Backing Chain Management in libvirt and qemu

Eric Blake <eblake@redhat.com> KVM Forum, August 2015



In this presentation

- How does the qcow2 format track point-in-time snapshots
- What are the qemu building blocks for managing backing chains
- How are these building blocks used together in libvirt



Part I Understanding qcow2



qcow2 history

- qcow format (QEMU Copy On Write) documented in 2006
- qcow2 created in 2008, adding things like:
 - Internal snapshots with reference counting
- Hacky addition in 2009 to add header extensions
 - Backing file format, to avoid format probing CVEs
- qcow2v3 created in April 2012, adding things like:
 - Feature bits (extension is easier!)
 - Efficient zero cluster management



Let's look under the hood

- Create a new file
- Write some guest data
- Create an internal snapshot
- Write more guest data
- Create an external snapshot
- Write even more guest data

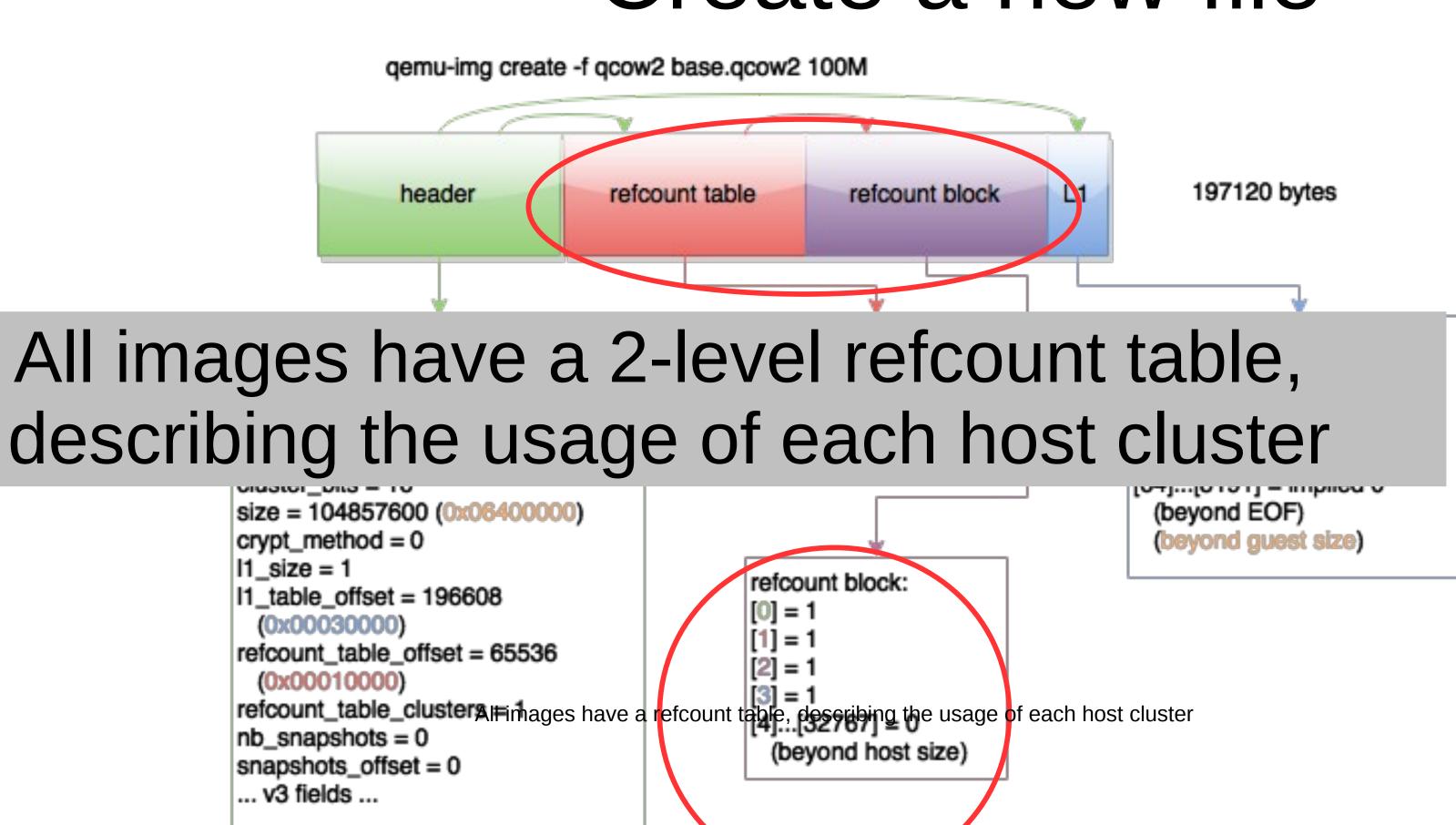


qemu-img create -f qcow2 base.qcow2 100M

qemu-img create -f qcow2 base.qcow2 100M 197120 bytes refcount block L1 header refcount table refcount table: L1 table: header: [0] = 131072magic = "QFI\xfb" $[0] = \{ ref_is_one = 0,$ (0x000200000)version = 3 offset = 0x000000000[1]...[8191] = 0 backing_file_offset = 0 [1]...[63] = 0 (beyond host size) (beyond guest size) backing_file_size = 0 [64]...[8191] = implied 0 cluster_bits = 16 size = 104857600 (0x06400000) (beyond EOF) (beyond guest size) $crypt_method = 0$ 11_size = 1 refcount block: I1_table_offset = 196608 [0] = 1(0x00030000)[1] = 1 refcount_table_offset = 65536 [2] = 1(0x00010000) [3] = 1refcount_table_clusters = 1 [4]...[32767] = 0 nb_snapshots = 0 (beyond host size) $snapshots_offset = 0$... v3 fields ...

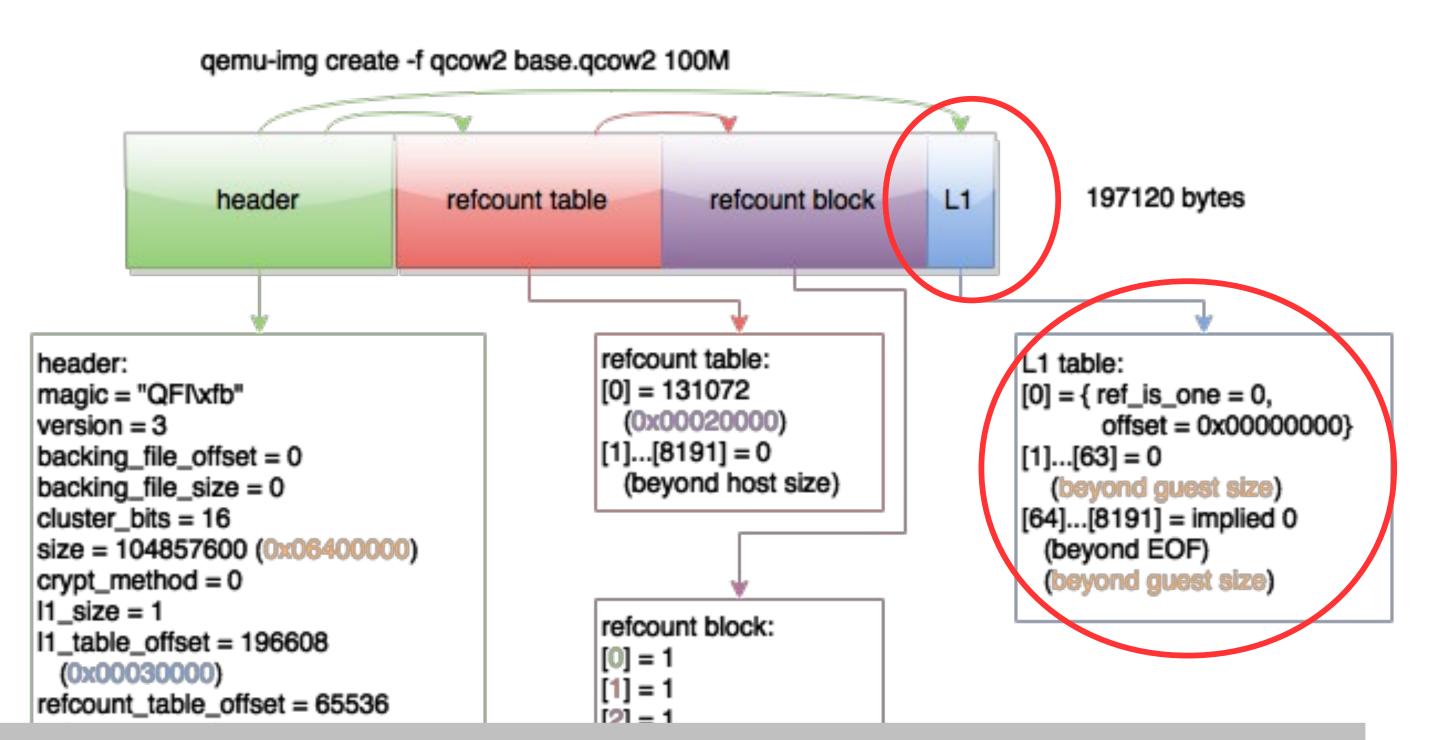
Guest sees:

