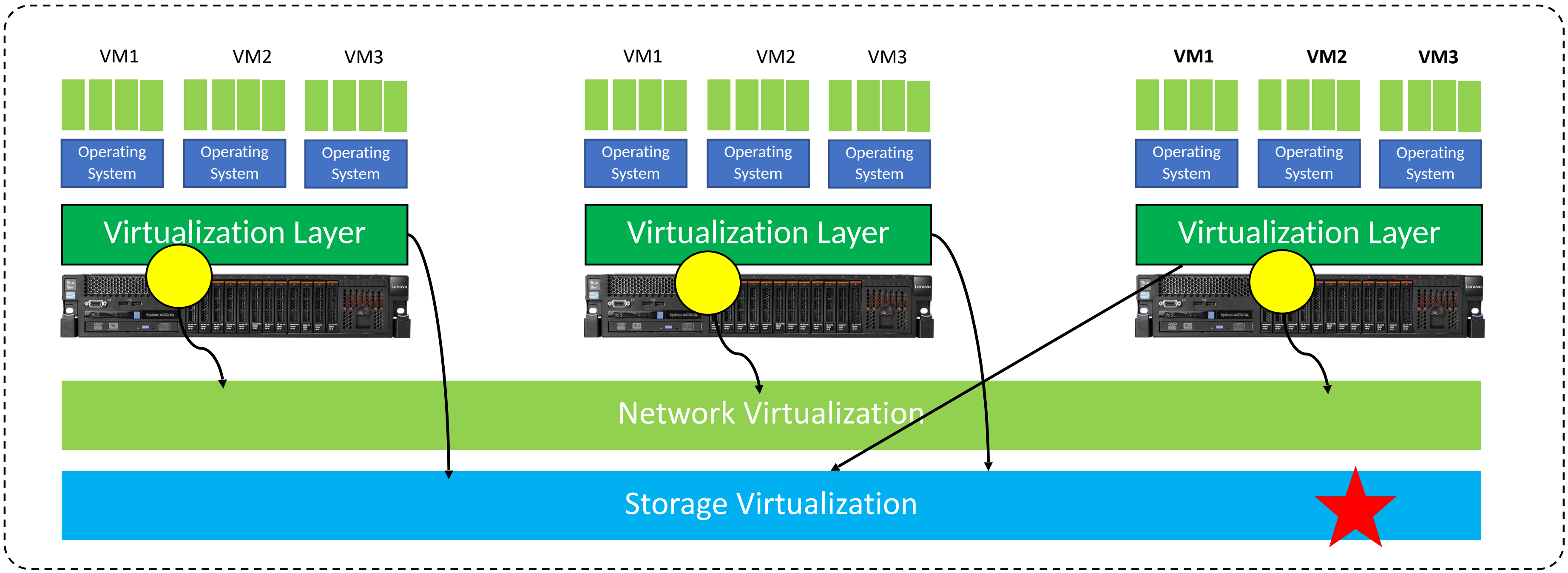
CS-452/552 Introduction to Cloud Computing

Storage Virtualization

Cloud operating system



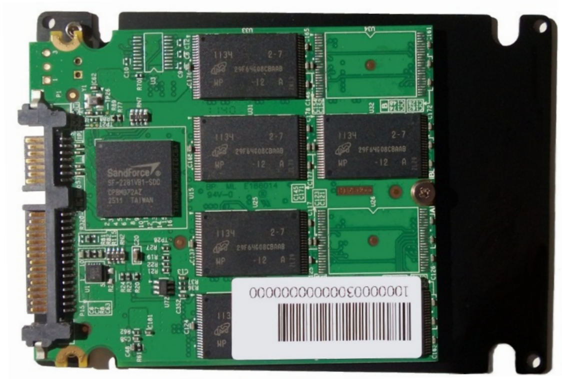
How to Provide Virtualization to Storage Resources?

Data Storage Systems

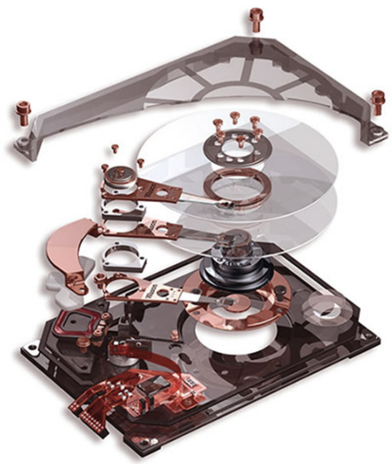
Data can be stored in various places in different manners
--- Hardware: CPU registers, caches, main memory and persistent storage
--- Software: File systems, object storage, databases (SQL databases and No-SQL databases.

