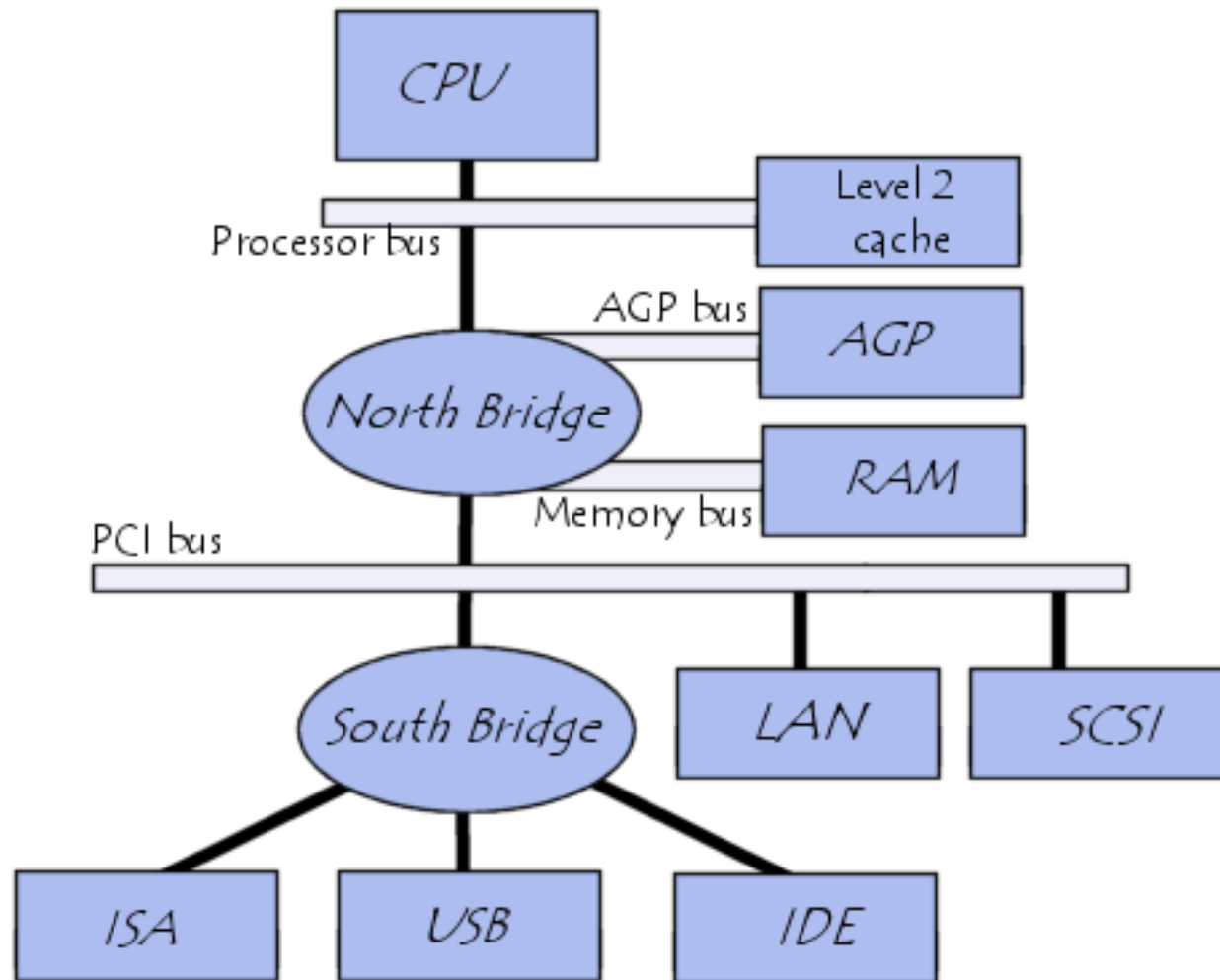




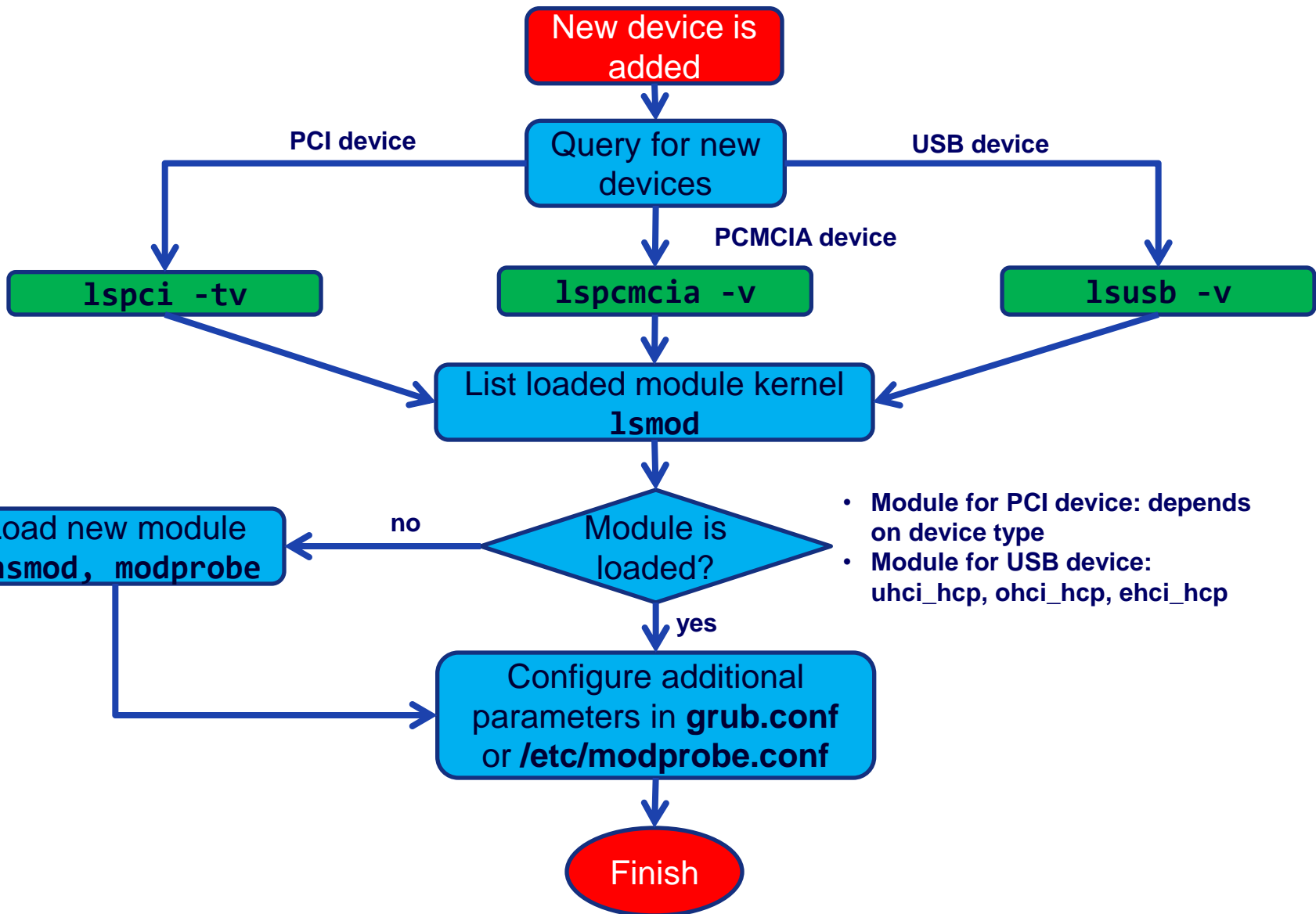
LPIC-2 TRAINING COURSE

Topic 204: Hardware

Computer Buses

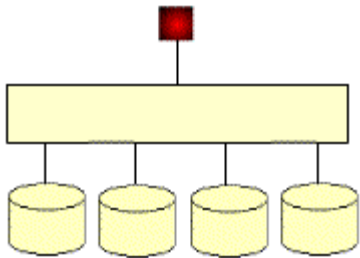


Adding new Hardware



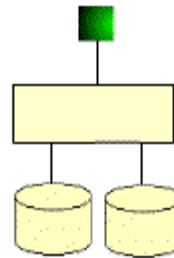
Introduction to RAID

- ❖ RAID = Redundant Arrays of Independent Disks
- ❖ RAID Levels: 0, 1, 2, 3, 4, 5, 6, 10, 01
- ❖ Software RAID vs. Hardware RAID
- ❖ Spare Disks (Hot Spare): used in case another disk in the RAID array fails