104857600 NUL bytes



Guest sees:

104857600 NUL bytes

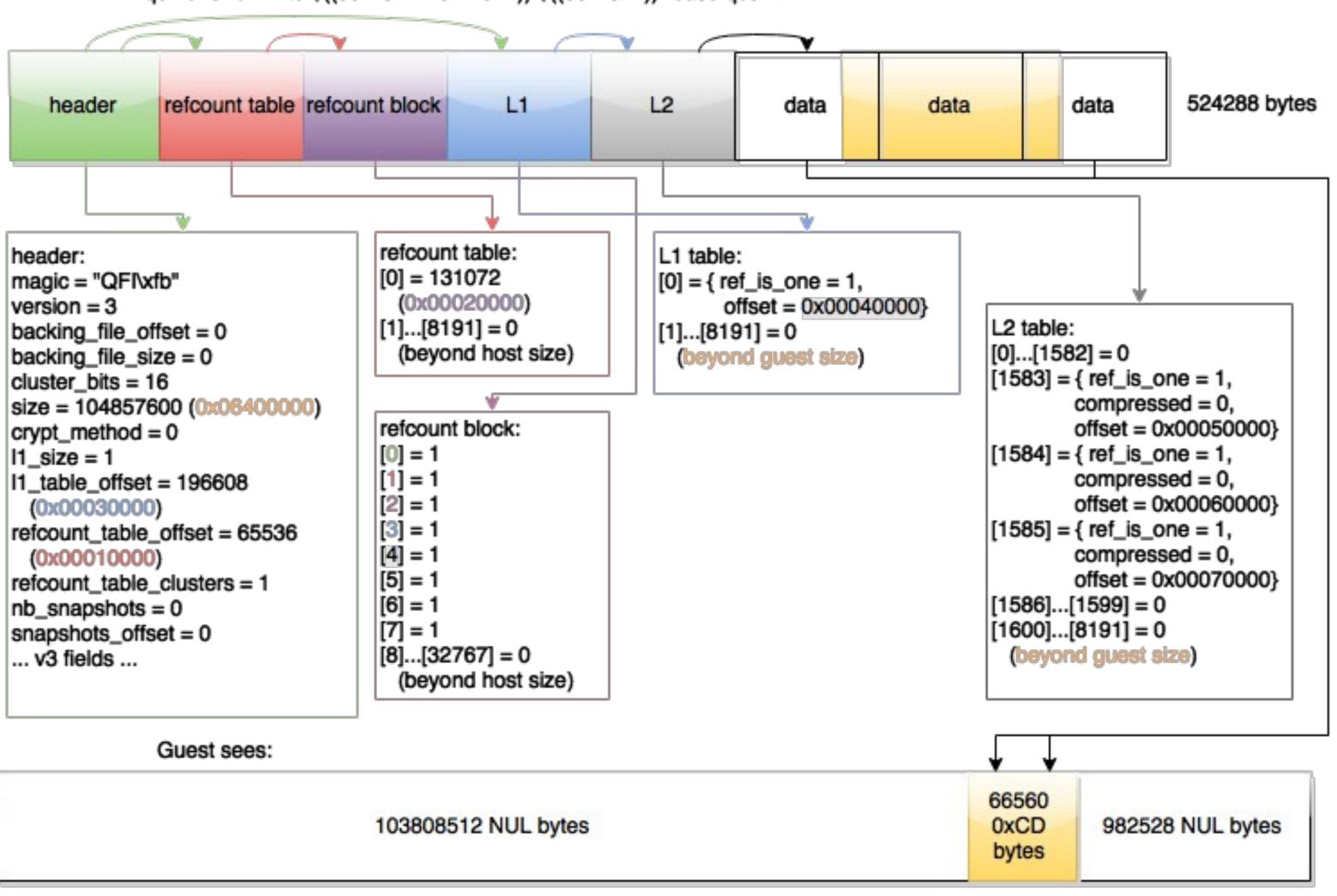


All images have an L1/L2 table, describing the mapping of each guest cluster (but with no data mapped, L2 is omitted)

104857600 NUL bytes

qemu-io -c "write \$((99*1024*1024-512)) \$((65*1024))" base.qcow2

qemu-io -c "write \$((99*1024*1024-512)) \$((65*1024))" base.qcow2



qemu-io -c "write \$((99*1024*1024-512)) \$((65*1024))" base.qcow2 524288 bytes refcount table refcount block header L1 L2 data data data Refcount table tracks additional clusters [1]...[8191] = 0 backing_file_offset = 0 L2 table: [1]...[8191] = 0 [0]...[1582] = 0 (beyond host size) backing_file_size = 0 (beyond guest size) [1583] = { ref_is_one = 1, cluster_bits = 16 compressed = 0,size = 104857600 (0x06400000) refcount block: offset = 0x00050000} $crypt_method = 0$ [0] = 1[1584] = { ref_is_one = 1, $11_size = 1$ [1] = 1compressed = 0,I1_table_offset = 196608 [2] = 1offset = 0x00060000} (0x00030000)[1585] = { ref_is_one = 1, [3] = 1 refcount_table_offset = 65536 [4] = 1compressed = 0,(0x00010000)[5] = 1offset = 0x00070000} refcount_table_clusters = 1 [6] = 1[1586]...