Storage I/O system within a single host

Persistent Storage media



FLASH



HDD OR DISK DRIVE

I/O layers within a single host

User space

Kernel space

Applications

System Call Interfaces

VFS

File System (ext3, ext4, btrfs)

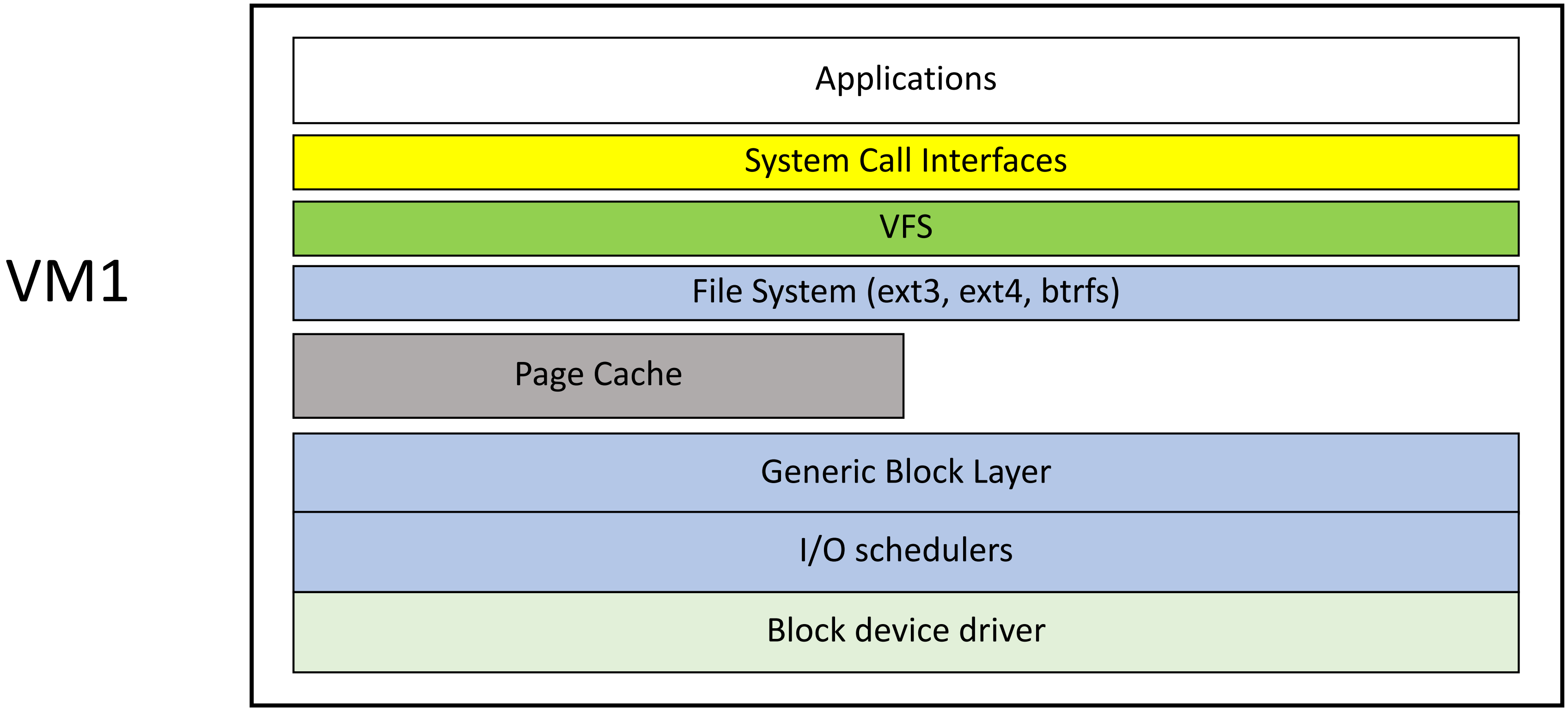
Page Cache

Generic Block Layer

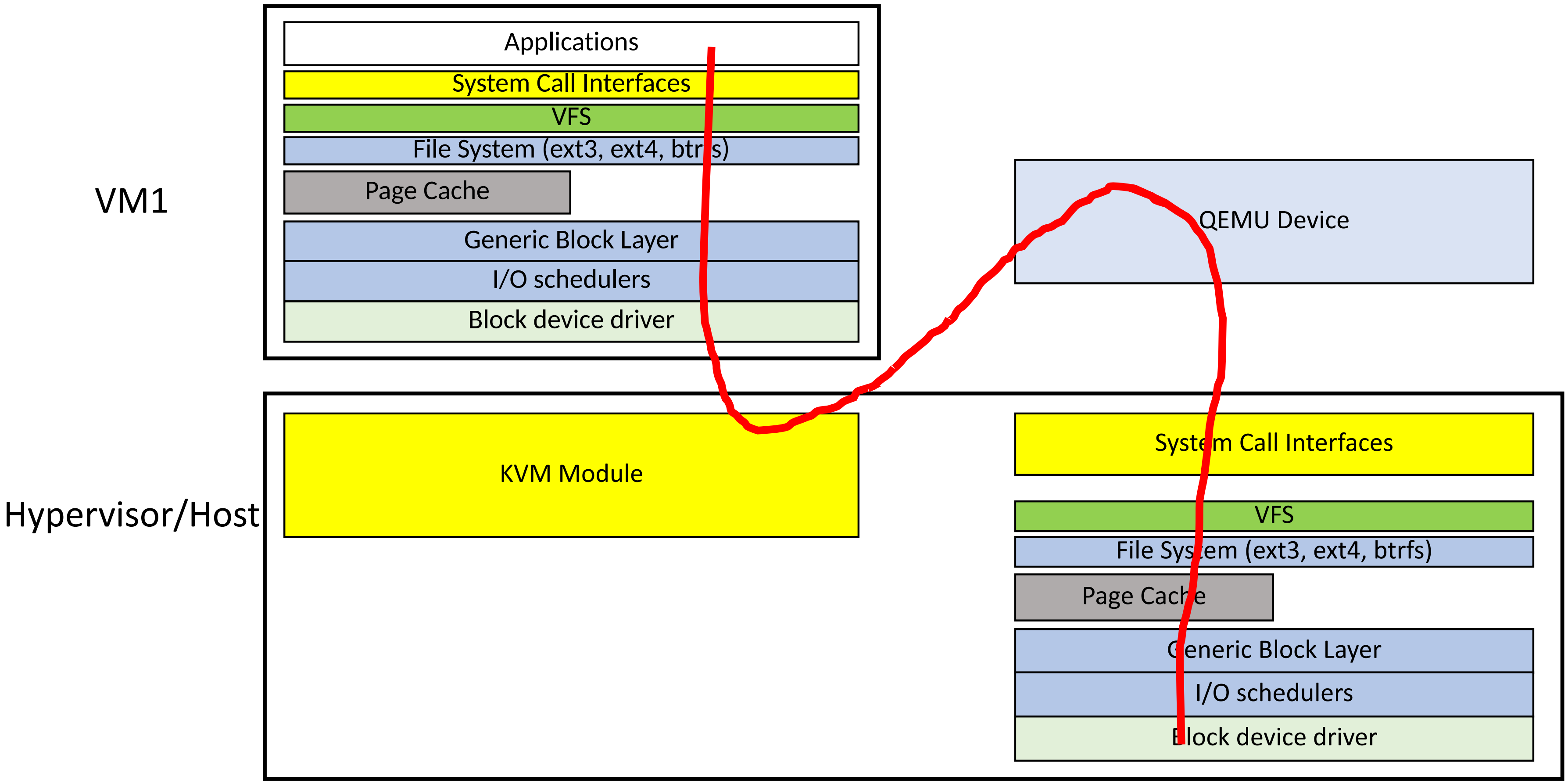
I/O schedulers

Block device driver

I/O layers within a VM