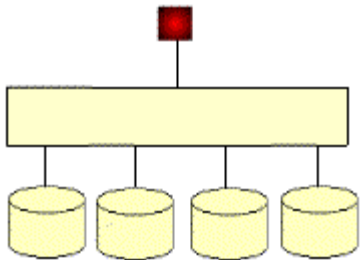
RAID 0:

- Data Striping
- Best Performance
- Use 100% capacity
- No redundancy
- Require any number of disks



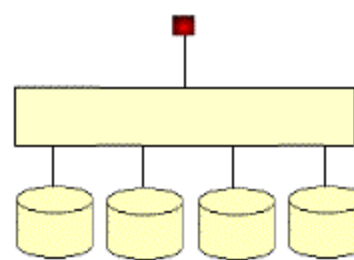
RAID 1:

- Data Mirroring
- Read advantage
- Lost 50% capacity
- Fully redundancy
- Require 2 disks



RAID 4:

- Striping + Dedicated Parity
- Good Performance
- Use (N-1) capacity
- Tolerate 1 disk lost
- Require 3 disks or more



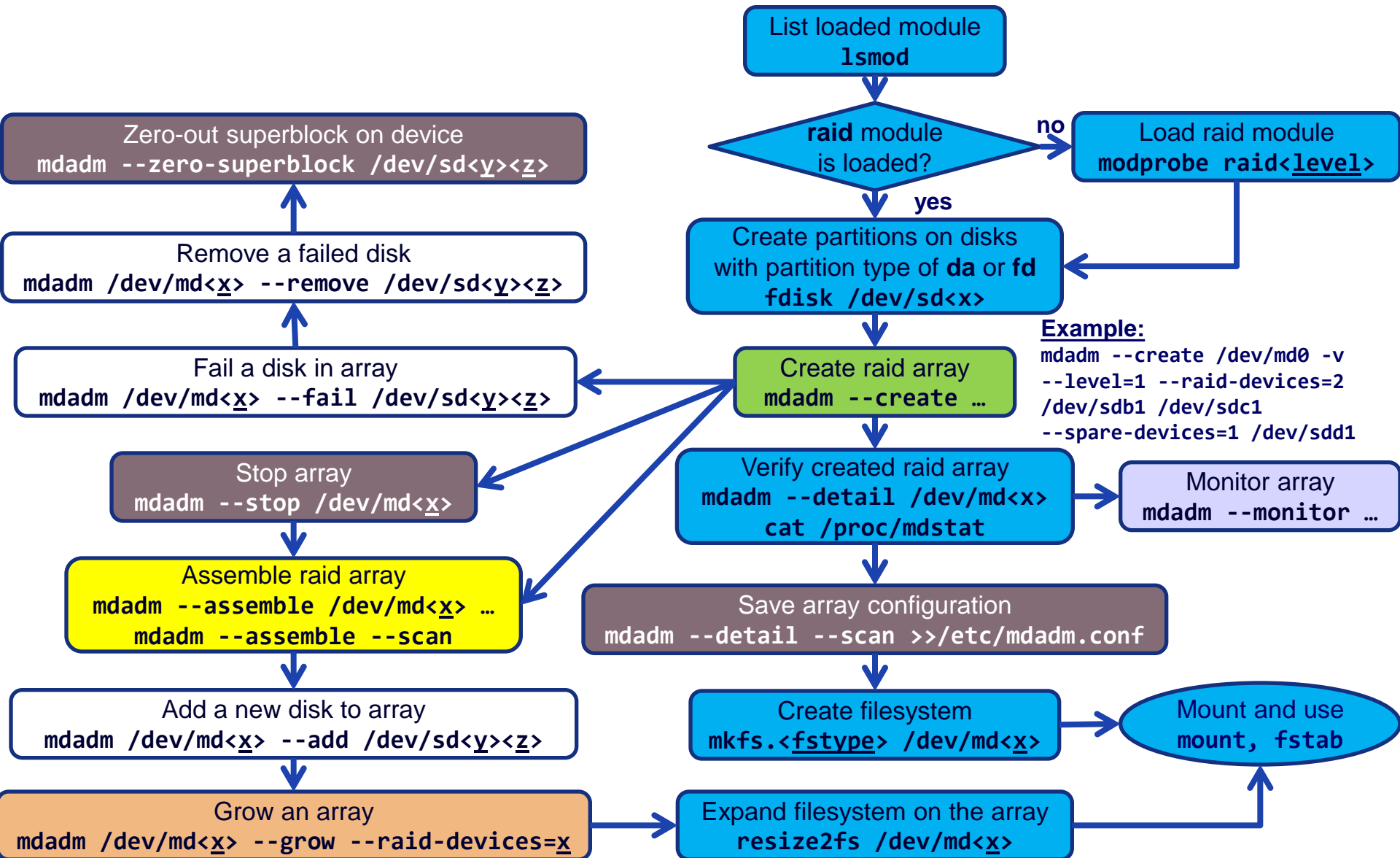
RAID 5:

