CS447/647

Storage

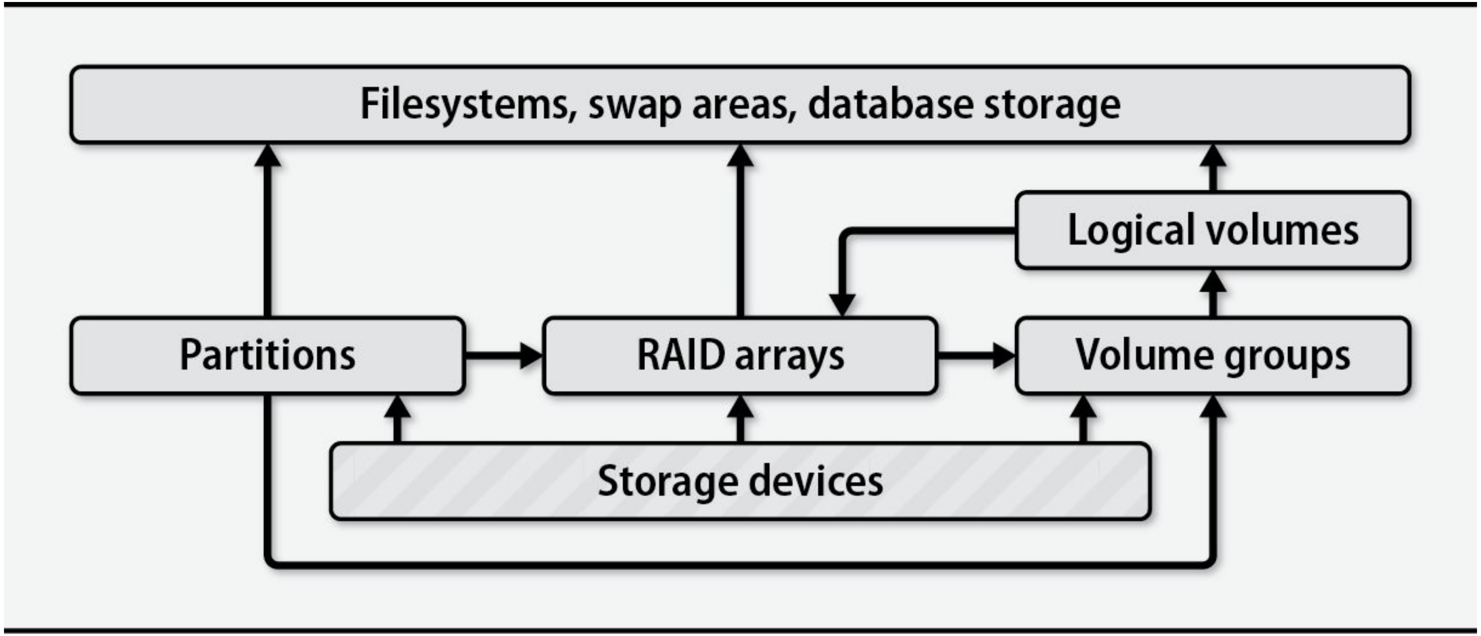
LVM and RAID

	Cloud Provider 1	Cloud Provider 2	Cloud Provider 3	WhiteBox
Cost per GB	$ 0.02	$ 0.02	$ 0.02	$ 0.01
Ingress Cost	$ -	$ -	$ -	
Egress Cost	$ 0.08	$ 0.09	$ 0.09	$ -
Monthly Cost	$ 2,048.00	$ 2,048.00	$ 2,048.00	$ -
20T Egress	$ 1,638.40	$ 1,843.20	$ 1,843.20	$ -
1st Year Cost	$ 24,576.00	$ 24,576.00	$ 24,576.00	$ 5,688.00
5 Years Cost	$ 124,518.40	$ 124,723.20	$ 124,723.20	$ 5,688.00

