MHPC
Master in High Performance Computing

LVM in a nutshell

Moreno Baricevic

SISSA
ma per seguir virtute e canoscenza
Scuola Internazionale Superiore
di Studi Avanzati

ICTP
The Abdus Salam
International Centre
for Theoretical Physics

MHPC
Master in High Performance Computing

What are we talking about?

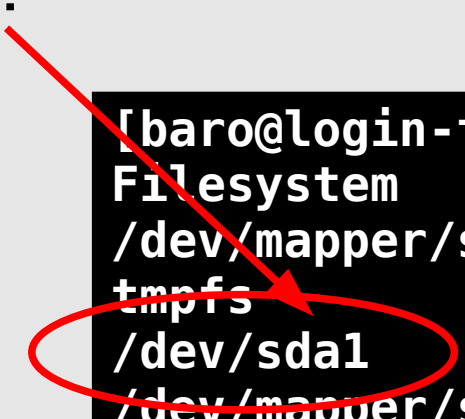
```
[baro@login-tmp ~]$ df -h
Filesystem                Size  Used Avail Use% Mounted on
/dev/mapper/sysVG-LV00     20G  2.9G   16G  16% /
tmpfs                     5.9G  151M  5.7G   3% /dev/shm
/dev/sda1                 194M   87M   98M  48% /boot
/dev/mapper/sysVG-LV02     49G  182M   46G   1% /tmp
/dev/mapper/sysVG-LV01     49G  491M   46G   2% /var
10.1.0.1:/u/shared        247G   20G  215G   9% /u/shared
10.1.1.2:/home             43T  144G   43T   1% /home
10.1.1.2:/scratch         256T  4.2T  250T   2% /scratch
```

SISSA
ICTP

MHPC
Master in High Performance Computing

What are we talking about?

???

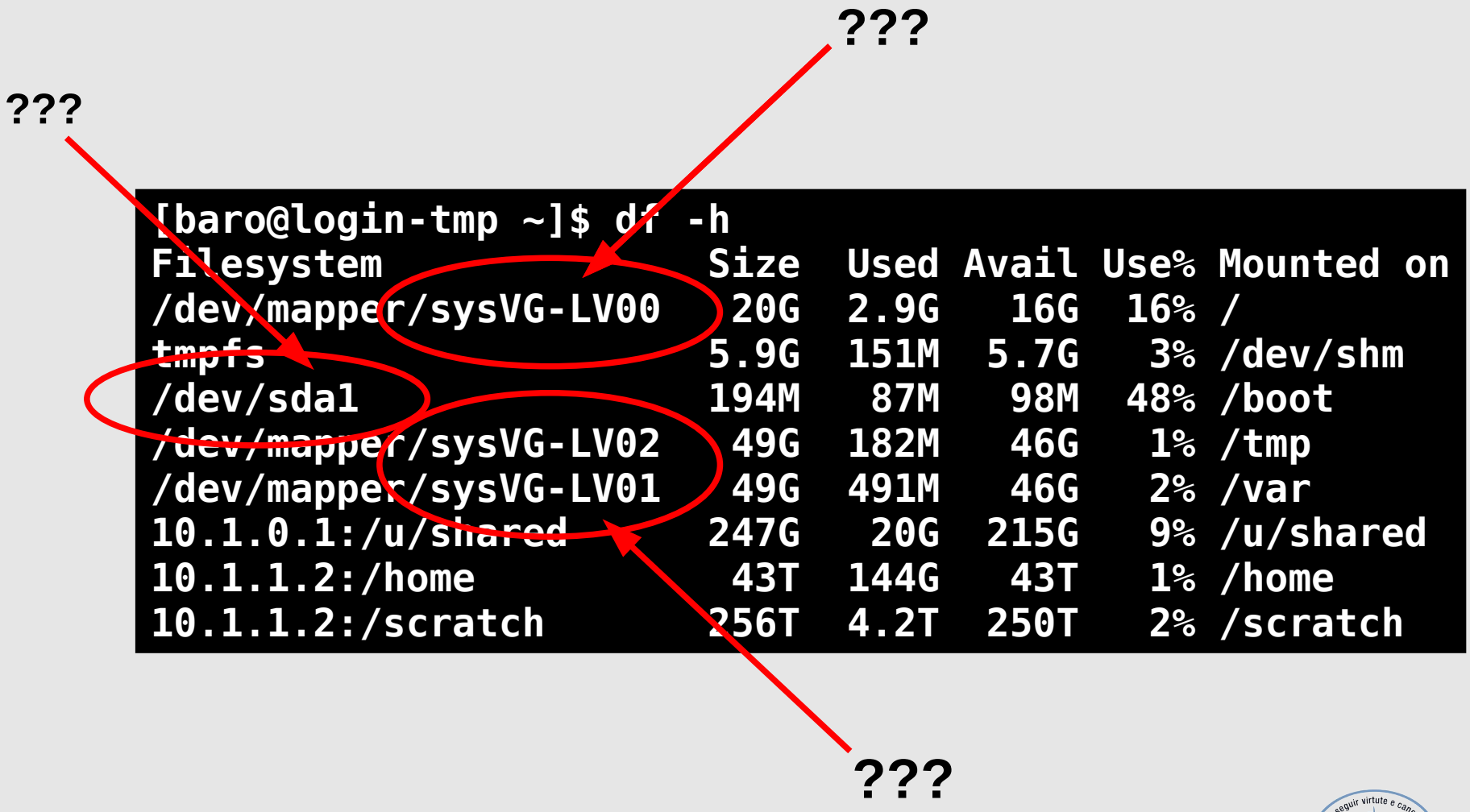


```
[baro@login-tmp ~]$ df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/sysVG-LV00
                       20G  2.9G   16G  16% /
tmpfs                 5.9G  151M  5.7G   3% /dev/shm
/dev/sda1             194M   87M   98M  48% /boot
/dev/mapper/sysVG-LV02
                       49G  182M   46G   1% /tmp
/dev/mapper/sysVG-LV01
                       49G  491M   46G   2% /var
10.1.0.1:/u/shared    247G   20G  215G   9% /u/shared
10.1.1.2:/home         43T  144G   43T   1% /home
10.1.1.2:/scratch     256T  4.2T  250T   2% /scratch
```

SISSA
ICTP

MHPC
Master in High Performance Computing

What are we talking about?