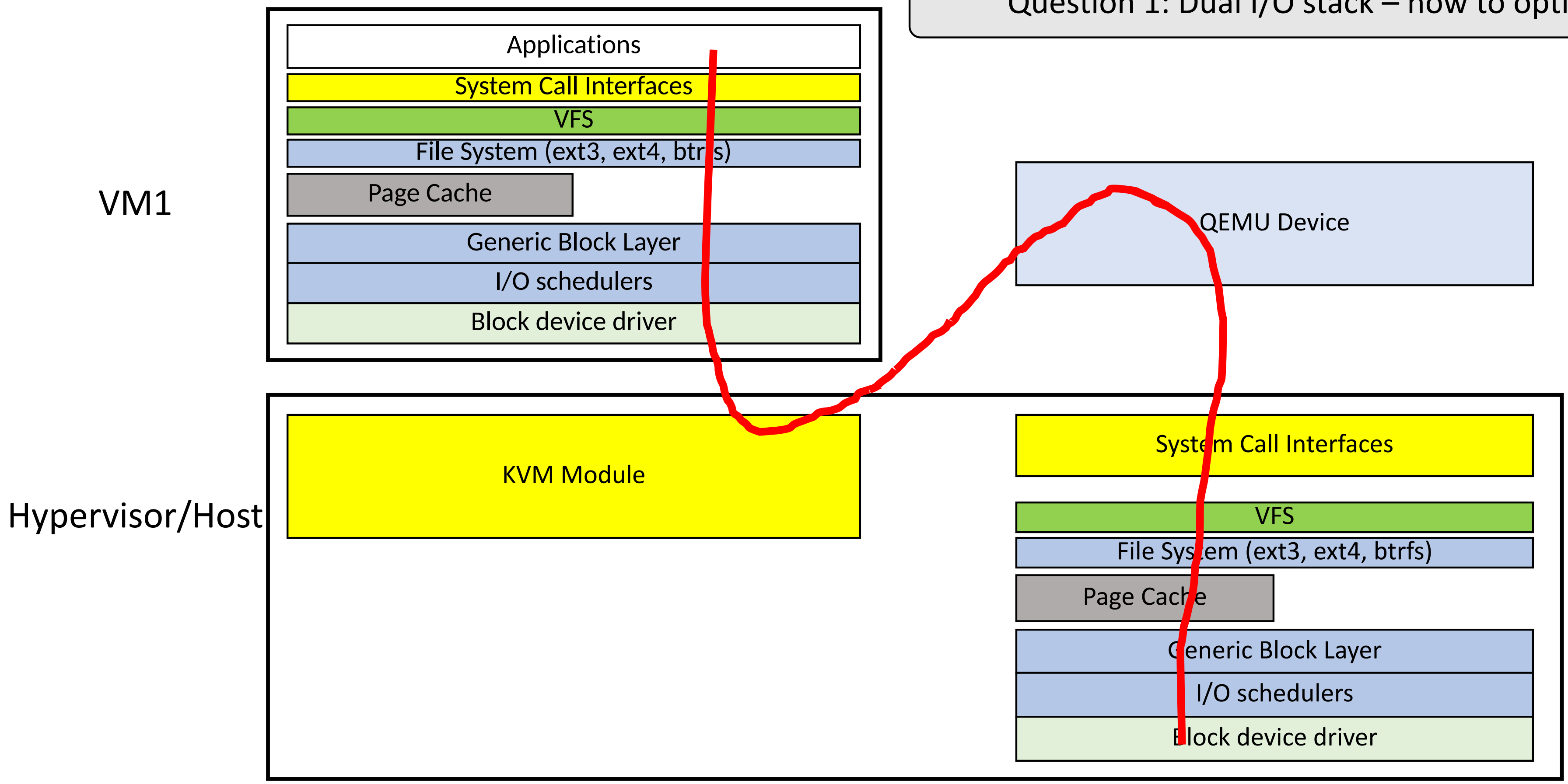


I/O layers in Virtualization



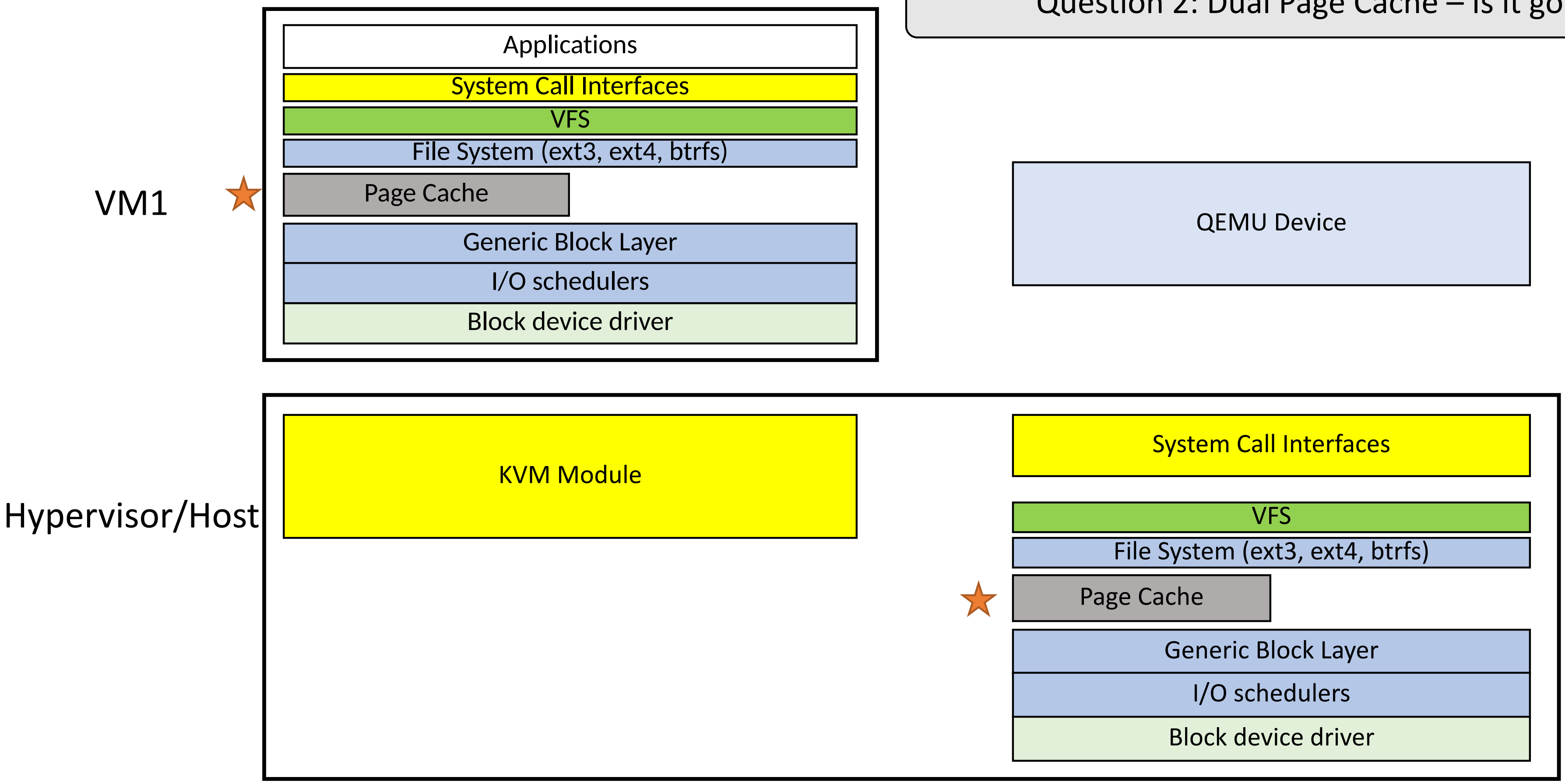
I/O layers in Virtualization

Question 1: Dual I/O stack – how to optimize?

