RAID & LVM

CIS 2235 Linux System Administration

Overview

Timing

- RAID is older – it has been around for a long time
- LVM is newer – it is a more recent development

Purposes:

- RAID – speed and backup protection
- LVM – ease of growth and configuration

RAID

Redundant Array of Independent Disks

- Distributes data across a set of disks to gain:
  - Speed
  - Data integrity

Speed is done by “striping” data across multiple drives.

Data integrity can be achieved with:

Mirroring
Checksum

Raid is organized by levels

RAID levels

7 levels of RAID commonly defined

each one is more complicated, powerful

only some are used in practice

level 0 Striping

level 1 Mirroring

level 3 Bit interleaved parity

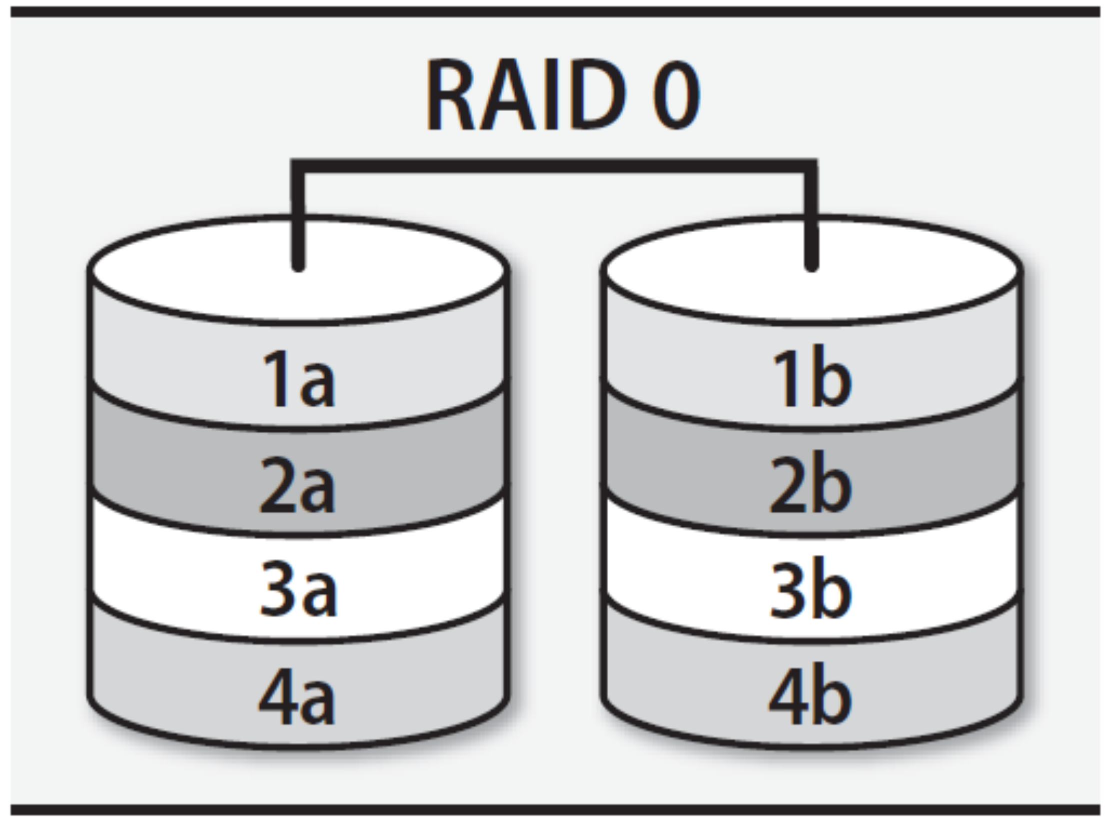
level 4 Block interleaved parity

level 5 Full RAID

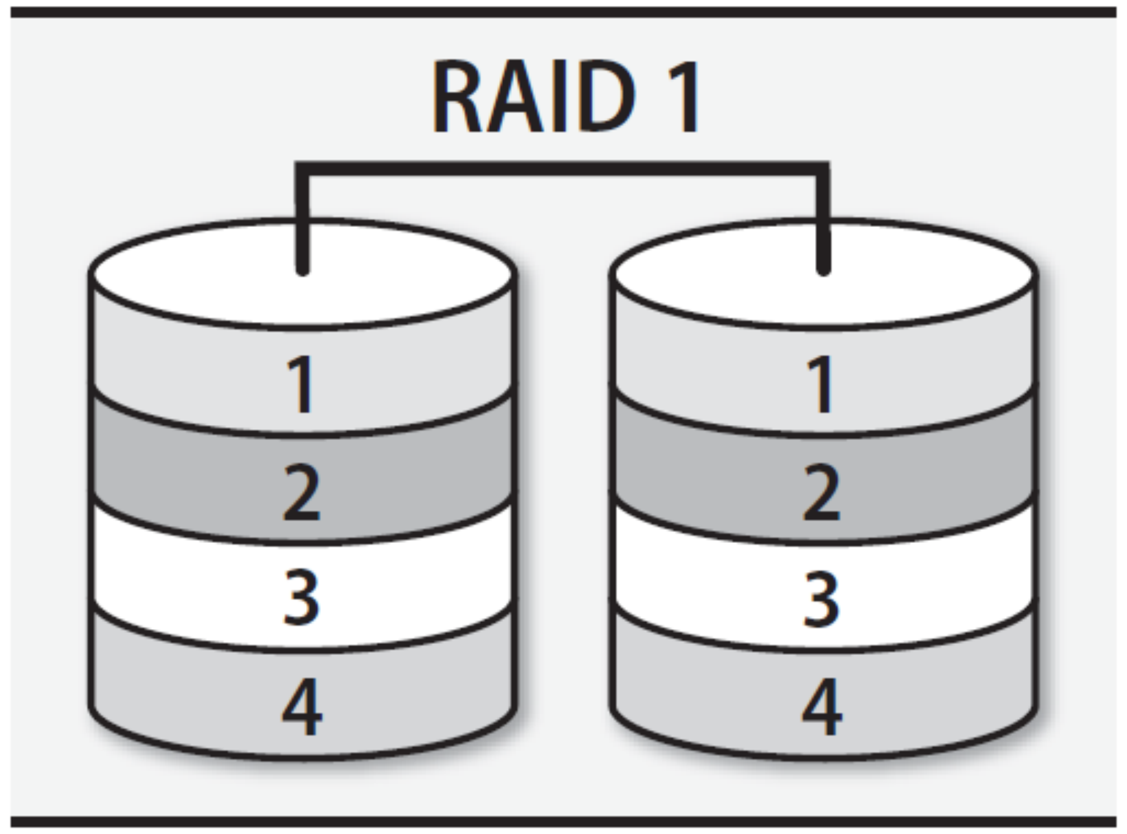
level 6 Full RAID with dual parity

level 10 Striping + Mirroring