ma per seguir virtute e canoscenza
SISSA
ICTP

Background

- a hard disk can be seen as a continuous row of logical blocks

MHPC
Master in High Performance Computing

Background

- a hard disk can be seen as a continuous row of logical blocks
- in order to store data on a disk this row needs to be cut in sections called partitions

SISSA
ma per seguir virtute e canoscenza
ICTP

Background

- a hard disk can be seen as a continuous row of logical blocks
- in order to store data on a disk this row needs to be cut in sections called partitions
- it can be
  - one huge partition covering the whole disk
  - several small partitions on one disk
  - a combination over several disks

MHPC
Master in High Performance Computing

Background

- a hard disk can be seen as a continuous row of logical blocks
- in order to store data on a disk this row needs to be cut in sections called partitions
- it can be
  - one huge partition covering the whole disk
  - several small partitions on one disk
  - a combination over several disks
- a partition must be a continuous chunk of blocks (here lies part of the problem)

SISSA
ICTP

MHPC
Master in High Performance Computing

Background

- a hard disk can be seen as a continuous row of logical blocks
- in order to store data on a disk this row needs to be cut in sections called partitions
- it can be
  - one huge partition covering the whole disk
  - several small partitions on one disk
  - a combination over several disks
- a partition must be a continuous chunk of blocks (here lies part of the problem)
- a partition is forever (ok, not really...)

SISSA
ICTP

MHPC
Master in High Performance Computing

What is LVM?

Logical Volume Manager

- a logical layer placed between disk partitions and file systems
- breaks filesystem / physical partition binding
- provides logical partitions called volumes

SISSA
ma per seguir virtute e canoscenza
ICTP

MHPC
Master in High Performance Computing

What is LVM?

Logical Volume Manager

- a logical layer placed between disk partitions and file systems
- breaks filesystem / physical partition binding
- provides logical partitions called volumes
- greater flexibility (live create / remove / resize)
- allows disks aggregation (MD compatible)
- live snapshots (*copy-on-write*) and cloning (mirror)

SISSA
ICTP

MHPC
Master in High Performance Computing

What is LVM?

Logical Volume Manager

- a logical layer placed between disk partitions and file systems
- breaks filesystem / physical partition binding
- provides logical partitions called volumes
- greater flexibility (live create / remove / resize)
- allows disks aggregation (MD compatible)
- live snapshots (*copy-on-write*) and cloning (mirror)

Cons? Additional layers of complexity.

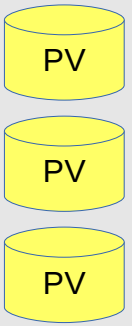
- disaster recovery becomes more difficult
- another abstraction layer in I/O operations
- advanced skills required

SISSA
ICTP

New terms

PV – Physical Volume

collects one or more disk partitions or whole disks (/dev/sda, /dev/sdc3, /dev/loop0, ...)



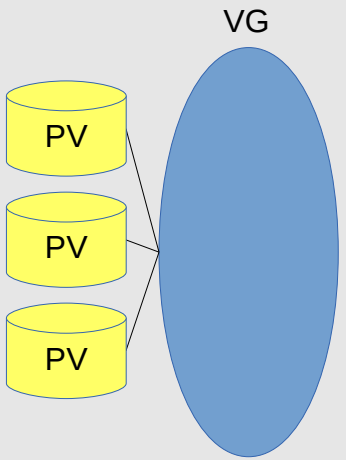
New terms

PV – Physical Volume

collects one or more disk partitions or whole disks (/dev/sda, /dev/sdc3, /dev/loop0, ...)

VG – Volume Group

creates one big virtual disk out of one or more PVs (vg-sys, vg-data)



New terms

PV – Physical Volume

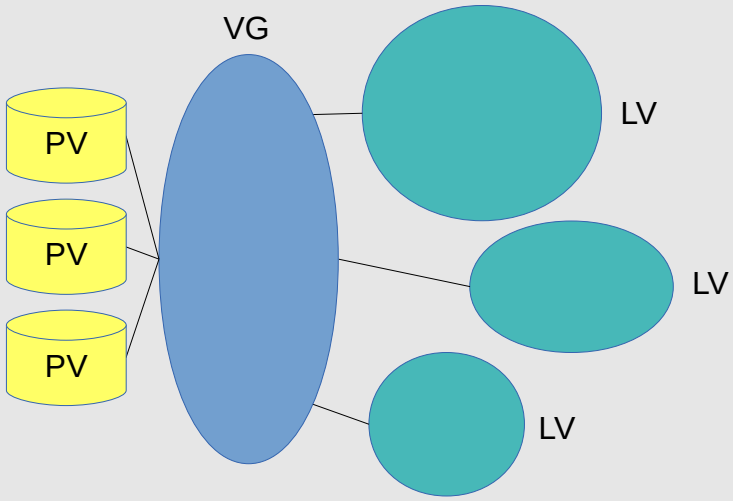
collects one or more disk partitions or whole disks (/dev/sda, /dev/sdc3, /dev/loop0, ...)

VG – Volume Group

creates one big virtual disk out of one or more PVs (vg-sys, vg-data)

LV – Logical Volume

the VG can be split up into several LVs and each of them can host a different filesystem (as for physical partitions) (lv-root, lv-home)



New terms

PV – Physical Volume

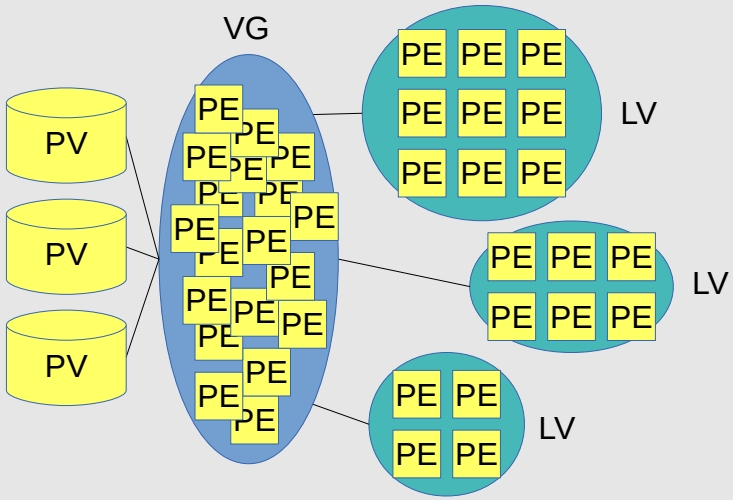
collects one or more disk partitions or whole disks (/dev/sda, /dev/sdc3, /dev/loop0, ...)