- Striping + Distributed Parity
- Good performance
- Use (N-1) capacity
- Tolerate 1 disk lost
- Require 3 disks or more

Linux Software RAID (MD Driver)

- ❖ Supported level: LINEAR, 0,1, 4, 5, 6,1+0, MULTIPATH, FAULTY
 - Spare disks are not supported with LINEAR, RAID-0 and MULTIPATH
 - Spare disks can be “*global*”
- ❖ RAID Management tool: **mdadm**
 - Configuration file: ***/etc/mdadm.conf***
 - **mdadm** modes:
 - **-C, --create** *create a new RAID array with per-device superblock*
 - **-A, --assemble** *assemble the parts of previously created array into an active array*
 - **-F, --monitor** *monitor one or more md devices and act on any state changes*
 - **-B, --build** *build a legacy array that doesn't have per-device superblock*
 - **-G, --grow** *grow or shrink an array*
 - **--add, --fail, --remove** *manage specific components of an array*
 - **Misc** *everything else mode*

Working with Linux MD Driver



Exercise 1: Working with Linux MD Driver

In this exercise, you're going to create a RAID1 array with 01 hotspare drive

1. Add 3 new SCSI harddisks to your Linux Virtual Machine, each is 2GB in size
 - Hint: your new disks will be assumed as **sdd**, **sde** and **sdf** throughout these hints
2. Verify that **raid1** module is loaded into the kernel. If not, load this module with **modprobe**
 - Hint: **lsmod | grep raid1**
 - Hint: **modprobe raid1**
3. Partition your new disks with **fdisk**, set the partition type to **fd**
 - Hint: After creating partitions on your disks, press **t** and enter **fd** as the partition type
4. Create a RAID1 array with 01 hotspare using these 3 new partitions
 - Hint: **mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/sdd1 /dev/sde1 --spare-devices=1 /dev/sdf1**
5. Verify the newly created array
 - Hint: **mdadm --detail /dev/md0; mdadm --examine /dev/sd[def]1 ; cat /proc/mdstat**
6. Reboot your machine. Verify that your array does not automatically come up after reboot
 - Hint: **mdadm --detail /dev/md0; mdadm --detail --scan**
7. Re-assemble your array and verify
 - Hint: **mdadm --assemble /dev/md0 /dev/sd[def]1; mdadm --detail /dev/md0**
8. Simulate a disk fault and verify that the hotspare replaced the fail disk
 - Hint: **mdadm /dev/md0 --fail /dev/sde1; mdadm --detail /dev/md0**
9. Remove the faulty disk from array
 - Hint: **mdadm /dev/md0 --remove /dev/sde1; mdadm --detail /dev/md0**
10. Add the disk back to the array
 - Hint: **mdadm /dev/md0 --add /dev/sde1 ; mdadm --detail /dev/md0**

Exercise 1: Working with Linux MD Driver (cont')

11. Save the array configuration to */etc/mdadm.conf*
 - Hint: `mdadm --detail --scan >> /etc/mdadm.conf`
12. Stop the array
 - Hint: `mdadm --stop /dev/md0; mdadm --detail /dev/md0`
13. Re-assemble the array
 - Hint: `mdadm --assemble /dev/md0; mdadm --detail /dev/md0`
14. Reboot your machine. Verify that your array automatically come up after reboot
 - Hint: `mdadm --detail /dev/md0`
15. Format the array with ext3 filesystem
 - Hint: `mkfs.ext3 /dev/md0`
16. Mount the filesystem
 - Hint: `mount /dev/md0 /mnt`
17. Monitoring the RAID array
 - Hint: `mdadm --monitor --scan --daemonise > /var/run/mdadm/monitor`