RAID levels



RAID 0: striping

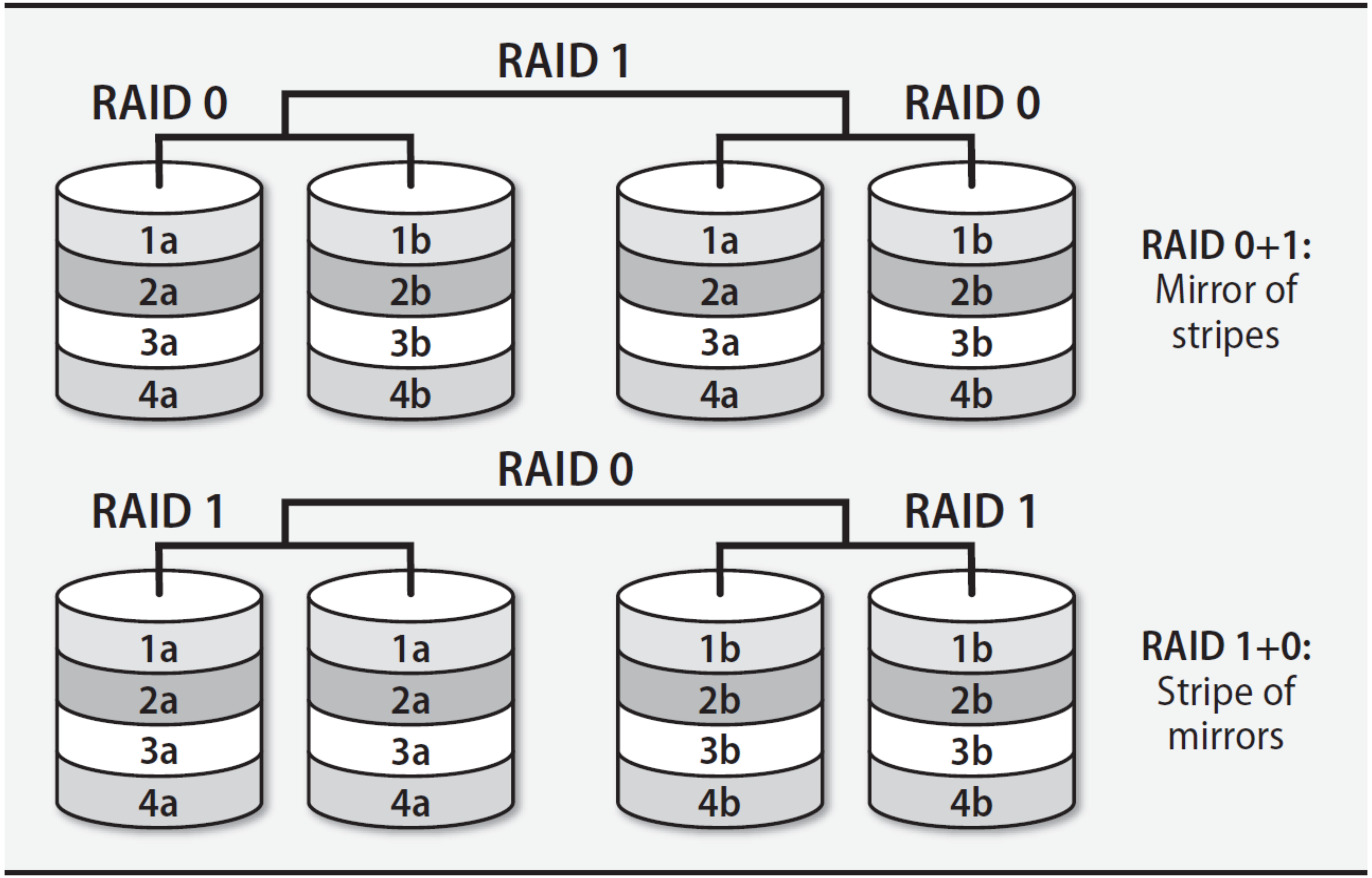


RAID 1: mirroring

Striped vs. Mirrored

- Striped has faster writes
- Mirrored is not any faster than no RAID
  - can be slightly slower (need to send data twice)
- Mirrored is more reliable
- Mirrored gives away 1/2 disk space
  - Striped allows all space to be used

RAID level pairs



RAID 0+1: Mirror of stripes

RAID 1+0: Stripe of mirrors

parity

Higher level RAID depends on parity

- or other more complicated schemes

parity comes in two flavors

- even parity or odd parity

Idea: add an extra bit to check others

- total number of 1's must be even (even parity)
- or odd (odd parity)

parity example

consider data in byte

10100011 ?

add one extra bit for parity check

assuming even parity

need to make total number of ones even

including parity bit

so parity bit should be? 0

more parity

how does this help?