References

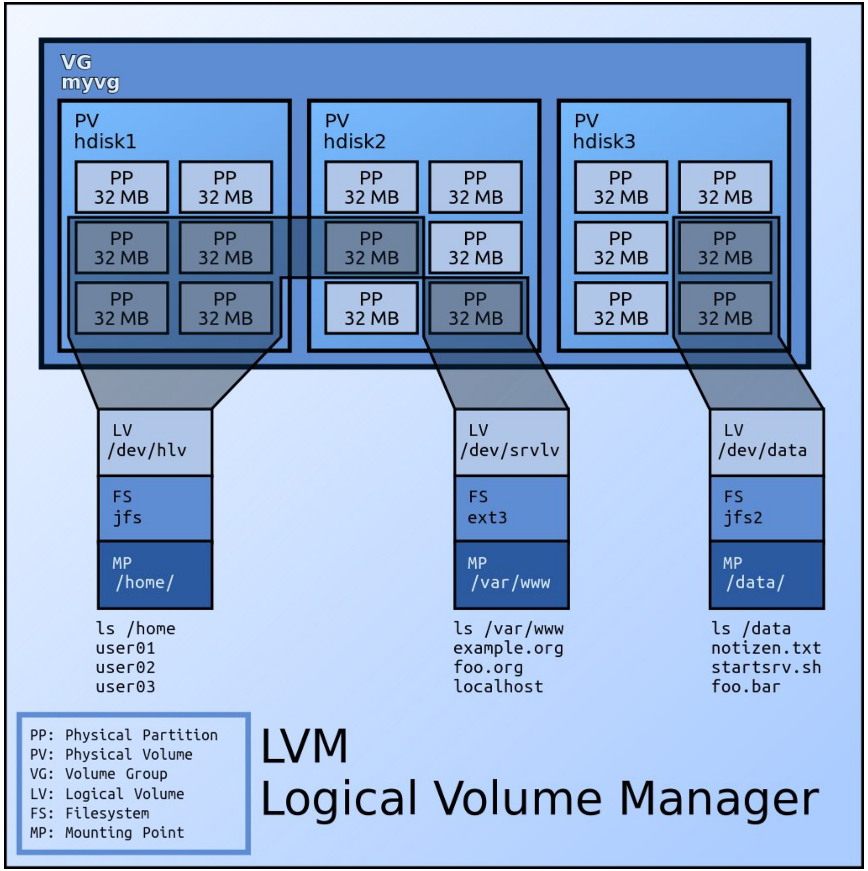
Nemeth, Evi, et al. *UNIX and Linux System Administration Handbook*. Addison-Wesley, 2018.

Remzi H., et al. *Operating Systems: Three Easy Pieces*, Arpaci-Dusseau Books, August, 2018 (Version 1.00)
https://pages.cs.wisc.edu/~remzi/OSTEP/


Filesystems, swap areas, database storage
Logical volumes
Partitions
RAID arrays
Volume groups
Storage devices

LVM - Logical Volume Management

- Provides tools to create virtual block devices from physical devices
- Virtual devices are <u>easier to manage</u> than physical devices
- Three requirements
  - Device-mapper kernel module
  - Userspace device-mapper
  - Userspace lvm2 tools
- Three components
  - Physical Volume
  - Volume Group
  - Logical Volume



https://en.wikipedia.org/wiki/Logical_Volume_Manager

LVM - Userspace Basics

```
truncate -s 5G disk1.img
truncate -s 5G disk2.img
losetup -d /dev/loop0
losetup --find --show disk1.img
losetup --find --show disk2.img
parted -s /dev/loop0 'mklabel gpt mkpart lvpart1 1M 1G'
parted -s /dev/loop1 'mklabel gpt mkpart lvpart1 1M 1G'
pvcreate -v /dev/loop0p1
pvcreate -v /dev/loop1p1
pvdisplay
```