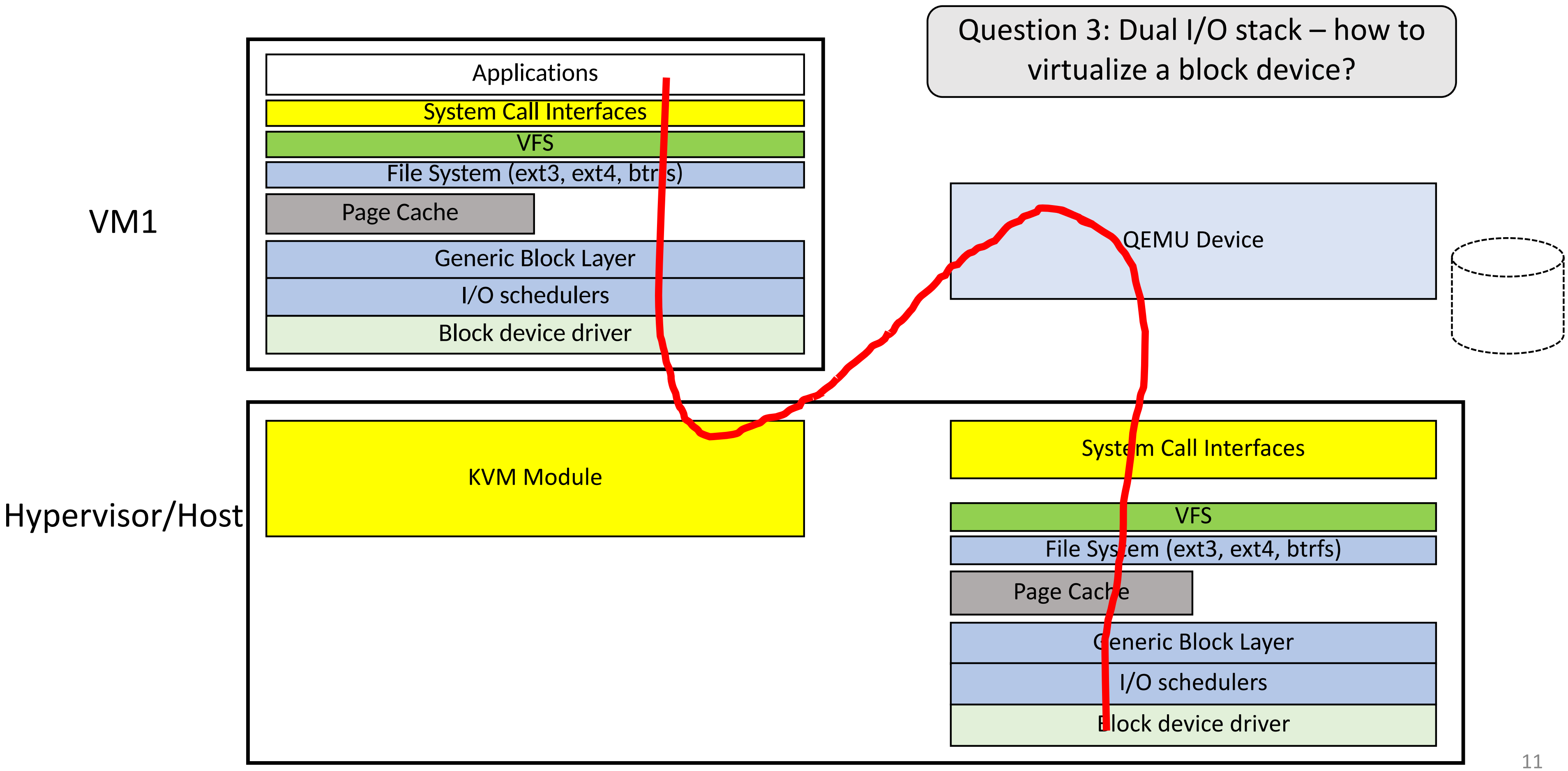


I/O layers in Virtualization

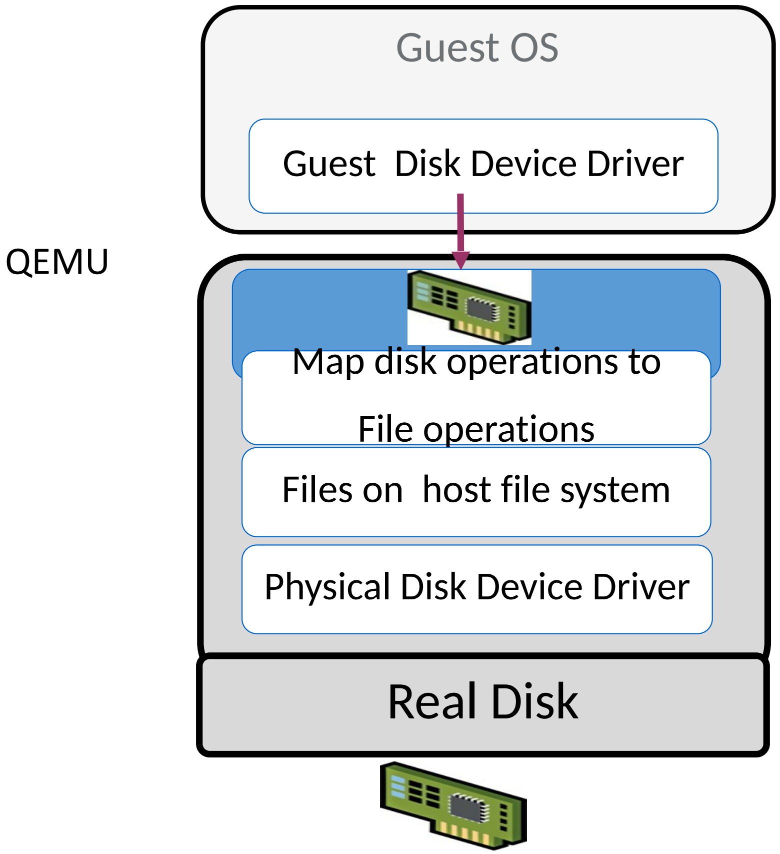
Question 2: Dual Page Cache – Is it good?