10100011 0

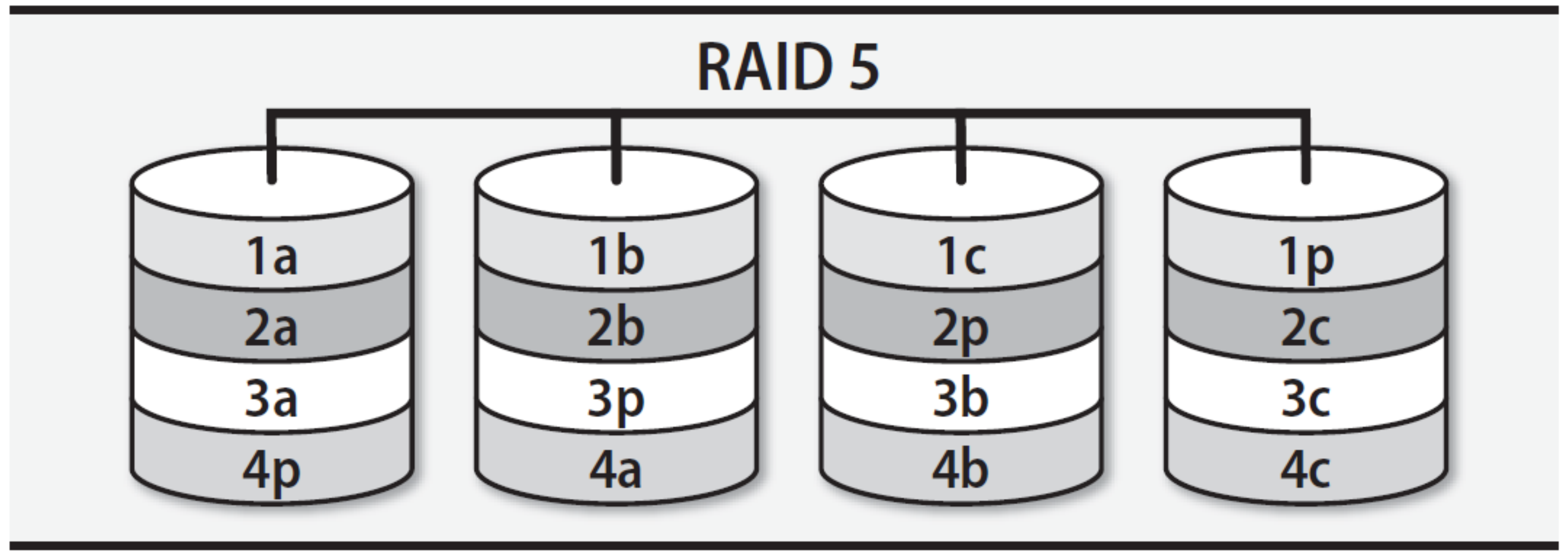
suppose part of data is clobbered

is parity still valid?

no!