LVM - Userspace Basics

```
vgcreate vg0 /dev/loop0p1
vgextend vg0 /dev/loop1p1
vgdisplay
lvcreate -L 1.5G -n lv0 vg0
lvdisplay
parted -s /dev/mapper/vg0-lv0 'print'
lvresize -L -.5G vg0/lv0 #Shrink
lvresize -L +.5G vg0/lv0 #Grow
```

LVM - Userspace Physical Volume Moves

```
truncate -s 5G disk3.img
losetup --find --show disk3.img
parted -s /dev/loop2 'mklabel gpt mkpart lvpart1 1M 1G'
pvcreate -v /dev/loop2p1
pvmove /dev/loop0p1 /dev/loop2p1
pvdisplay -m
```

LVM - Userspace Snapshots

```
pvcreate /dev/loop0
pvcreate /dev/loop1
vgcreate vg0 /dev/loop0 /dev/loop1
lvcreate -L 1G -n lv0 vg0
mkfs.ext /dev/mapper/vg0-lv0
mount /dev/mapper/vg0-lv0 /mnt
touch /mnt/file
lvcreate -L1G -s -n lv0-snap vg0/lv0 #Create a snapshot named
lv0-snap
umount /mnt
mount /dev/mapper/vg0-lv0--snap /mnt
umount /mnt && lvremove vg0/lv0-snap
```

RAID

- We often want disks to be
    - faster
    - larger
    - more reliable
- Redundant Array of Inexpensive Disks (RAID)
    - Developed in the late 1980’s by the CS department at Berkeley
    - Technique to make multiple disks to appear as a single disk
        - More storage, better performance and reliability
    - Complex
        - Multiple Disks
        - RAM
        - Processors

RAID

- Advantages
  - Performance
  - Capacity
  - Reliability
    - Redundancy - Tolerate the loss of a disk
- Transparency - Easing Deployment
  - New functionality
  - Demands no changes to the rest of the system
  - RAID is a perfect example
    - Looks like one big disk
  - Solved the deployment problem

Interface and RAID Internals

- Filesystem sees one big disk
- When a logical IO request is made, a RAID must:
  - Calculate which disk
  - Issue physical IO request
    - Mirroring results is 2 physical writes for 1 logical
- SATA, SCSI or NVME
  - NVME is software RAID only
    - mdraid
  - Hardware RAID for SATA and SCSI

RAID0 - Striping

- Upper Bound of Performance and Capacity
- “Perfect reliability”
  - One disk fails the whole array fails
- Excellent Performance
  - All disks are utilized
  - Often parallel
- Best Capacity
  - All Disks combined

Disk 0	Disk 1	Disk 2	Disk 3
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Figure 38.1: **RAID-0: Simple Striping**

RAID Mapping Problem

- How does the RAID map logical blocks to physical disks?
  - Logical Block A
  - `Disk = A % number_of_disk`
  - `Offset = A / number_of_disks`
- So, a write to A = 14 with a 4 disk RAID
  - `Disk: 14 % 4 = 2`
  - `Offset: 14 / 4 = 3`

Chunk Size

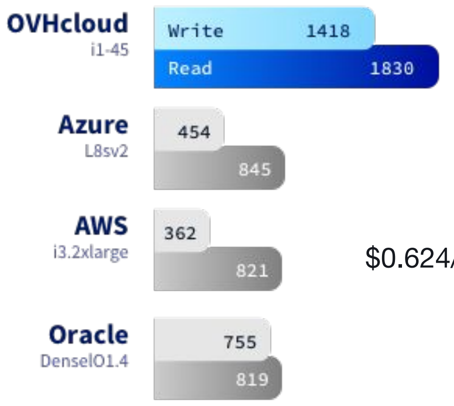
- Affects performance
  - 64Kb, 512Kb (common)
- Group blocks together on a single disk
- Small chunks means files are striped across many disks
- Large chunks reduce intra-file parallelism
- Art more than a science