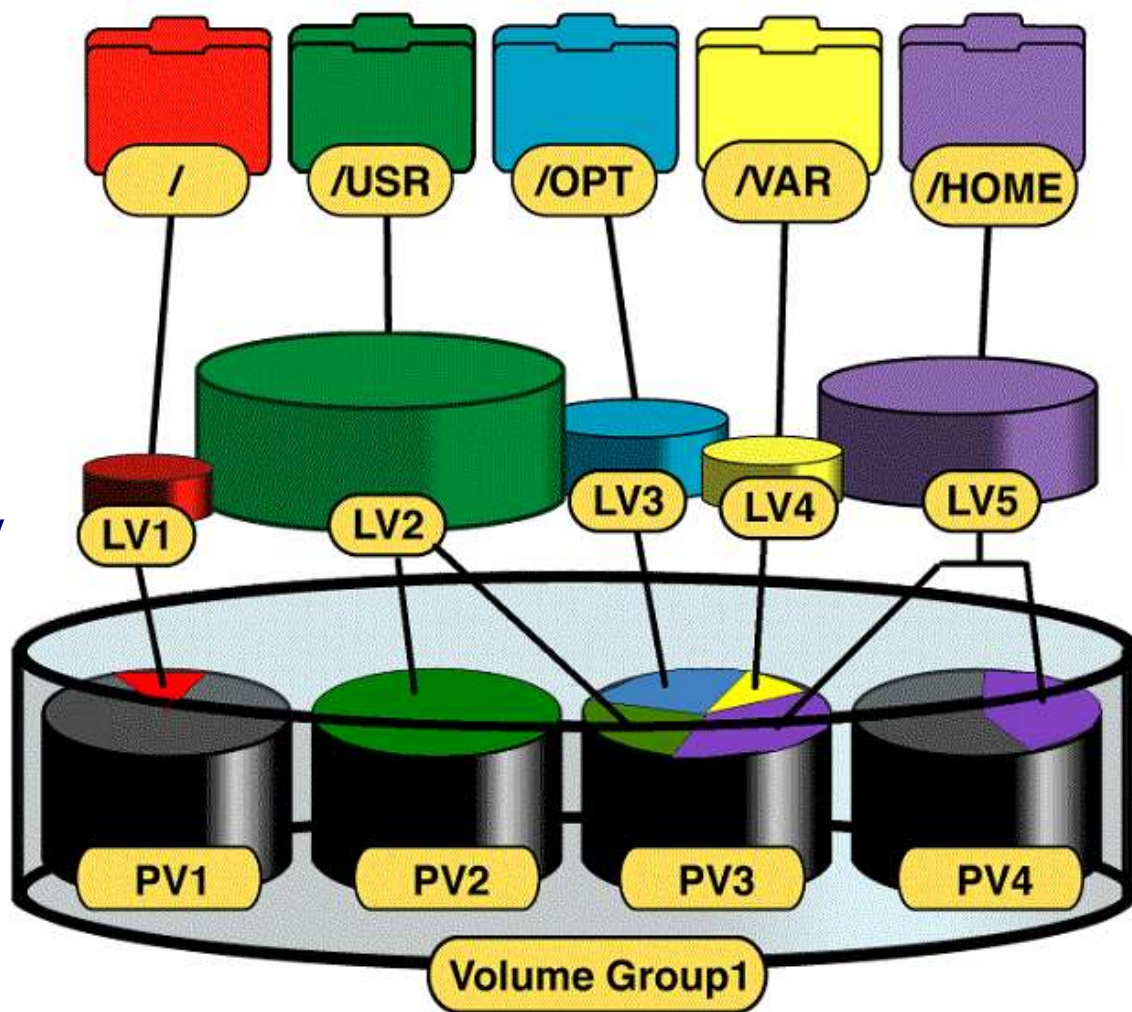
Introduction to LVM

❖ LVM = Logical Volume Manager

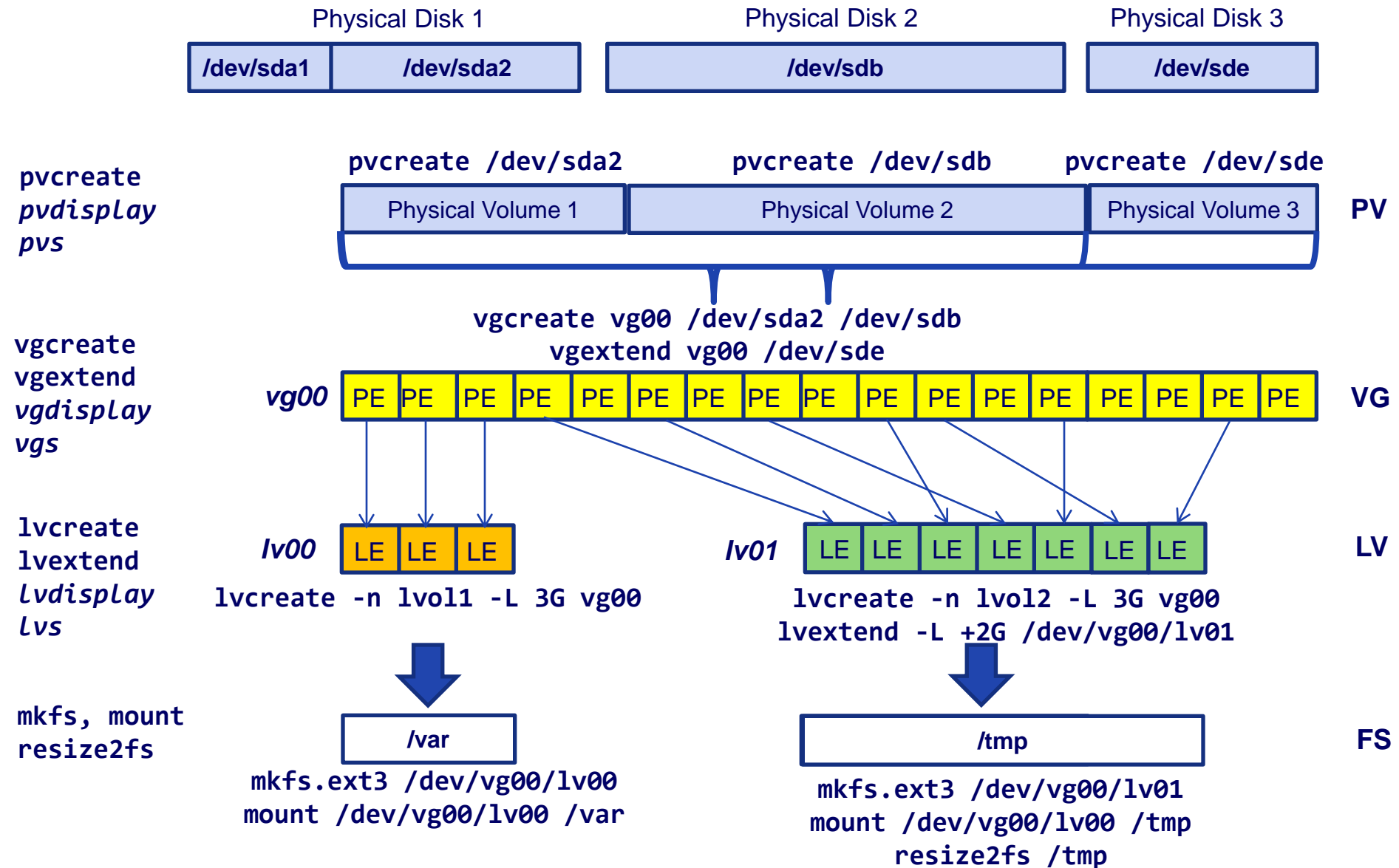
- **PV** = Physical Volume
 - Physical Disk/Disk Partition
- **VG** = Volume Group
 - Comprised of PV
- **LV** = Logical Volume
 - Virtual partition
 - Can be formatted and mount

❖ Benefit of LVM:

- Online resizing of VG
- Online resizing of LV
- Online moving LV between PV
- Create snapshots
- Stripe whole or parts of LV across multiple PV (similar to RAID 0)
- Mirror whole or parts of LV (similar to RAID 1)
- Split or merge VGs

