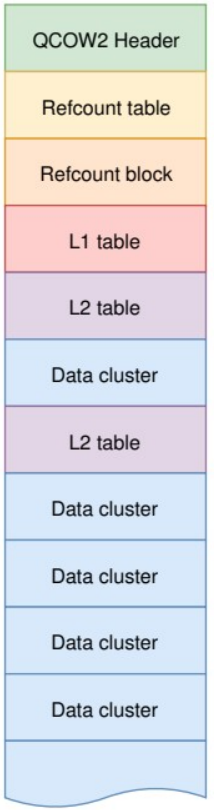
QCOW2 update

QCOW2 format

- QCOW2 vs. raw used to be features vs. performance but COW2's performance are now comparable to raw in most cases
- QCOW2 features include encryption, compression, internal COW snapshots, COW backing files (external snapshots)

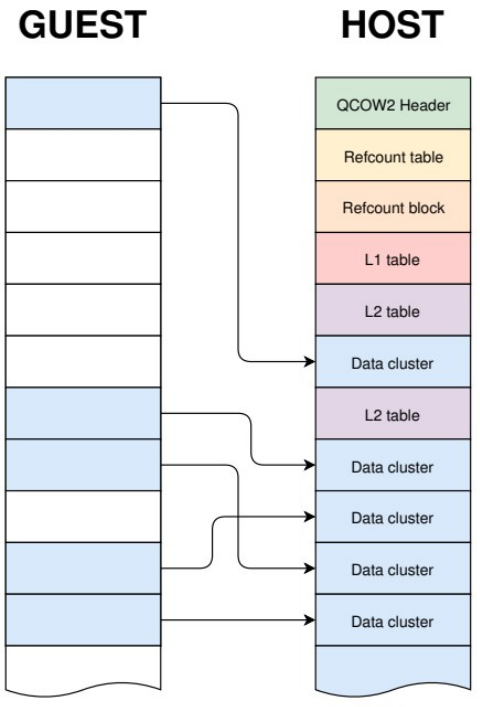
QCOW2 format

- QCOW2 vs. raw used to be features vs. performance but COW2's performance are now comparable to raw in most cases
- QCOW2 features include encryption, compression, internal COW snapshots, COW backing files (external snapshots)
- **File divided into "clusters" of equal size, default 64 KB**



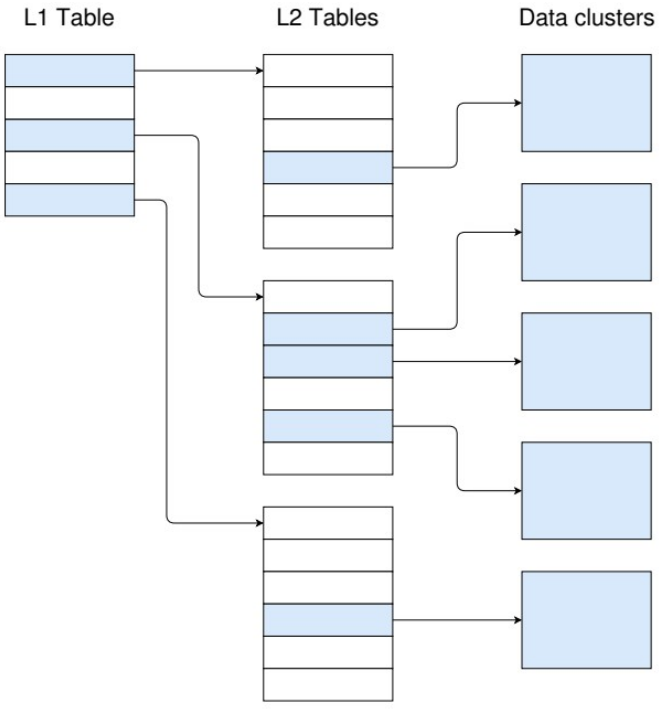
QCOW2 format

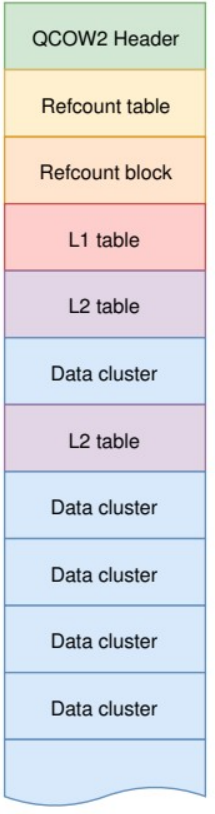
- QCOW2 vs. raw used to be features vs. performance but COW2's performance are now comparable to raw in most cases
- QCOW2 features include encryption, compression, internal COW snapshots, COW backing files (external snapshots)
- File divided into "clusters" of equal size, default 64 KB
- **Virtual disk seen by the VM divided into clusters of the same size, mapped to host clusters**