RAID0 - Performance

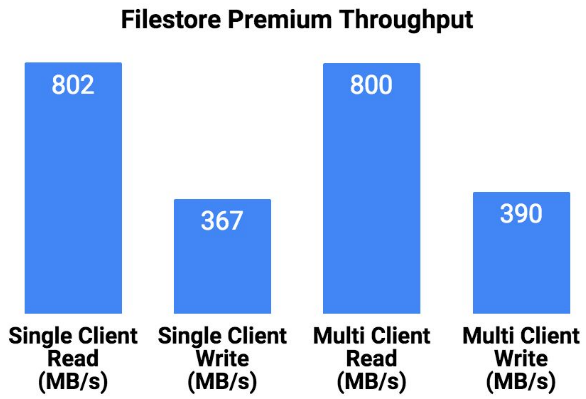
- Sequential Workload - Large continuous chunks
- Random Workload - Small requests for random disk locations (blocks)
    - Databases
- In general Sequential > Random
- Number of Disks * Random Rate
- Number of Disks * Sequential Rate
- Full Bandwidth

How do we benchmark?

- `dd(1)` - Basic Sequential Read\Write
- `hdparm(1)` - Basic buffered reads test
- `fio(1)` - synthetic benchmarks, ‘real world’ workloads



$0.624/h



RAID1 - Mirroring

- Copy of each block on a different disk
- Each logical write is two physical writes
  - Slowest of the two
  - Happen in parallel
- Sequential Performance: (N/2) * Sequential Rate
- Random Performance: (N/2) * Random Rate

Disk 0	Disk 1	Disk 2	Disk 3
0	0	1	1
2	2	3	3
4	4	5	5
6	6	7	7

Figure 38.3: **Simple RAID-1: Mirroring**

RAID4 - Parity

- Each stripe has a parity block
- Parity calculated using XOR
- Can lose 1 disk
  - Replacement has to be rebuilt
- Performance
  - (N - 1) * Rate
  - Random write does not improve when you add disks

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P0
4	5	6	7	P1
8	9	10	11	P2
12	13	14	15	P3

Figure 38.4: **RAID-4 With Parity**

C0	C1	C2	C3	P
0	0	1	1	XOR(0,0,1,1) = 0
0	1	0	0	XOR(0,1,0,0) = 1

Disk and Flash Drive Rebuild Times

RAID	Capacity TB	Capacity GB	Capacity MB	Seq Write Speed MB/sec	Rebuild Time Minimum secs	Minutes	Hours
Disk	0.72	72	72,000	80	900	15	0.25
	1	1,000	1,000,000	115	8,696	145	2.42
	4	4,000	4,000,000	115	34,783	580	9.66
SSD FlashMax III	2.2	2,200	2,200,000	1,400	1,571	26	0.44
Intel D3600	2	2,000	2,000,000	1,500	1,333	22	0.37
Micron 9100	3.2	3,200	3,200,000	2,000	1,600	27	0.44
Intel DC P3608	4	4,000	4,000,000	3,000	1,333	22	0.37

https://www.theregister.com/2016/05/13/disak_versus_ssd_raid_rebuild_times/

RAID5 - Rotating Parity

- Operates identically to RAID4
- Random read performance slightly better
- Random Write: (N/4) * R

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P0
5	6	7	P1	4
10	11	P2	8	9
15	P3	12	13	14
P4	16	17	18	19

Figure 38.7: **RAID-5 With Rotated Parity**

	RAID-0	RAID-1	RAID-4	RAID-5
Capacity	$N \cdot B$	$(N \cdot B)/2$	$(N-1) \cdot B$	$(N-1) \cdot B$
Reliability	0	1 (for sure) $\frac{N}{2}$ (if lucky)	1	1
Throughput				
Sequential Read	$N \cdot S$	$(N/2) \cdot S^1$	$(N-1) \cdot S$	$(N-1) \cdot S$
Sequential Write	$N \cdot S$	$(N/2) \cdot S^1$	$(N-1) \cdot S$	$(N-1) \cdot S$
Random Read	$N \cdot R$	$N \cdot R$	$(N-1) \cdot R$	$N \cdot R$
Random Write	$N \cdot R$	$(N/2) \cdot R$	$\frac{1}{2} \cdot R$	$\frac{N}{4} R$
Latency				
Read	T	T	T	T
Write	T	T	$2T$	$2T$

