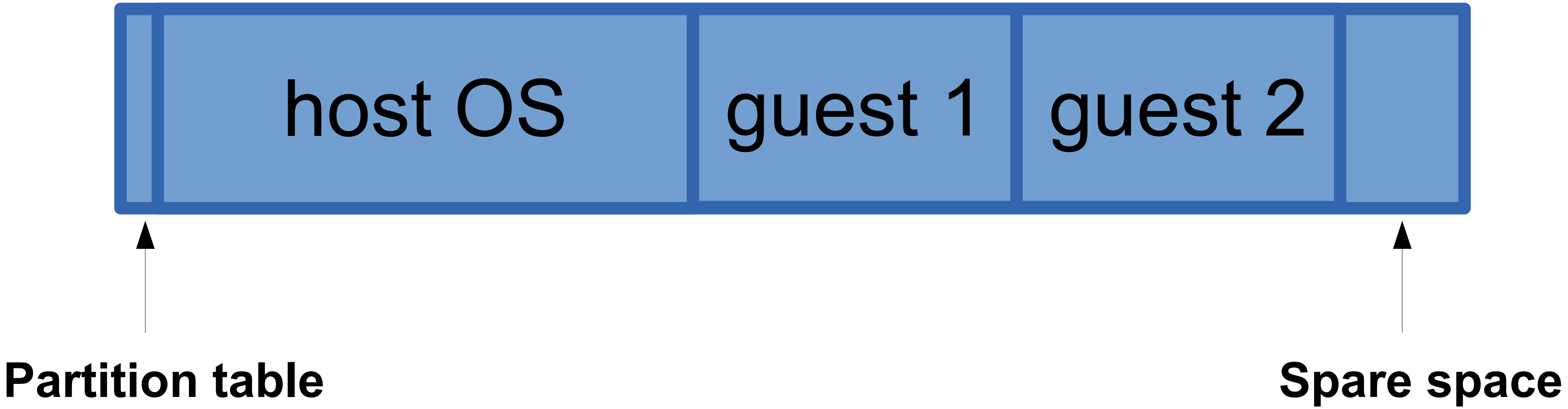
Introduction to LVM

NSRC

Problems with disk image files

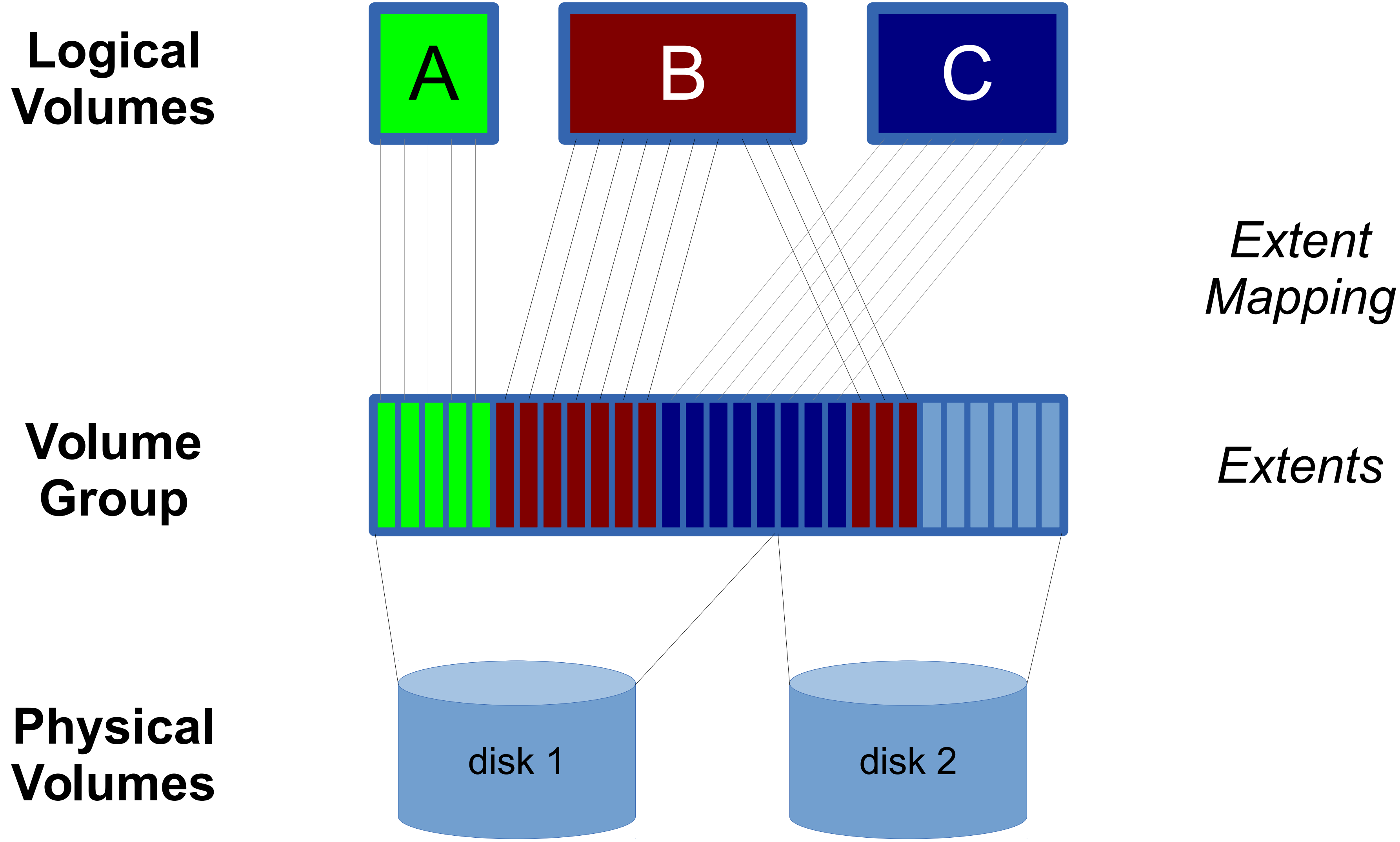
- 1. Overhead of passing through filesystem layers just to read/write blocks to disk
- 2. Possible unnecessary caching in host VFS
- 3. Risk of fragmentation of the host filesystem
  - leads to many more disk seeks and thus *much* worse performance
- Direct-to-disk access would give us better performance