I/O layers in Virtualization

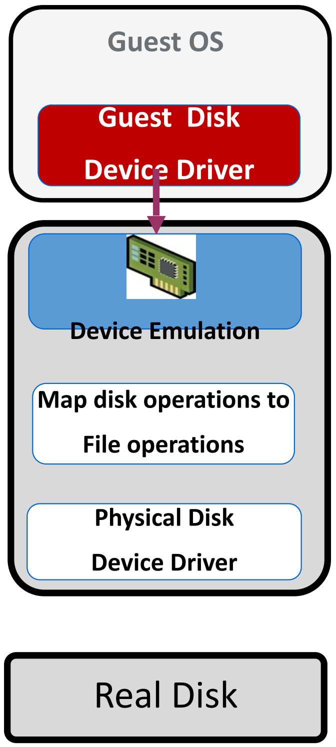


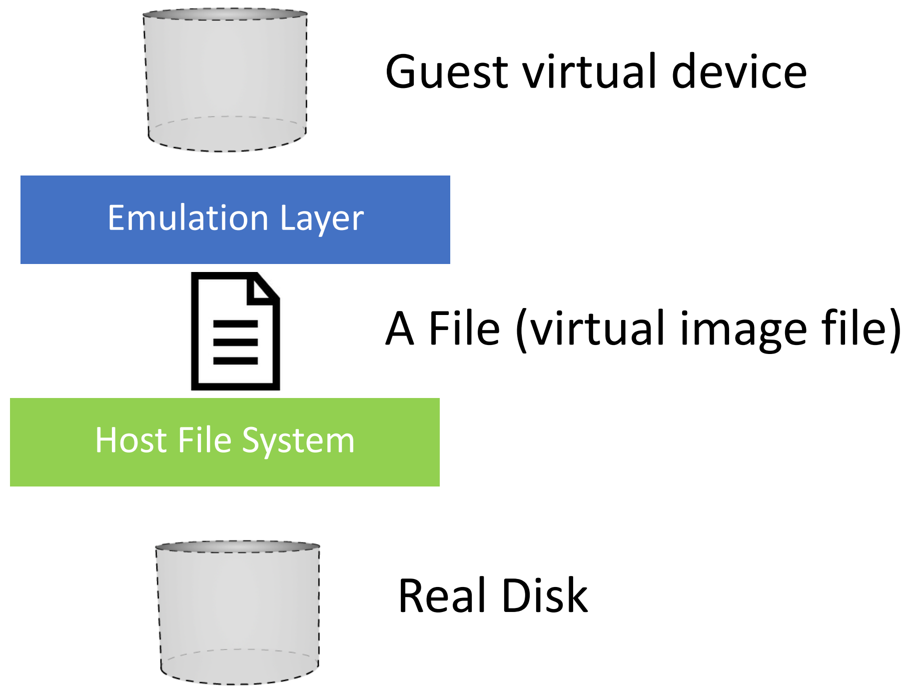
Virtualize Block Device



- Virtual disk is stored as a single file on the real file system of the host
- Operations to the block device is emulated by QEMU
- Guest issues block reads & writes
- Qemu converts them to file operations
- All these file operations, just like traditional file operations, operate on the virtual disk file

Virtualize Storage Device





Virtual Disk Image Type Matters!

- A “pre-allocated” disk image (1:1 blocks):
  - A 10 GB disk image reserves 10 GB of disk space, regardless of whether the virtual machine guests uses 1 GB or 10 GB (allocated at creation time)
- An “extensible” disk image, useful for growing on demand
  - From the VM point of view, it sees a full size disk, but the hypervisor is actually lying to the VM, and is allocating the disk blocks on the HOST side on demand

Disk images - pros / cons

- A “pre-allocated” disk image
  - Pros: Fast
  - Cons: Uses all space
- An extensible disk image
  - Pros: Less space
  - Cons: A bit overhead, fragmentation
- It depends on what we are trying to achieve: system design tradeoff

Discussion

- Assume that each virtual machine (VM) needs a disk image. If we are only going to create a single VM, it's easy:
  - Create VM
    - (1) create disk image
    - (2) attach ISO image (installation) to start VM
    - (3) install operating system
    - (4) Done!

- What if we want to install 2 VMs ? We could probably install a second time. What about when we have to build 5 ? 40 ? And do this very often (e.g., cloud service vendors)?
  - How do you increase the efficiency of such VM creation?

Two Concrete Techniques

- Raw disks ("pre-allocated")
  - Byte-for-byte disk image, byte 0 = byte 0 of the disk

- QEMU-KVM's "QCOW2" (Qemu Copy On Write, v.2) format (extensible)
  - Grow-on-demand
  - Compression support
  - Encryption support
  - Copy-on-write!

What is Copy-on-Write?

- Traditionally (e.g., raw disks):
  - When programs inside the guest VM write to the virtual disk, the changes are written to the disk image in place.