VG – Volume Group

creates one big virtual disk out of one or more PVs (vg-sys, vg-data)

LV – Logical Volume

the VG can be split up into several LVs and each of them can host a different filesystem (as for physical partitions) (lv-root, lv-home)



PE – Physical Extent

smallest allocatable chunk for LVs in a VG (default 4MiB, min 1KiB)

New terms

PV – Physical Volume

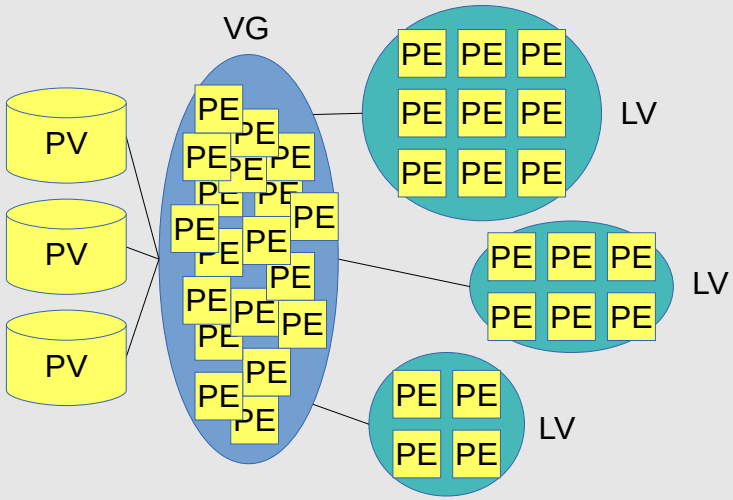
collects one or more disk partitions or whole disks (/dev/sda, /dev/sdc3, /dev/loop0, ...)

VG – Volume Group

creates one big virtual disk out of one or more PVs (vg-sys, vg-data)

LV – Logical Volume

the VG can be split up into several LVs and each of them can host a different filesystem (as for physical partitions) (lv-root, lv-home)



PE – Physical Extent

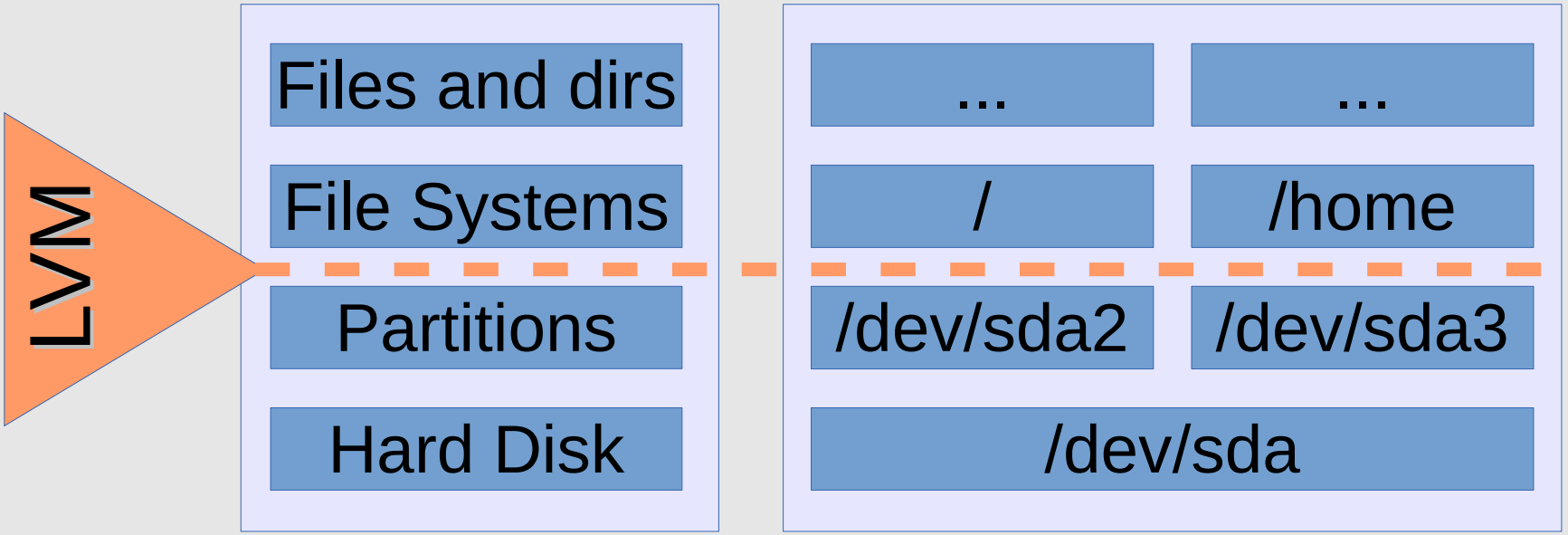
smallest allocatable chunk for LVs in a VG (default 4MiB, min 1KiB)