simple parity can detect one bit errors

RAID levels 5 and 6 - stripes and parity

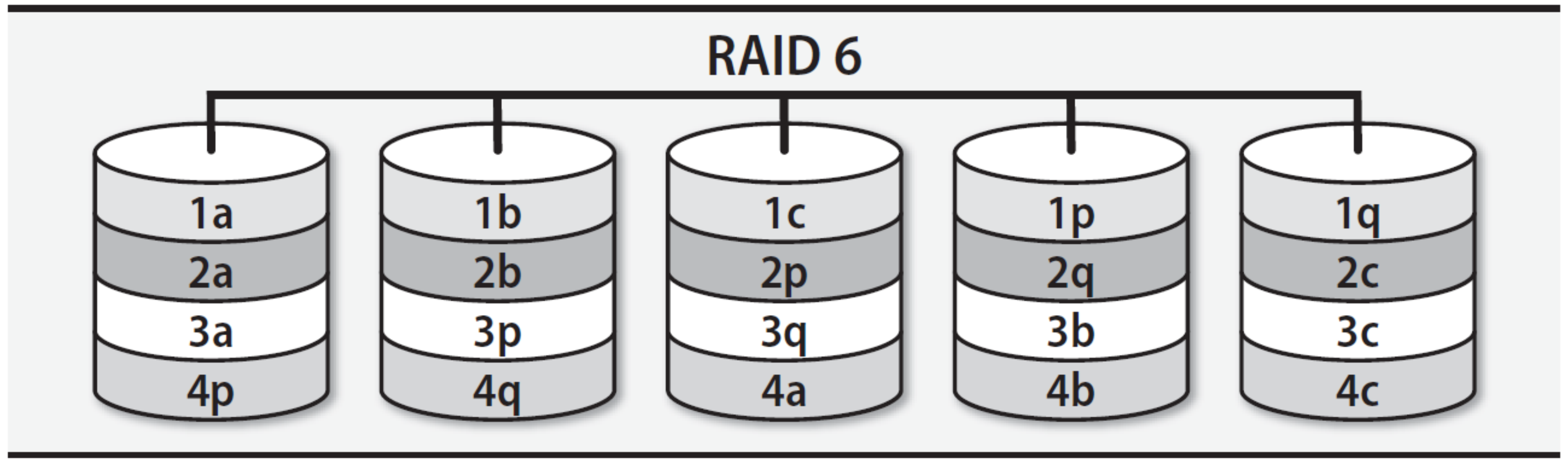


RAID 5

1a	1b	1c	1p
2a	2b	2p	2c
3a	3p	3b	3c
4p	4a	4b	4c

Raid 5: 1 parity block per stripe
1 drive can fail

1	1	0	0
0	1	0	1



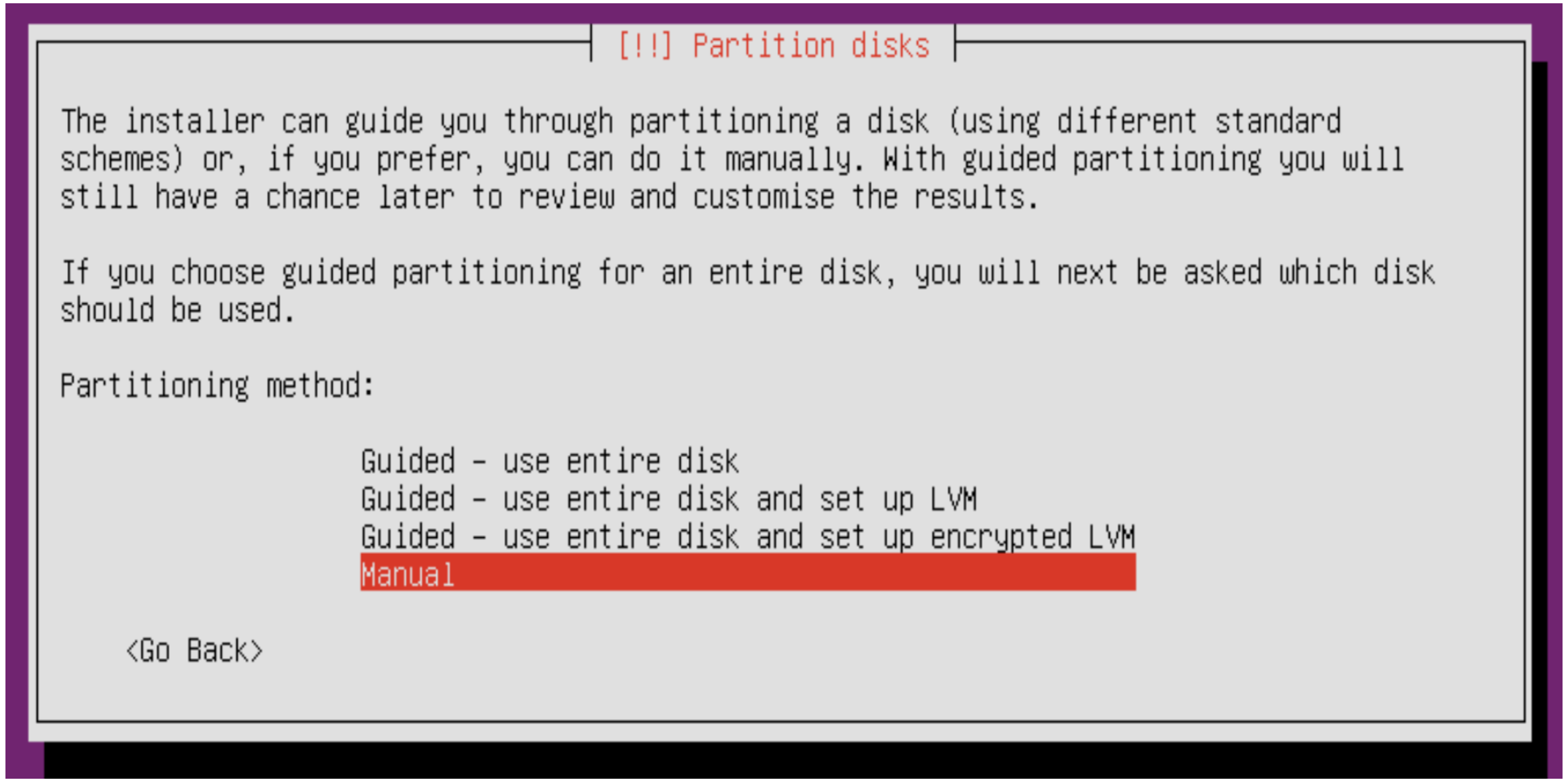
RAID 6

1a	1b	1c	1p	1q
2a	2b	2p	2q	2c
3a	3p	3q	3b	3c
4p	4q	4a	4b	4c

Raid 6: 2 parity blocks per stripe, 2 drives can fail

Ubuntu Server Setup

RAID can be setup from Ubuntu Server install
Go through “Manual” partitioning



```
[!!] Partition disks

The installer can guide you through partitioning a disk (using different standard
schemes) or, if you prefer, you can do it manually. With guided partitioning you will
still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk
should be used.

Partitioning method:

                    Guided - use entire disk
                    Guided - use entire disk and set up LVM
                    Guided - use entire disk and set up encrypted LVM
                    Manual

    <Go Back>
```

but we are going to do it manually…

Command: `mdadm`

3 key create/install steps:

- Define md array (level=0,1)

```
# mdadm --create /dev/md0 --level=mirror \
  --raid-devices=2 /dev/sdb1 /dev/sdc1
```

- Append array details to mdadm.conf:

```
# mdadm -Es >> /etc/mdadm/mdadm.conf
```

- update the boot ramdisk in the kernel:

```
# update-initramfs -u
```

Manually

After md array is created:

- Create the filesystem in the md array

```
mkfs.ext4 /dev/md0
```

- Add it to `/etc/fstab`, if you want to mount after reboot:

```
/dev/md0 /store ext4 defaults 0 2
```

To query RAID arrays

2 ways to query RAID array to verify:

```
# cat /proc/mdstat
# mdadm --query --detail /dev/md*
```

Online cheat sheet of RAID commands:

http://www.ducea.com/2009/03/08/mdadm-cheat-sheet/

A disk fails! Ack!