Could we just use partitions?



- Certainly possible, but:
  - Partitions are a pain to manage / move
  - Partitions cannot span across drives

Solution: Logical Volume Manager



About LVM

- LVM stores data on "physical volumes"
  - Combined into "volume groups"
- Physical volumes divided into "extents"
  - usually 4MB
- Logical volume is a collection of extents
- You can grow a logical volume by adding extents
- When you remove a LV, its extents are freed and can be used for other LVs

About LVM

- LVM stores a small amount of metadata
  - small table of mappings from logical vols to extents
  - IDs to allow the physical volumes to be recognised and assembled into volume groups
- Extent mapping is very quick
- No need to move any data when adding, removing or resizing volume groups
- Can add new physical volumes to a vol group

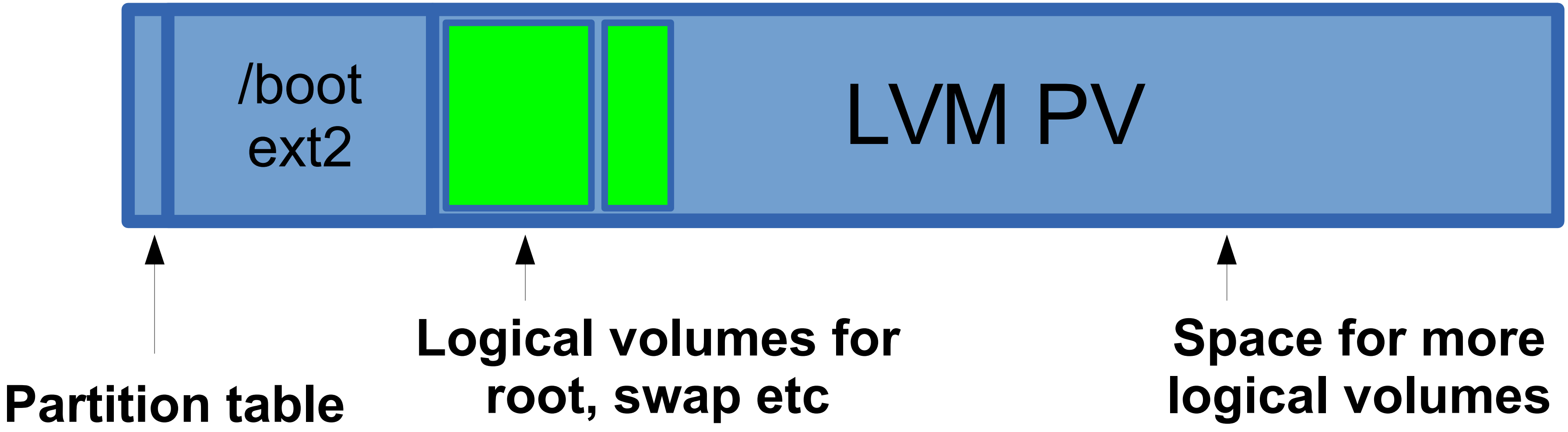
Accessing logical volumes

- Logical volumes appear as block devices
  - ***/dev/VOLGROUP/VOLUME or***
  - ***/dev/mapper/VOLGROUP-VOLUME***
- CLI tools in the "lvm2" package
  - `pvscan     # list all physical vols`
  - `lvscan     # list all logical vols`
  - `lvdisplay  # more detail`
  - `lvcreate --size 1G --name foo myvg`
  - `lvextend --size +512M /dev/myvg/foo`
  - `lvremove /dev/myvg/foo`