Figure 38.8: **RAID Capacity, Reliability, and Performance**

mdraid - Linux Software Raid

- RAID devices are virtual devices created from two or more block devices
- Many devices to one virtual device
- RAID Levels offer performance and redundancy
- Levels
  - **LINEAR** - concatenates devices in a single device. Like LVM Volume Group
  - **RAID0** (striping) - No redundancy, performance
    RAID1 (mirroring) - Mirror disks
  - **RAID4** - RAID0 plus a parity disk
  - **RAID5** - RAID4 with parity spread across disks, lose 1 disk
  - **RAID6** - RAID5 with two parity segments, lose two disks
  - **RAID10** - striped mirroring
  - MULTIPATH - Not a RAID. Multiple paths to same storage device. iSCSI
  - FAULTY - provides a layer over a true device that can be used to inject faults
  - CONTAINER - Set of devices

```
.config - Linux/x86 4.15.18 Kernel Configuration
> Device Drivers > Multiple devices driver support (RAID and LVM)
              Multiple devices driver support (RAID and LVM)
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y> includes, <N>
  excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help,
  </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module

      --- Multiple devices driver support (RAID and LVM)
      {*}   RAID support
      [*]     Autodetect RAID arrays during kernel boot
      <M>     Linear (append) mode
      {M}     RAID-0 (striping) mode
      {M}     RAID-1 (mirroring) mode
      {M}     RAID-10 (mirrored striping) mode
      {M}     RAID-4/RAID-5/RAID-6 mode
      <M>     Multipath I/O support
      <M>     Faulty test module for MD
      <M>     Cluster Support for MD
      <M>   Block device as cache
      [ ]     Bcache debugging
      [ ]     Debug closures
      <*>   Device mapper support
      [ ]     request-based DM: use blk-mq I/O path by default
      [ ]     Device mapper debugging support
      [ ]     Block manager locking
      <M>     Crypt target support
      <M>     Snapshot target
      (+)

      <Select>    < Exit >    < Help >    < Save >    < Load >
```

mdadm

```
mdadm -C /dev/md0 -l stripe -n 2 /dev/loop0 /dev/loop1 #RAID0
mdadm -C /dev/md0 -l raid1 -n 2 /dev/loop0 /dev/loop1  #RAID1

#RAID5
mdadm -C /dev/md0 -l raid5 -n 3 /dev/loop0 /dev/loop1 /dev/loop2 -x 1 /dev/loop3

mdadm --detail /dev/md0
cat /proc/mdstat

#Persist across reboots
mdadm --detail --scan --verbose > /etc/mdadm.conf
```

mdadm

```
mdadm -C /dev/md0 -l raid1 -n 2 /dev/loop0 /dev/loop1 #RAID1

#Fail a disk
mdadm --fail /dev/md0 /dev/loop0

mdadm --remove /dev/md0 /dev/loop0

mdadm --add /dev/md0 /dev/loop0
```