What happens *when* (not "if") a disk fails?

(Assuming it is configured with mirroring or raid 5,6)

Determining if disk is bad:

```
$ cat /proc/mdstat
```

- For testing can *simulate* a disk failure:

```
$ mdadm /dev/md0 --fail /dev/sdb1
```

- Remove it from the MD

```
mdadm --manage /dev/md0 --remove /dev/sdb1
```

- Shutdown, remove failed drive, add & prep new drive
- Add new drive to the existing MD

```
mdadm --manage /dev/md0 --add /dev/sdc1
```

LVM

LVM

Logical Volume Management

Rather than *physical* storage units, everything is *logical*

A *software layer* goes between the Linux OS and the physical HW

A logical volume becomes a '*flexible*' partition

Examples:

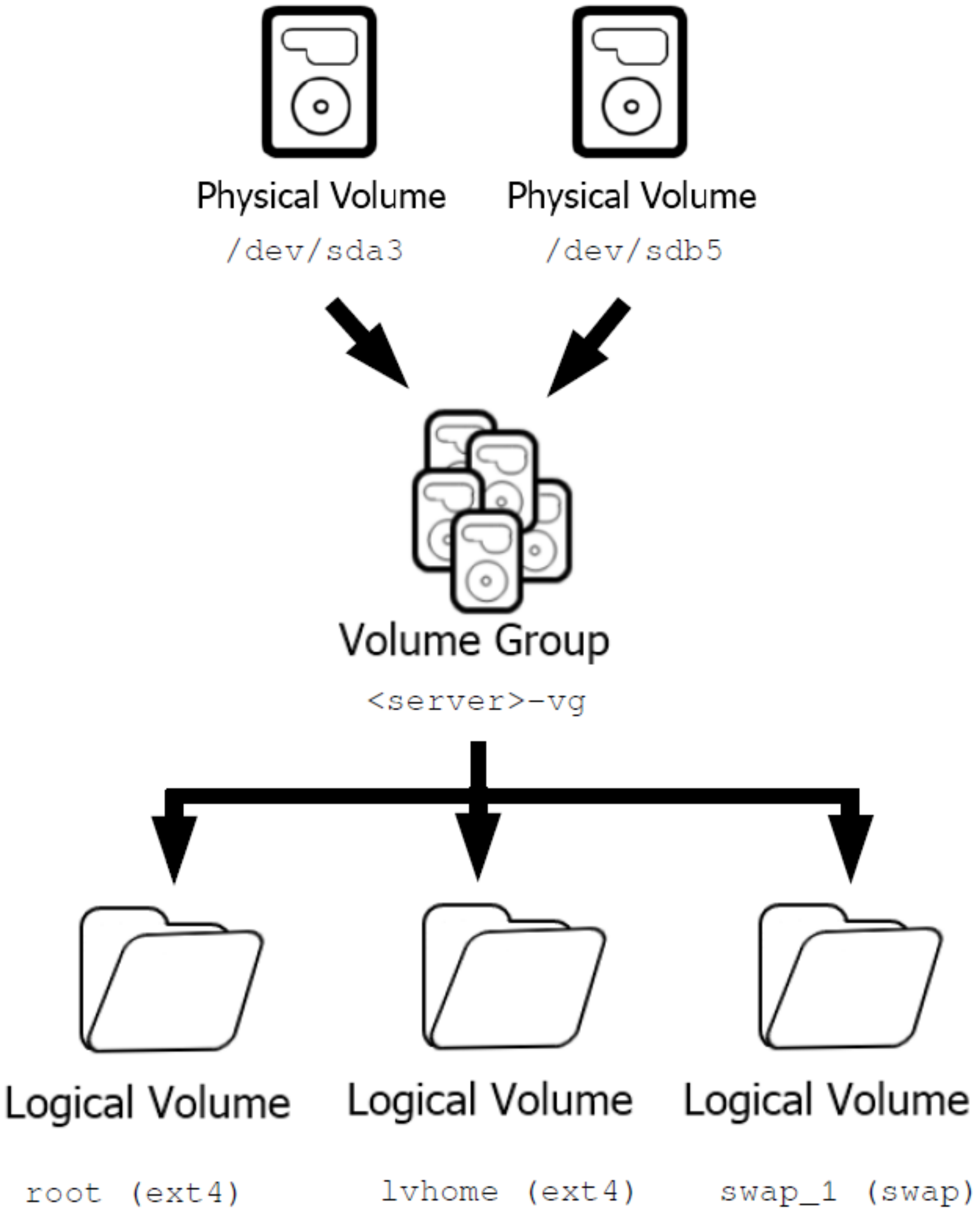
- A volume could be two or three physical storage devices.
- You can add more storage to an existing volume.
- You can resize volumes and re-allocate size very easily.

