

### **cache=none**

- host doesn't do caching
- guest disk cache is writeback

This mode causes qemu-kvm to interact with the disk image file or block device with O\_DIRECT semantics, so the host page cache is bypassed and I/O happens directly between the qemu-kvm userspace buffers and the storage device. Because the actual storage device may report a write as completed when placed in its write queue only, the guest's virtual storage adapter is informed that there is a writeback cache, so the guest would be expected to send down flush commands as needed to manage data integrity. Equivalent to direct access to your hosts' disk, performance wise.

### **cache=writeback**

- host does use read cache
- guest disk cache mode is writeback

Writeback make a fsync for each write. So it's the more secure cache mode, you can't lose data. It's also the slower. This mode causes qemu-kvm to interact with the disk image file or block device with O\_DSYNC semantics, where writes are reported as completed only when the data has been committed to the storage device. The host page cache is used in what can be termed a writeback caching mode. The guest's virtual storage adapter is informed that there is no writeback cache, so the guest would not need to send down flush commands to manage data integrity. The storage behaves as if there is a writeback cache.

### **cache=directsync**

- host doesn't do caching
- guest disk cache mode is writeback

This mode causes qemu-kvm to interact with the disk image file or block device with both O\_DSYNC and O\_DIRECT semantics, where writes are reported as completed only when the data has been committed to the storage device, and when it is also desirable to bypass the host page cache. Like cache=writeback, it is helpful to guests that do not send flushes when needed. It was the last cache mode added, completing the possible combinations of caching and direct access semantics.

### **cache=writeback**

- host do read/write cache
- guest disk cache mode is writeback

This mode causes qemu-kvm to interact with the disk image file or block device with neither O\_DSYNC nor O\_DIRECT semantics,

so the host page cache is used and writes are reported to the guest as completed when placed in the host page cache, and the normal page cache management will handle commitment to the storage device. Additionally, the guest's virtual storage adapter is informed of the writeback cache, so the guest would be expected to send down flush commands as needed to manage data integrity.

Analogous to a raid controller with RAM cache.

### **cache=unsafe**

This mode is similar to the cache=writeback mode discussed above. The key aspect of this unsafe mode, is that all flush commands from the guests are ignored. Using this mode implies that the user has accepted the trade-off of performance over risk of data loss in the event of a host failure. Useful, for example, during guest install, but not for production workloads.