- Copy-on-write:
  - Write delta and store somewhere else (don’t modify the original copy)

Use of CoW

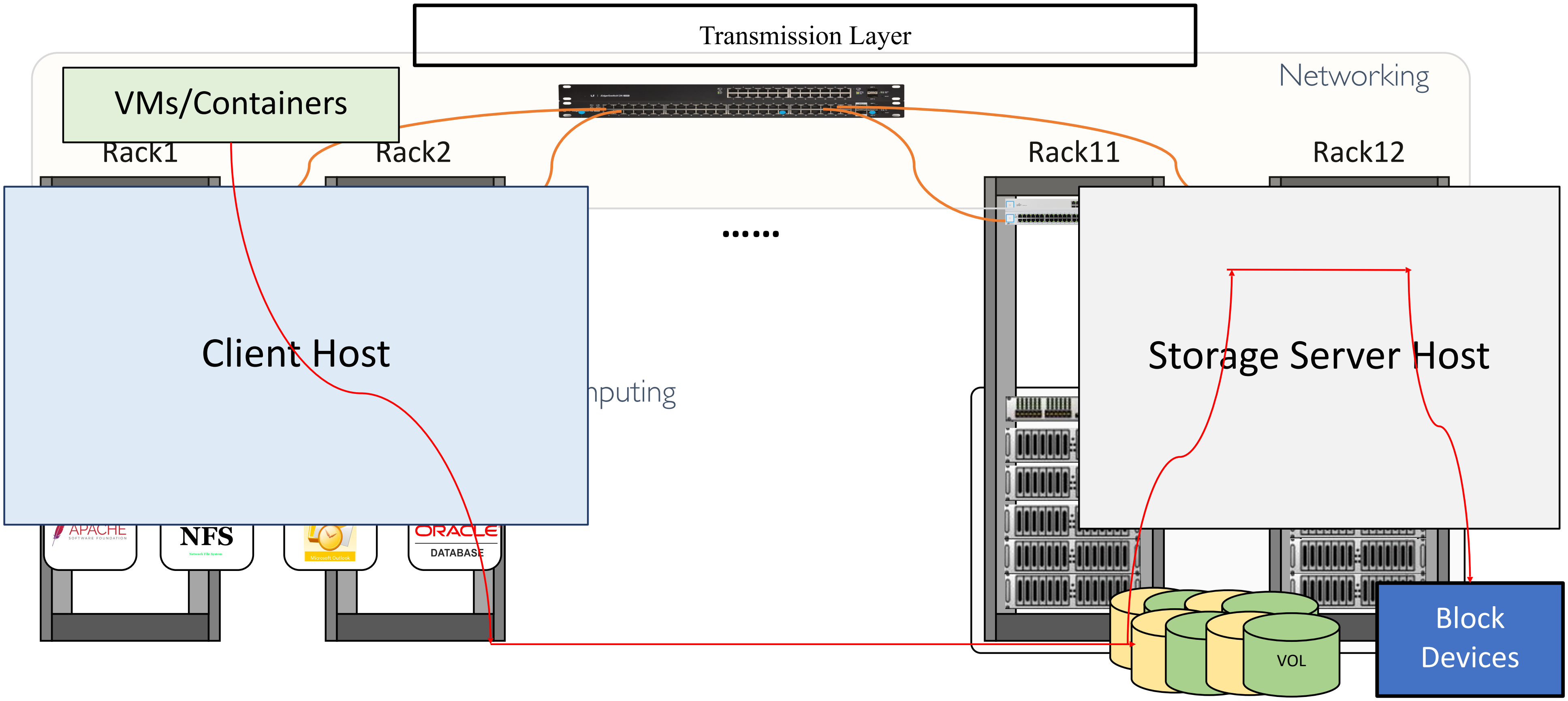
- A new disk image, originates from a “master” image as a backing file.
  - E.g, qemu-img create -o backing_file=master_image.qcow2 -f guest1.qcow2 10G

- Initially, the size of
  - guest1.qcow2 is 0.
  - backing file (master_image.qcow2) is 10 GB.