LVM

Problem: Running out of space on a partition.

1. Buy a new HD and make a new partition. Copy everything over.
2. Make a different partition smaller. Make full partition bigger.

LVM Schematic

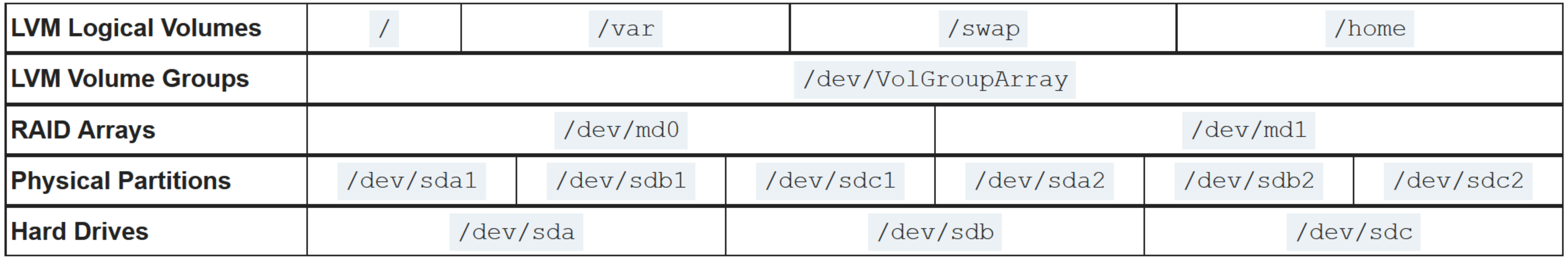


PV physical

VG software

LV OS

Schematic



<table>
<tr><td><b>LVM Logical Volumes</b></td><td><code>/</code></td><td><code>/var</code></td><td colspan="2"><code>/swap</code></td><td colspan="2"><code>/home</code></td></tr>
<tr><td><b>LVM Volume Groups</b></td><td colspan="6"><code>/dev/VolGroupArray</code></td></tr>
<tr><td><b>RAID Arrays</b></td><td colspan="3"><code>/dev/md0</code></td><td colspan="3"><code>/dev/md1</code></td></tr>
<tr><td><b>Physical Partitions</b></td><td><code>/dev/sda1</code></td><td><code>/dev/sdb1</code></td><td><code>/dev/sdc1</code></td><td><code>/dev/sda2</code></td><td><code>/dev/sdb2</code></td><td><code>/dev/sdc2</code></td></tr>
<tr><td><b>Hard Drives</b></td><td colspan="2"><code>/dev/sda</code></td><td colspan="2"><code>/dev/sdb</code></td><td colspan="2"><code>/dev/sdc</code></td></tr>
</table>

LVM Linux Commands

All LVM commands come from one super-command: `$ lvm`

All LVM commands start with: `pv, vg, lv`

```
pvcreate, vgcreate, lvcreate
pvdisplay, ...
```