/dev/vg-sys/lv-root == /dev/mapper/vg-sys-lv-root

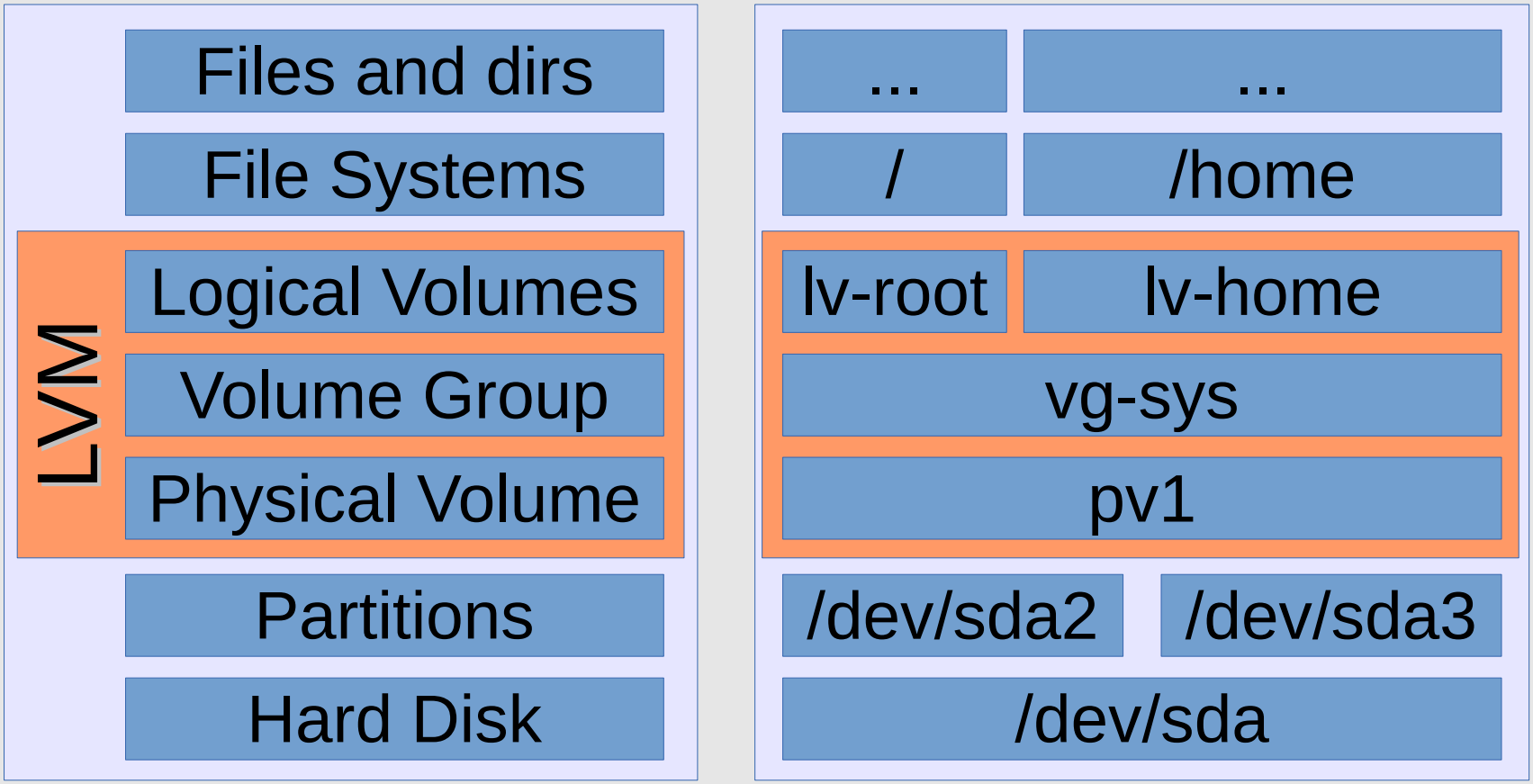
/dev/vg-sys/lv-home == /dev/mapper/vg-sys-lv-home