mdadm

#Fake Corruption

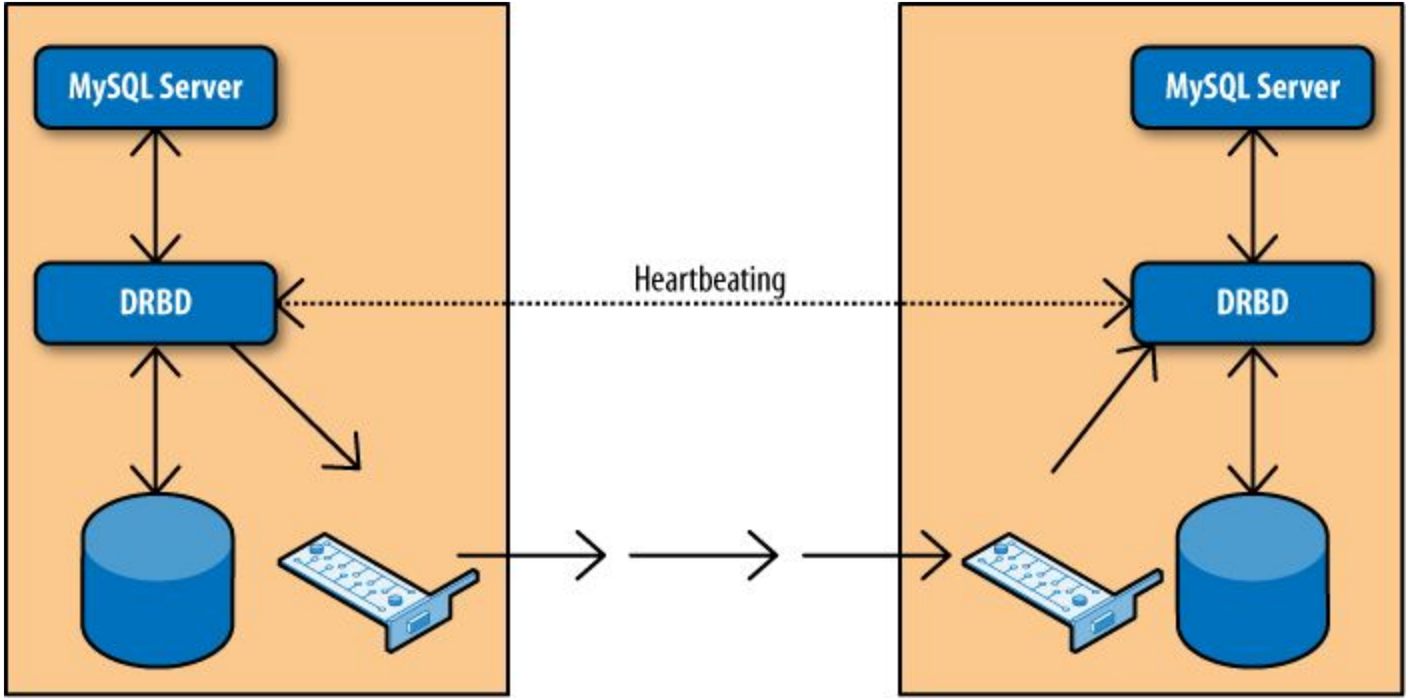
```
dd if=/dev/zero of=/dev/loop0 bs=1M count=128

echo check > /sys/block/md0/md/sync_action

cat /proc/mdstat
```

drbd

- Distributed Replicated Block Device
- High-Availability Storage
- RAID1 over the network
- Active/Passive Setup



MySQL High Availability

drbd

```
# begin resource drbd0

resource drbd0 {
     protocol C;
     startup {degr-wfc-timeout 120;}
     disk {on-io-error detach;}
     net {}
     syncer {
          rate 100m;
          al-extents 257;
     }
     on server1 {
          device /dev/drbd0;
          disk /dev/sdb;
          address 192.168.1.230:7788;
          meta-disk internal;
     }
}
```

```
on server2 {
          device /dev/drbd0;
          disk /dev/sdb;
          address 192.168.1.231:7788;
          meta-disk internal;
     }
```

DRBD

```
apt install -y drbd-utils
mkdir -p /srv/drbd && cd /srv/drbd
truncate -s 1G res0.img
losetup --show --find res0.img
#Create /etc/drbd.d/drbd0.res
drbdadm create-md drbd0
drbdadm up drbd0
cat /proc/drbd
drbdadm primary --force drbd0
```

Why does this matter?

Complex container and virtual machine managers use LVM, mdadm and drbd extensively. They are heavily scripted.