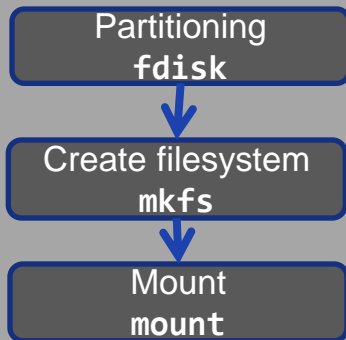


Working with Linux LVM



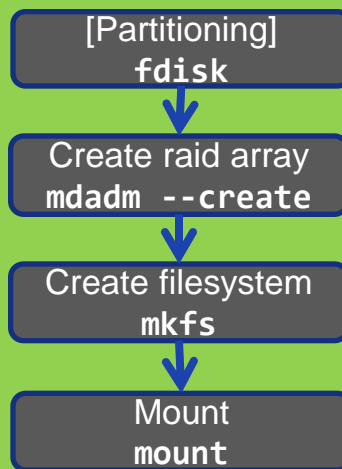
Physical Disk vs. RAID vs. LVM

Physical Disk



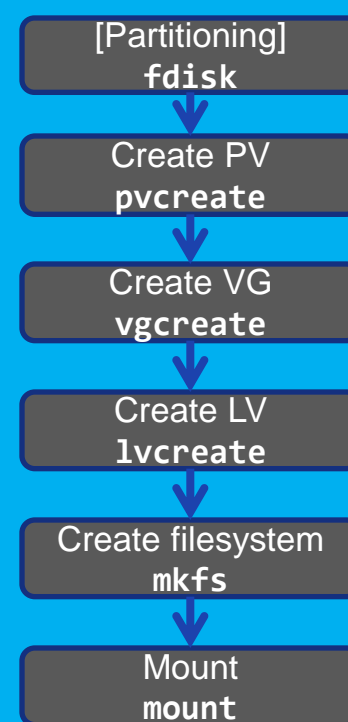
Simplicity

RAID



**Availability
&
Performance**

LVM



Managability

Exercise 2: Working with LVM

In this exercise, you're going to create a LVM Volume with 3 harddisks

1. Add 3 new SCSI harddisks to your Linux Virtual Machine
 - Hint: You can choose any size for your disks. Your new disks will be assumed as **sdg**, **sdh** and **sdi** throughout these hints
2. Partition the first disk (with only one partition), set the partition type to **8e** (Linux LVM).
 - Hint: `fdisk /dev/sdg`
3. [Optional] Erase the partition table of the second disk by zeroing its first sector
 - Hint: `dd if=/dev/zero of=/dev/sdh bs=512 count=1`
4. Initialize the first partition on the first disk (**sdg1**) and the second disk (**sdh**) for use by LVM
 - Hint: `pvcreate /dev/sdg1; pvcreate /dev/sdh; pvdisplay /dev/sdg1; pvs`
5. Create a new Volume Group named **vg00** with these two Physical Volume and verify it.
 - Hint: `vgcreate -v vg00 /dev/sdg1 /dev/sdh; vgdisplay -v /dev/vg00; vgs`
6. Create a new Logical Volume named **lv00** with the size of 1GB on **vg00** and verify it.
 - Hint: `lvcreate -n lv00 -L 1G vg00; lvdisplay -v /dev/vg00/lv00; lvs`
7. Format **lv00** with **ext3** filesystem and mount it to **/mnt**
 - Hint: `mkfs.ext3 /dev/vg00/lv00; mkdir /mnt; mount /dev/vg00/lv00 /mnt`
8. Create some data on **/mnt** (by copying files from somewhere) and verify its size
 - Hint: `cp /var/log/* /mnt; df -h /mnt`

Exercise 2: Working with LVM (cont')

9. Online extend **lv00** to 1.5GB and verify
 - Hint: `lvextend -L +512M /dev/vg00/lv00; lvs /dev/vg00`
10. Verify that filesystem size on **lv00** is still 1GB (you can try umount/remount this filesystem)
 - Hint: `df -h /mnt`
11. Online extend the filesystem on **lv00** and verify **/mnt** is now 1.5GB
 - Hint: `resize2fs /dev/vg00/lv00; df -h /mnt`
12. Extend **vg00** with the third disk and verify **vg00**
 - Hint: `fdisk /dev/sdi; pvcreate /dev/sdi1; vgextend vg00 /dev/sdi1; vgdisplay`
13. Create the snapshot of **lv00** with the size of 512MB
 - Hint: `lvcreate -L 512M -s -n lv00-snap vg00; lvs`
14. Mount the snapshot volume on **/snap** and verify its content
 - Hint: `mkdir /snap; mount /dev/vg00/lv00-snap /snap; ls -l /snap`
15. Unmount and remove the snapshot
 - Hint: `umount /snap; lvremove /dev/vg00/lv00-snap`