Standard layout

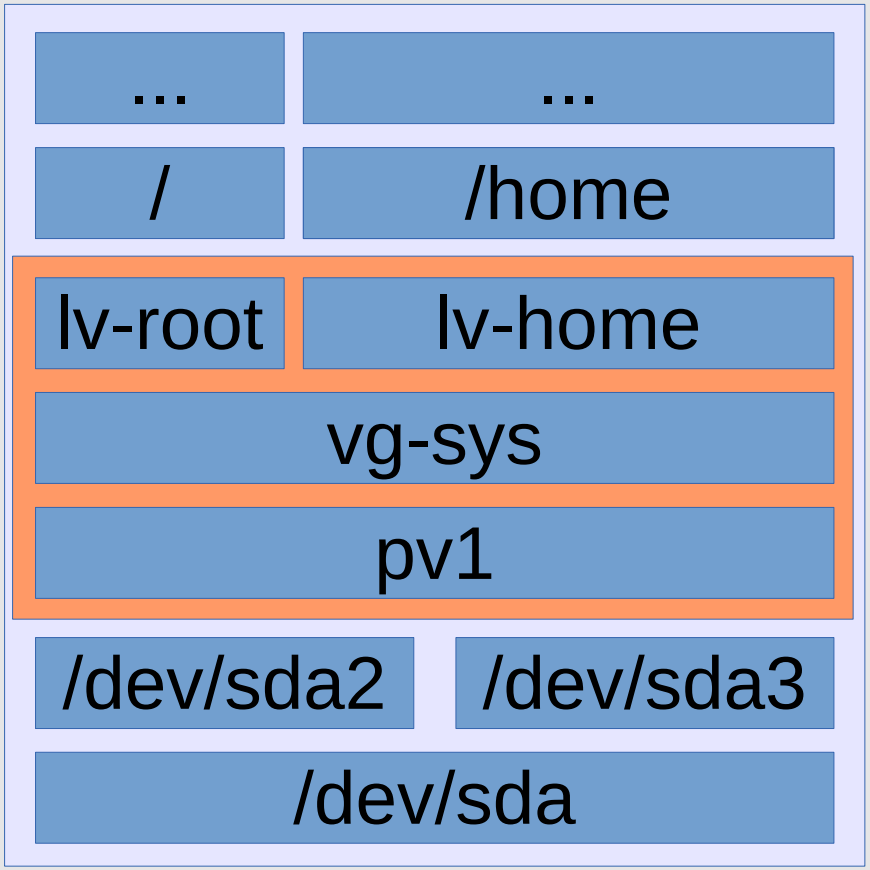
Standard layout



LVM layout

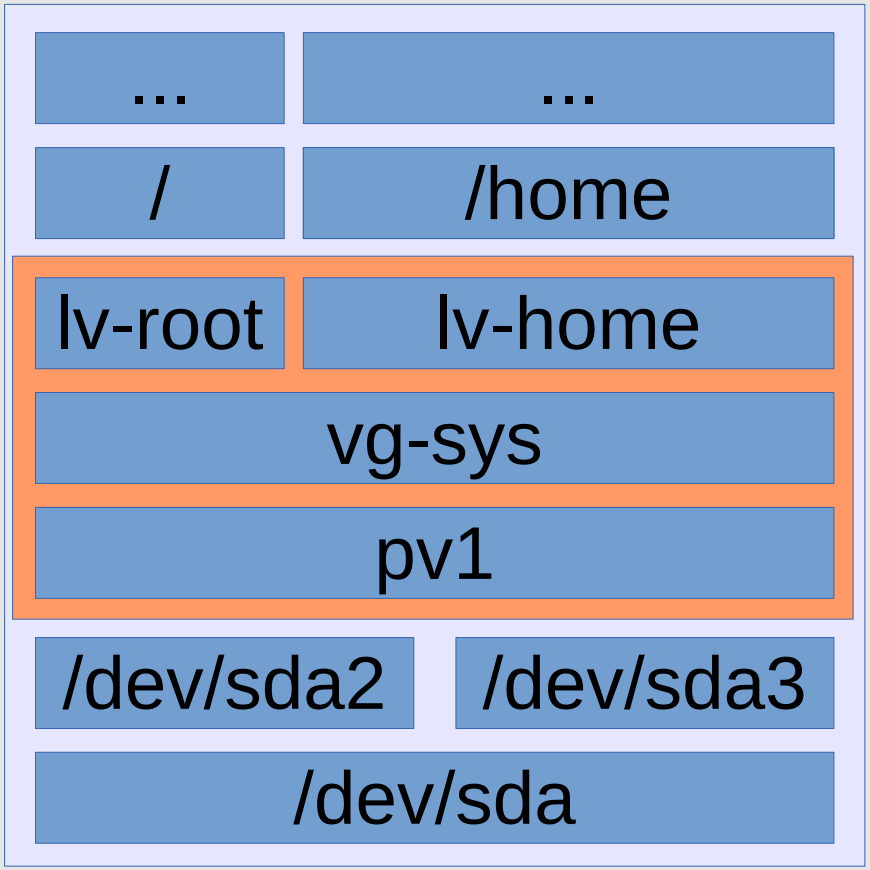

Files and dirs
File Systems
LVM
Logical Volumes
Volume Group
Physical Volume
Partitions
Hard Disk
...
...
/
/home
lv-root
lv-home
vg-sys
pv1
/dev/sda2
/dev/sda3
/dev/sda

Example: expand



Example: expand

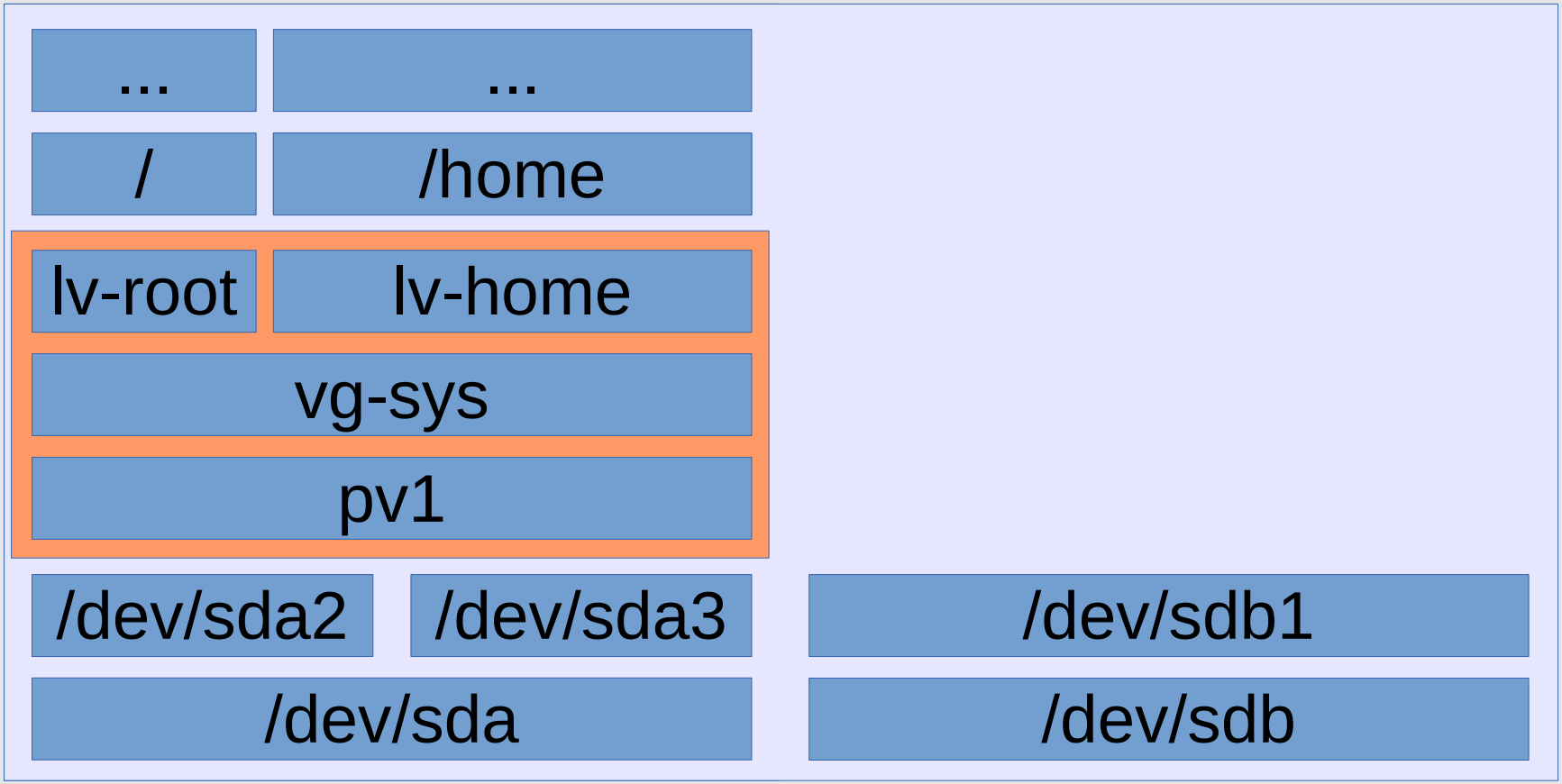
No space left on lv-home



Example: expand

No space left on lv-home

Add a new physical disk (sdb)