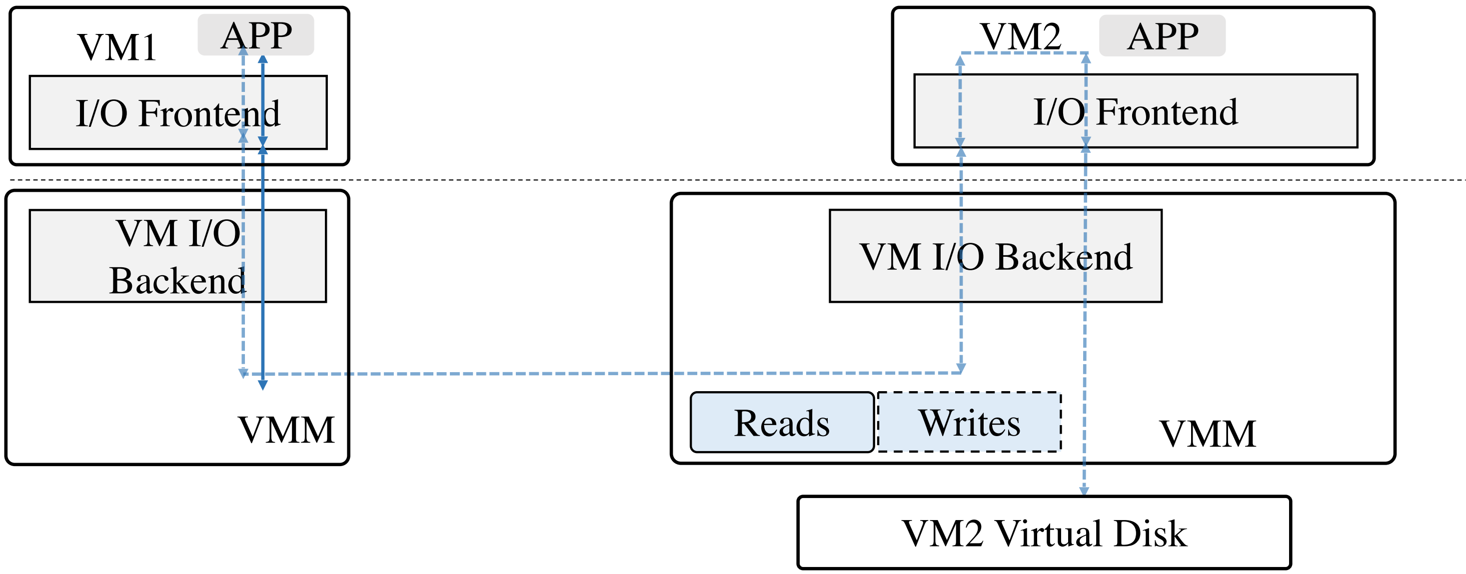
- For writes, KVM will write the changes to the guest1.qcow2. The file master image.qcow2 is never written to.

- For reads, KVM will read the block from the master image.qcow2 or guest1.qcow2 (whichever is latest).

Cloud Block Storage System



Example: I/O Data Plane Redundancy



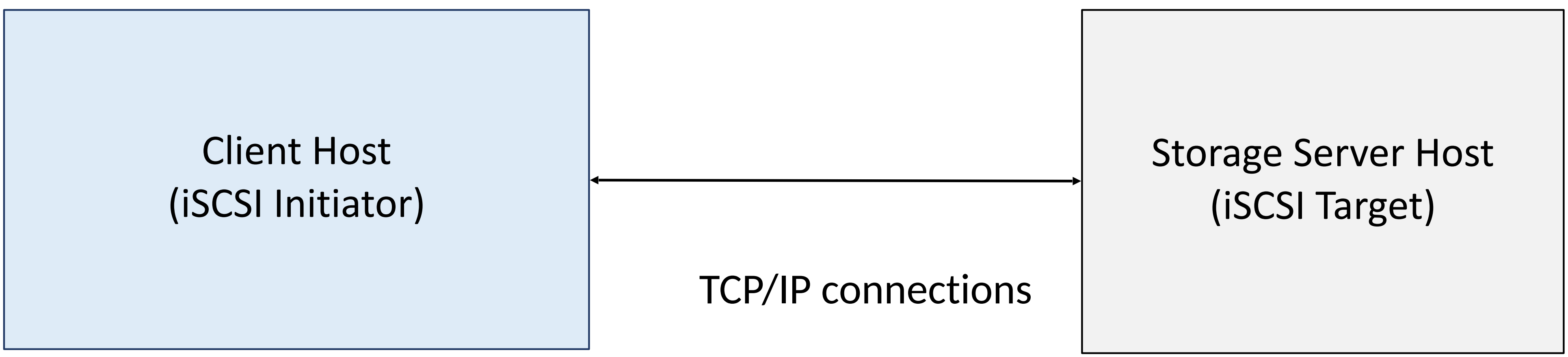
At least 5 data copying for data communication between two VMs!

Storage Area Network (SAN)

- Dedicated high-speed network interconnects and presents shared pools of storage devices to servers. (e.g., Fibre Channel)
- Light-weight solution: Protocols: iSCSI – Reuse Ethernet Network (by encapsulating SCSI commands into IP packets that don't require an FC connection)

iSCSI

- iSCSI is a Storage Area Network (SAN) protocol that allows for SCSI command transmission over a TCP/IP network
- iSCSI allows for the sharing of I/O devices over a long distance,
- iSCSI maintains the SCSI notion of an Initiator and Target device



Data Deduplication

- Duplicate data is deleted leaving, only one copy of the data to be stored.

- Data deduplication turns the incoming data into segments, uniquely identifies the data segments, and compares these segments to the data that has already been stored. If the incoming data is new data then it is stored on disk, but if it is a duplicate of what has already been stored then it is not stored again and a reference is created to it.

- “Only one unique instance of the data is actually retained on storage media (e.g., disk). Redundant data is replaced with a pointer to the unique data copy.”

Deduplication Methods

- In-line deduplication:
  - Hash calculations are created as the data is entered in real time.
  - If the target device identifies a block that has already been stored then it simply references to the existing block.

- Pros: Inline deduplication significantly reduces the raw disk capacity needed in the system since the full, not-yet-deduplicated data set is never written to disk

- Cons: However, “because hash calculations and lookups takes so long, data writes can be slower thereby reducing the backup throughput of the device.”

- What is off-line deduplication?

Sources

- Understanding Data Deduplication — and Why It's Critical for Moving Data to the Cloud: https://www.druva.com/blog/a-simple-definition-what-is-data-deduplication/
- Understanding data replication and its impact on business strategy: https://www.stitchdata.com/resources/data-replication/