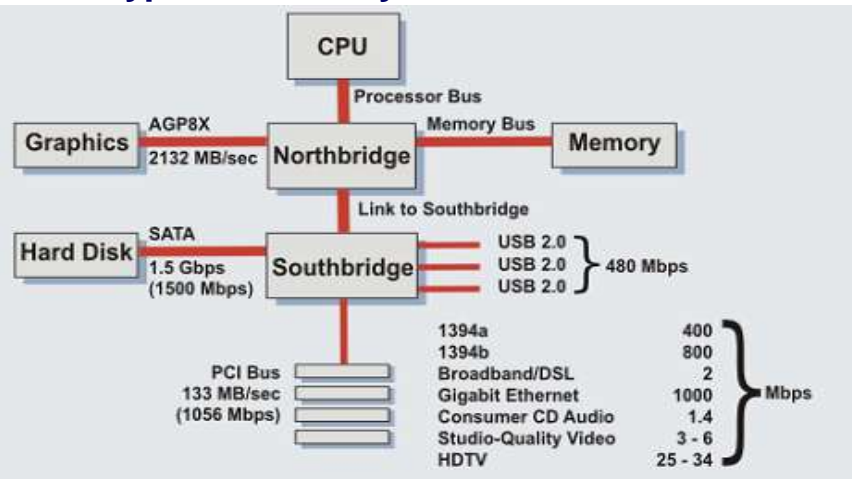


Thank You !

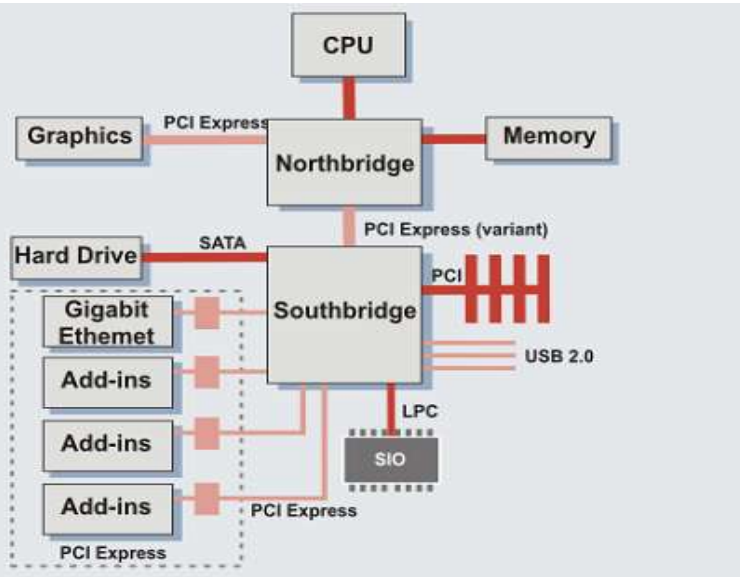


BACKUP SLIDES

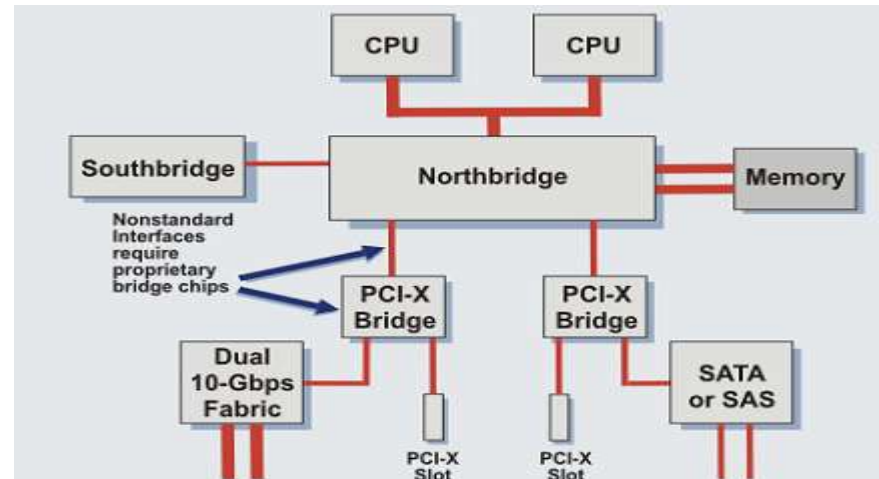
Typical Client System Architecture



Sample Desktop Architecture



Current Dual-Processor Server Architecture



Sample Server Architecture