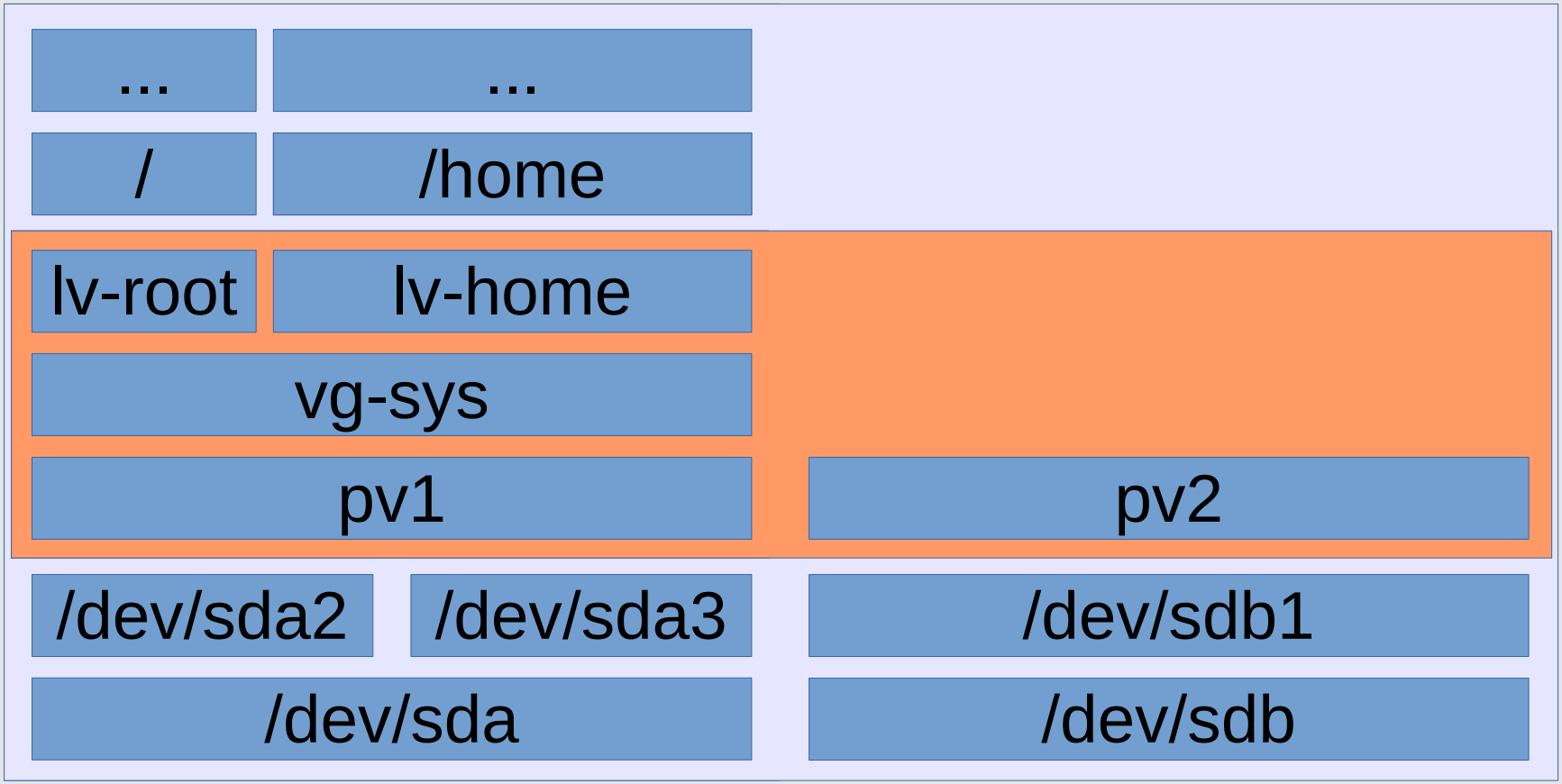


Example: expand

No space left on lv-home

Add a new physical disk (sdb)

Add the new disk to LVM as new PV



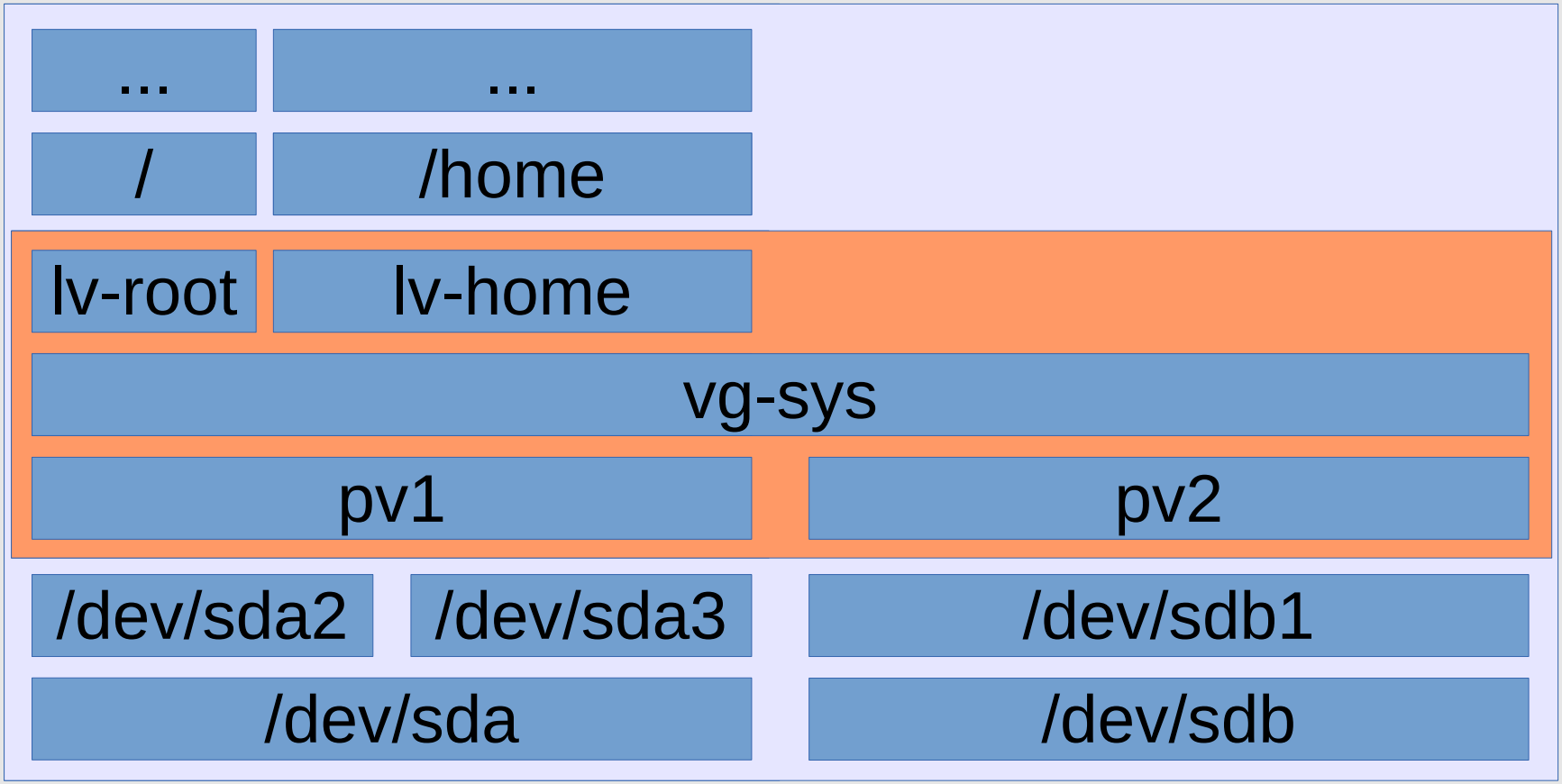
Example: expand

No space left on lv-home

Add a new physical disk (sdb)