Note on physical volumes

- An LVM "physical volume" need not be an entire disk
- It can just be a partition
- Hence you can mix LVM and non-LVM on the same disk
- This is important if you don't have a separate boot disk

Partitioning and LVM

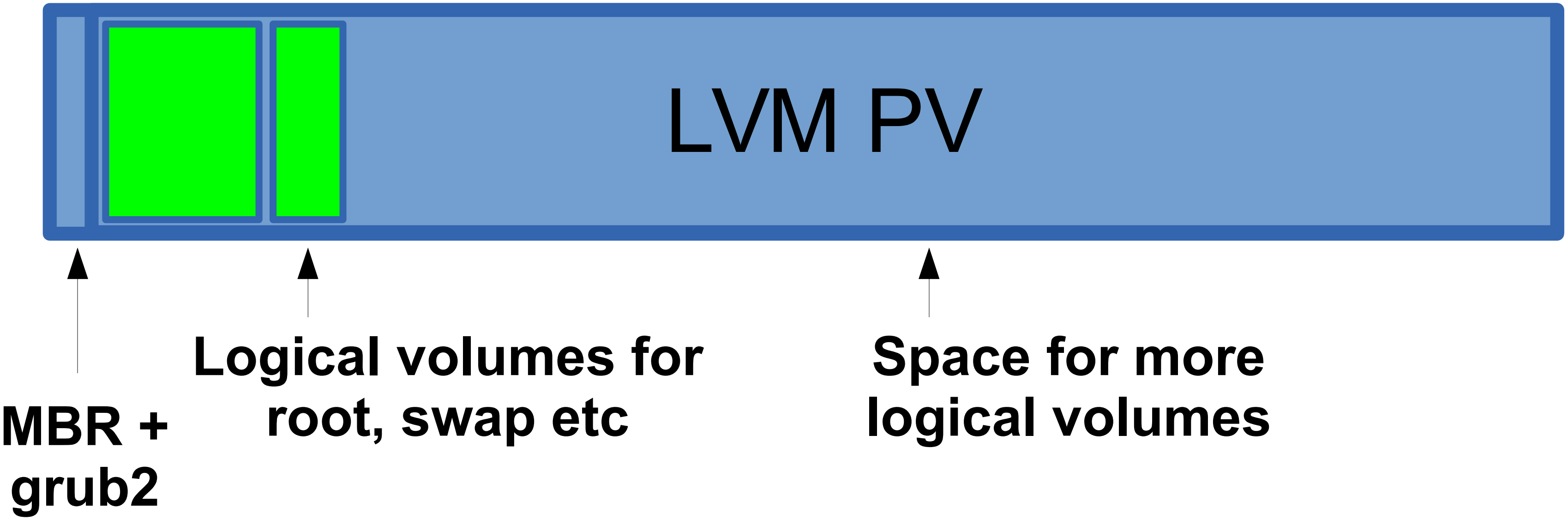

/boot
ext2
LVM PV
Partition table
Logical volumes for root, swap etc
Space for more logical volumes

- Partition table includes Master Boot Record
- sda1 (e.g. 1GB) partition for **/boot** filesystem
- sda2 (rest of disk) is LVM physical volume

Systems with grub2

- The kernel and initrd are stored under `/boot`
- grub2 is able to read the kernel from inside LVM
- So newer systems don't need a /boot partition
- Still need a partition for LVM though
  - grub2 is installed in the MBR and the blocks following it
  - first partition should start at offset 2048

Partitioning and LVM (grub2)


LVM PV
Logical volumes for root, swap etc
Space for more logical volumes
MBR + grub2

- blocks 0-2047 for MBR and grub2
- sda1 is LVM "physical volume"

Whole disk LVM

- Other boot options are possible
  - Separate bootable OS disk
  - Boot kernel from USB stick
  - Boot kernel over network (PXEboot)
- In these cases, you *could* make the whole data disk be a physical volume (no partition table)
- Simpler? You decide

Take care!

- Dealing with logical volumes like dealing with raw partitions, with the same dangers
- Easy to write to the wrong volume device!
  - especially if LVs have auto-assigned names
- Don't mount the same LV on the host *and* in a virtual machine, or in multiple VMs
  - Filesystem corruption is guaranteed *

* Unless you are using an esoteric cluster filesystem e.g. GFS, OCFS2

Summary

- LVM breaks disk space into 4MB extents
- Logical Volumes can be assembled out of any extents in a Volume Group
- A Volume Group can span multiple Physical Volumes
- Gives the speed of direct-to-disk access without the inflexibility of partitioning