LVM commands

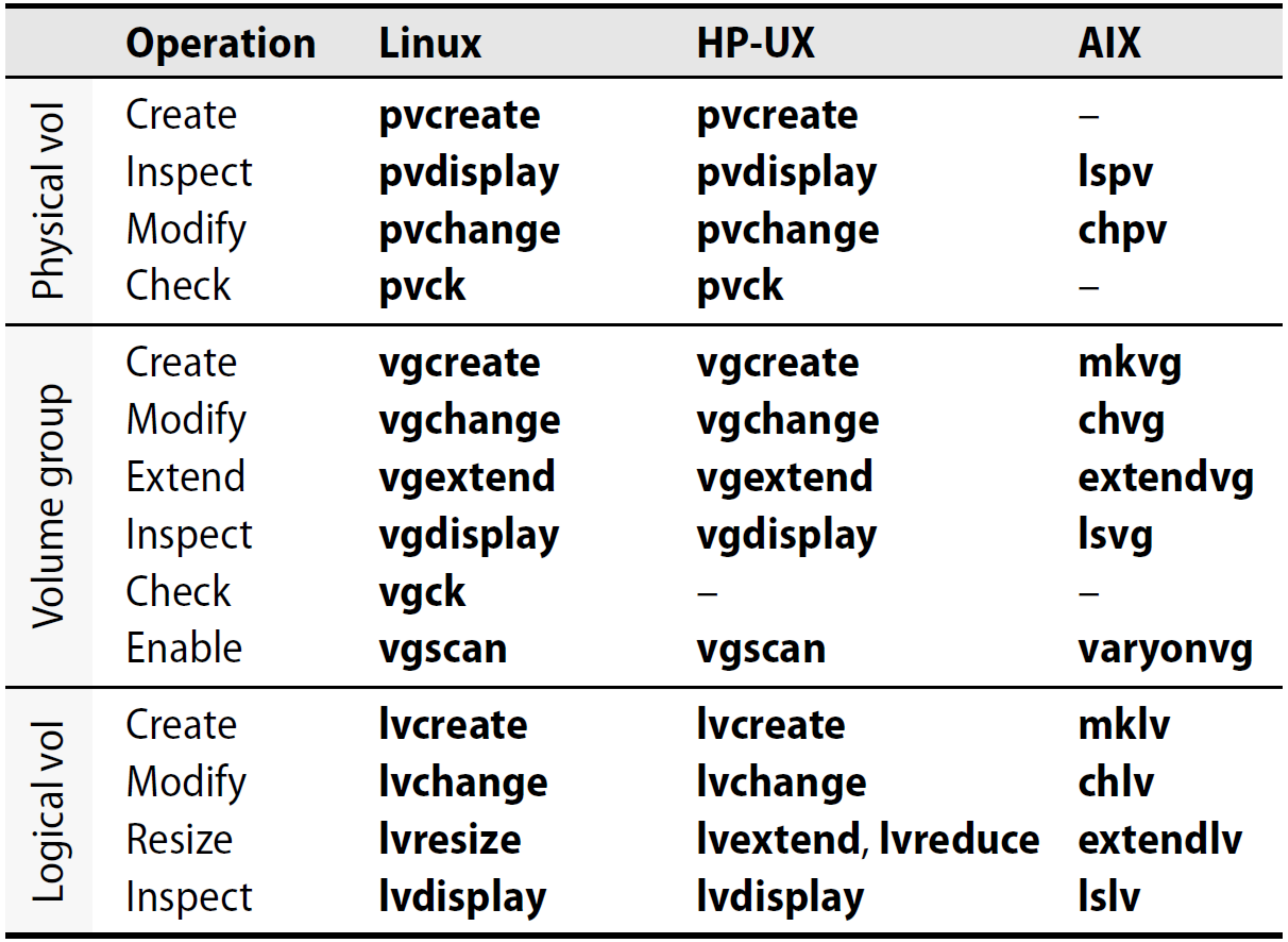


Table 8.4 Comparison of LVM commands

	Operation	Linux	HP-UX	AIX
Physical vol	Create	**pvcreate**	**pvcreate**	–
	Inspect	**pvdisplay**	**pvdisplay**	**lspv**
	Modify	**pvchange**	**pvchange**	**chpv**
	Check	**pvck**	**pvck**	–
Volume group	Create	**vgcreate**	**vgcreate**	**mkvg**
	Modify	**vgchange**	**vgchange**	**chvg**
	Extend	**vgextend**	**vgextend**	**extendvg**
	Inspect	**vgdisplay**	**vgdisplay**	**lsvg**
	Check	**vgck**	–	–
	Enable	**vgscan**	**vgscan**	**varyonvg**
Logical vol	Create	**lvcreate**	**lvcreate**	**mklv**
	Modify	**lvchange**	**lvchange**	**chlv**
	Resize	**lvresize**	**lvextend, lvreduce**	**extendlv**
	Inspect	**lvdisplay**	**lvdisplay**	**lslv**

Manually Setting Up LVM

Create the PV's (physical volumes)

```
# pvcreate /dev/sdb /dev/sdc
# pvdisplay
```

Note: can be entire disk OR partitions

Create VG's (volume groups)

```
# vgcreate vg0 /dev/sdb /dev/sdc
# vgdisplay
```

Create LV's (logical volumes)

```
# lvcreate -n lvhome -L 10Gi vg0
# lvdisplay
# ls /dev/mapper/*
# ls /dev/vg0
```

Manually Setting Up LVM

Add filesystem to LV

```
 # mkfs -t ext4 /dev/mapper/vg0-lvhome
```

OR

```
 # mkfs -t ext4 /dev/vg0/lvhome
```

mount

```
 # mount /dev/mapper/vg0-lvhome /mnt/h1
```

Add to fstab if you want

All done? Want to remove lvm & raid?

Reverse order

```
# lvdisplay
# lvremove <lv>

# vgdisplay
# vgremove <vg>

# pvdisplay
# pvremove <pv>

# cat /proc/mdstat
# mdadm -stop     /dev/<md>
# mdadm --remove /dev/<md>
```