Add the new disk to LVM as new PV

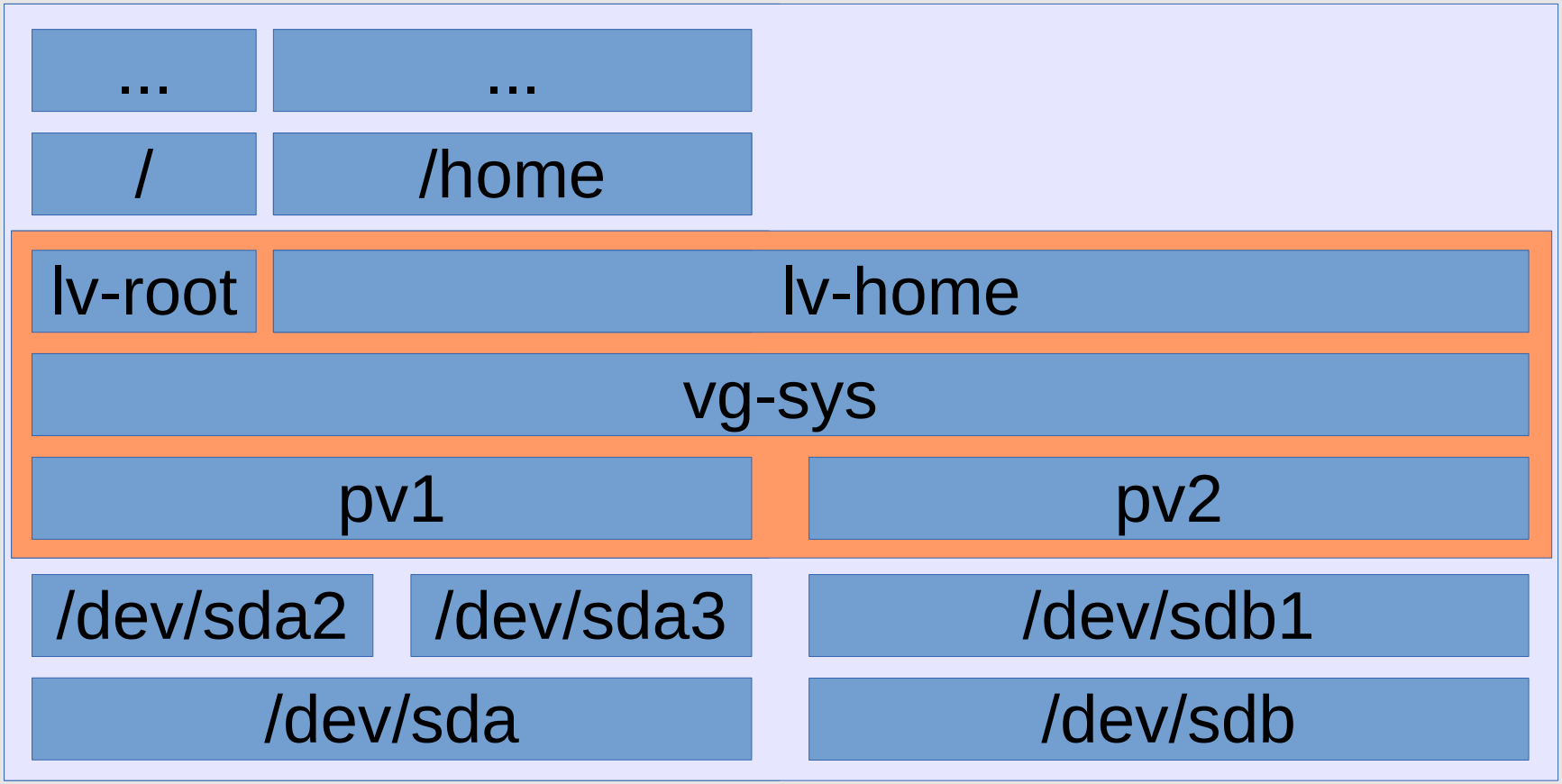
Expand the VG



Example: expand

No space left on lv-home
Add a new physical disk (sdb)
Add the new disk to LVM as new PV