[1599] = 0 nb_snapshots = 0 [7] = 1[1600]...[8191] = 0 snapshots_offset = 0 [8]...[32767] = 0 (beyond guest size) ... v3 fields ... (beyond host size) Guest sees: 66560 982528 NUL bytes 0xCD 103808512 NUL bytes bytes

qemu-io -c "write \$((99*1024*1024-512)) \$((65*1024))" base.qcow2 324288 bytes refcount table refcount block L1 L2 data data data header refcount table: header: L1 table: [0] = 131072magic = "QFI\xfb" [0] = { ref_is_one = 1, (0x00020000)version = 3 offset = 0x00040000} L2 table: [1]...[8191] = 0 backing_file_offset = 0 [1]...[8191] = 0 [0]...[1582] = 0 (beyond host size) backing_file_size = 0 (beyond guest size) [1583] = { ref_is_one = 1, L2 table tracks guest data in compressed = 0,offset = 0x00050000} [1584] = { ref_is_one = 1, aligned clusters compressed = 0,offset = 0x00060000} [1585] = { ref_is_one = 1, TEICOUTIL_LADIE_UTISEL = 00000 [4] = 1compressed = 0,(0x00010000) offset = 0x00070000[5] = 1refcount_table_clusters = 1 [1586]...[1599] = 0 [6] = 1nb_snapshots = 0 [7] = 1[1600]...[8191] = 0 snapshots_offset = 0 [8]...[32767] = 0beyond guest size) ... v3 fields ... (beyond host size) Guest sees: 66560 982528 NUL bytes 0xCD 103808512 NUL bytes bytes