 2621440 blocks
131072 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2684354560
80 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks: 
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632
Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
root@vmhost:~#
Practice

- Use a spare HDD or flat file
  - Create a 10G flat file with `dd`:
    - `dd if=/dev/zero of=XXX bs=1M count=10k`
- Play with `fdisk/parted` and LVM
- Go through the docs or Wikipedia
  - Learn the features
Homework: Write a script

- Given 2 HDDs /dev/sdb /dev/sdc

- Use parted in script mode to create LVM partitions on both HDDs
- Add them to LVM volume group
  - Use your student ID as the name
- Create a 20G LV named ID-1 and format as ext4
- Create a 20G LV named ID-2 and format as XFS
- Undo (delete everything)

- Be sure to try it SAFELY before submitting
- Do not include any useless commands