QCOW2 format

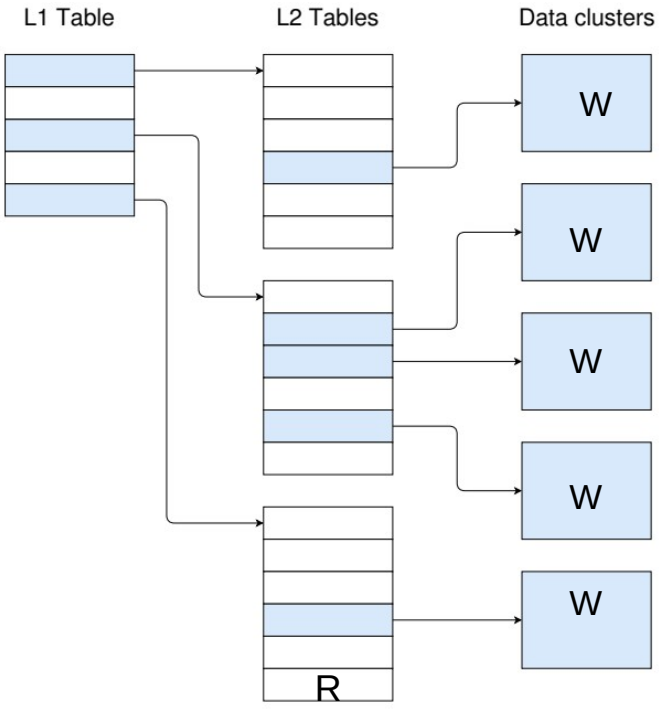
- L1/L2 tables used to map guest addresses (virtual disk sectors addressed by the guest driver) to host addresses (offset in the file)

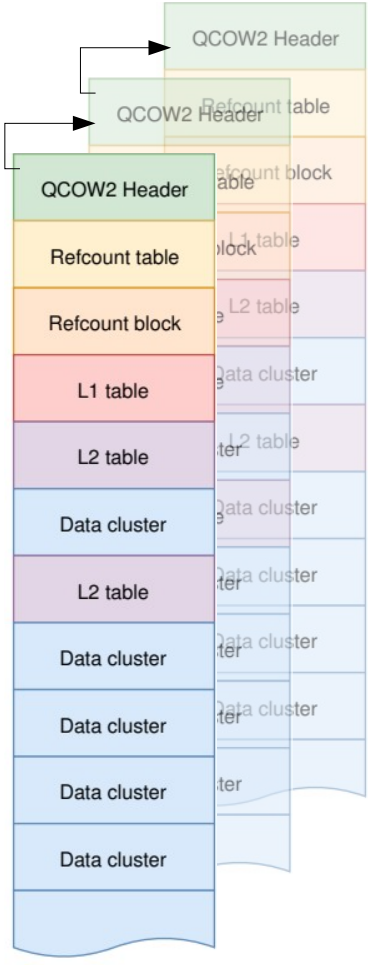




QCOW2 format

- L1/L2 tables used to map guest addresses (virtual disk sectors addressed by the guest driver) to host addresses (offset in the file)
- With backing files (snapshots), writes goes into the active layer, read server from the layer containing the last version





QCOW2 formats

- A few problems that I saw mentioned a lot and that I believe are **not** responsible of the performance issues we see
  - L2 cache size (I set it up to contain all L2)
  - L2 cache thrashing with old entries after snapshots (I reboot before each read test)
  - The need to read a COW cluster from backing file when only a subset is written

https://people.igalia.com/berto/files/kvm-forum-2017-slides.pdf

https://www.linux-kvm.org/images/9/92/Qcow2-why-not.pdf

https://blogs.igalia.com/berto/2015/12/17/improving-disk-io-performance-in-qemu-2-5-with-the-qcow2-l2-cache/

So what can it be?

- Trying to understand what happens exactly on a read request for a cluster which is very far down the chain, for example 300
  - I think we go through 300 L1 and L2 tables
  - Lookup seems to be in constant times and should be cached – at least for L2
    - What about L1 entries – they are supposed to be easy to cache because L1 table is small but now we have one L1 per snapshot