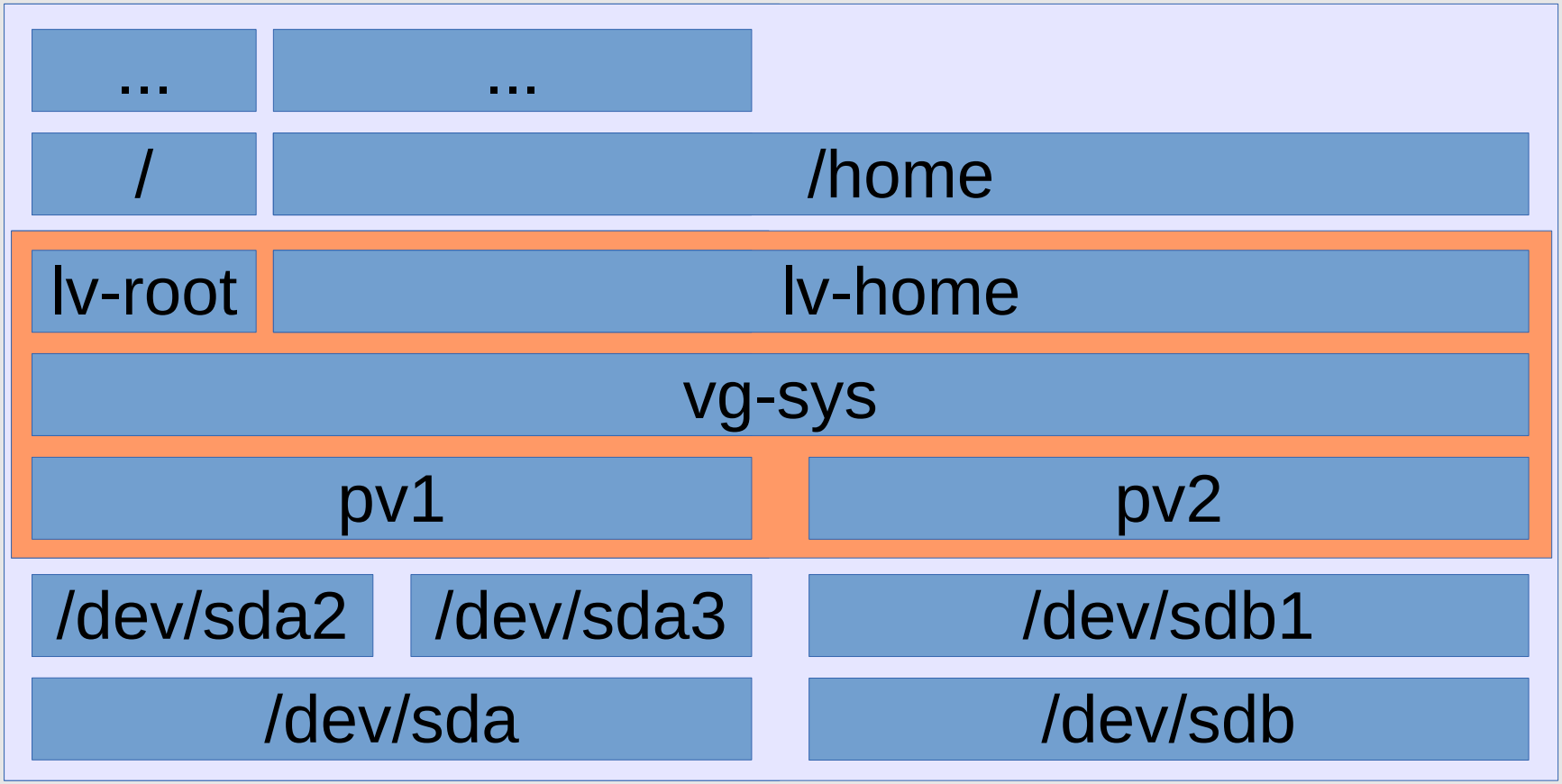
Expand the VG
Expand the LV



Example: expand

No space left on lv-home
Add a new physical disk (sdb)
Add the new disk to LVM as new PV

Expand the VG
Expand the LV
Resize the filesystem



LVM is not RAID

Not yet, but getting closer. LVM is now almost fully compatible with linux kernel MD interface (without *mdadm*):

LVM is not RAID

Not yet, but getting closer. LVM is now almost fully compatible with linux kernel MD interface (without *mdadm*):

- *lvcreate* supports --type raid[0-6,10], stripes, recovery rate, ...
- not well documented, not yet widely used as RAID solution, not supported by *grub*, less reliable (less tested) than *mdadm*
- manual rebuild vs automatic rebuild

LVM is not RAID

Not yet, but getting closer. LVM is now almost fully compatible with linux kernel MD interface (without *mdadm*):

- *lvcreate* supports --type raid[0-6,10], stripes, recovery rate, ...
- not well documented, not yet widely used as RAID solution, not supported by *grub*, less reliable (less tested) than *mdadm*
- manual rebuild vs automatic rebuild

mdadm + LVM is still “best practice”

MHPC
Master in High Performance Computing

Questions?

SISSA
ma per seguir virtute e canoscenza
ICTP

Commands:

test env setup, create, mkfs, mount

```
# dd if=/dev/zero of=/dev/shm/disk1 bs=1M count=0 seek=100
# dd if=/dev/zero of=/dev/shm/disk2 bs=1M count=0 seek=100
# losetup /dev/loop1 /dev/shm/disk1
# losetup /dev/loop2 /dev/shm/disk2
# pvcreate /dev/loop1
# pvcreate /dev/loop2
# vgcreate VGTEST /dev/loop1 /dev/loop2
# lvcreate -l 50%FREE -n LVTEST VGTEST
# lvresize -l+100%FREE /dev/VGTEST/LVTEST
# mkfs.ext4 -v /dev/VGTEST/LVTEST
# mkdir -vp /mnt/tmp
# mount /dev/VGTEST/LVTEST /mnt/tmp
# df /mnt/tmp
```

MHPC
Master in High Performance Computing

Commands:

display, lvextend, resizefs

```
# pvdisplay
# vgdisplay
# lvdisplay

# lvextend --extents +100%FREE /dev/VGTEST/LVTEST
# lvdisplay /dev/VGTEST/LVTEST

# umount /mnt/tmp
# fsck.ext4 -f -v /dev/VGTEST/LVTEST
# resize2fs /dev/VGTEST/LVTEST
# dumpe2fs -h /dev/VGTEST/LVTEST
# mount /dev/VGTEST/LVTEST /mnt/tmp
# df
```

SISSA
ICTP

Commands:

add disk, vg/lv extend, resizefs

```
# dd if=/dev/zero of=/dev/shm/disk3 bs=1M count=0 seek=100
# losetup /dev/loop3 /dev/shm/disk3
# pvcreate /dev/loop3
# vgdisplay
# vgextend VGTEST /dev/loop3
# vgdisplay
# lvextend --extents +100%FREE /dev/VGTEST/LVTEST
# lvdisplay
# umount /mnt/tmp
# fsck.ext4 -f -v /dev/VGTEST/LVTEST
# resize2fs /dev/VGTEST/LVTEST
```

Commands:

snapshot, remove, destroy test env

```
# mkdir -vp /mnt/tmp2
# lvcreate --size 10m --snapshot
--name SNAP /dev/VGTEST/LVTEST
# mount -r  /dev/VGTEST/SNAP
/mnt/tmp2/
# echo ciao > /mnt/tmp/testfile
# ls /mnt/tmp
# ls /mnt/tmp2
# umount /mnt/tmp2
# lvremove -f /dev/VGTEST/SNAP
```

```
# umount /mnt/tmp
# vgchange -a n VGTEST
```

(up to this point, non-destructive ops)

```
# lvremove /dev/VGTEST/LVTEST
# vgremove VGTEST
```

```
# losetup -d /dev/loop1
# losetup -d /dev/loop2
# losetup -a
# vgdisplay
# pvdisplay
# rm -fv /dev/shm/disk[12]
```