Live Block Operations:

Snapshots, Merging, and Mirroring

Jeff Cody

Red Hat

KVM Forum 2012, Barcelona

What this covers

- Background of live block operations
- Live snapshots
- Live snapshot merge:
  - Block Stream
    - QEMU v1.1
  - Block Commit
    - QEMU v1.3
- Drive Mirroring
  - QEMU v1.3

Live Block Operations

Live Block Operations

- Manipulate block storage devices and data, while guest is running
- Can be synchronous, or asynchronous
  - Synchronous operations occur in QAPI handler
  - Asynchronous operations use block jobs

Live Snapshots

Live Snapshots

- Always synchronous
- Refers to 'external' snapshots only
  - New snapshot must be an image format that supports backing files
- Transactional QMP Command, and atomic across multiple devices
  - Since 1.1

Live Snapshots

Two methods:

- Multiple devices
- Single device

Live Snapshots

Two methods:

- Multiple devices
- Single device

Example:

```
{ "execute": "transaction", "arguments":
  {'actions': [
    { 'type': 'blockdev-snapshot-sync', 'data' :
      { 'device': 'virtio0', 'snapshot-file': '/tmp/driveA-snp-1.img' } },
    { 'type': 'blockdev-snapshot-sync', 'data' :
      { 'device': 'virtio1', 'snapshot-file': '/tmp/driveB-snp-1.img' } } ] } }
```

Live Snapshots

Two methods:

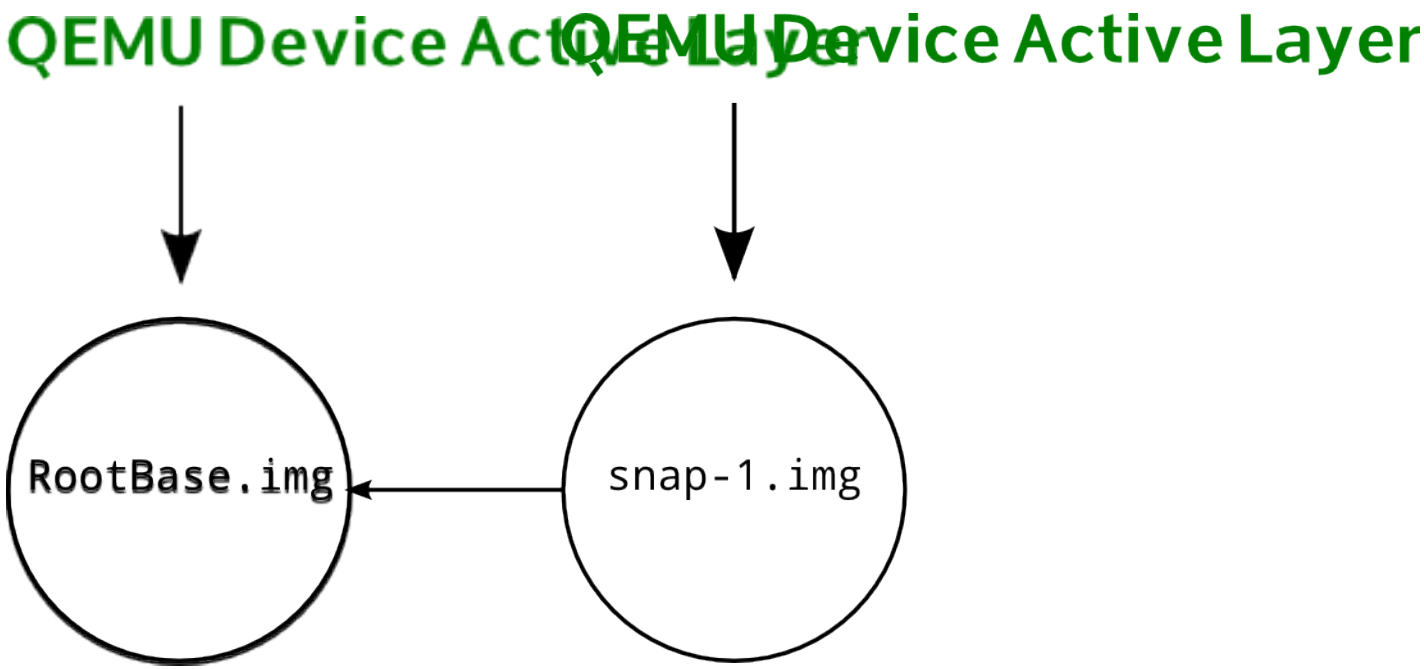
- Multiple devices
- Single device

Single Drive Example:

```
{ "execute": "transaction", "arguments":
  {'actions': [
    { 'type': 'blockdev-snapshot-sync', 'data' :
      { 'device': 'virtio0', 'snapshot-file': '/tmp/driveA-snp-1.img' } }] } }
```

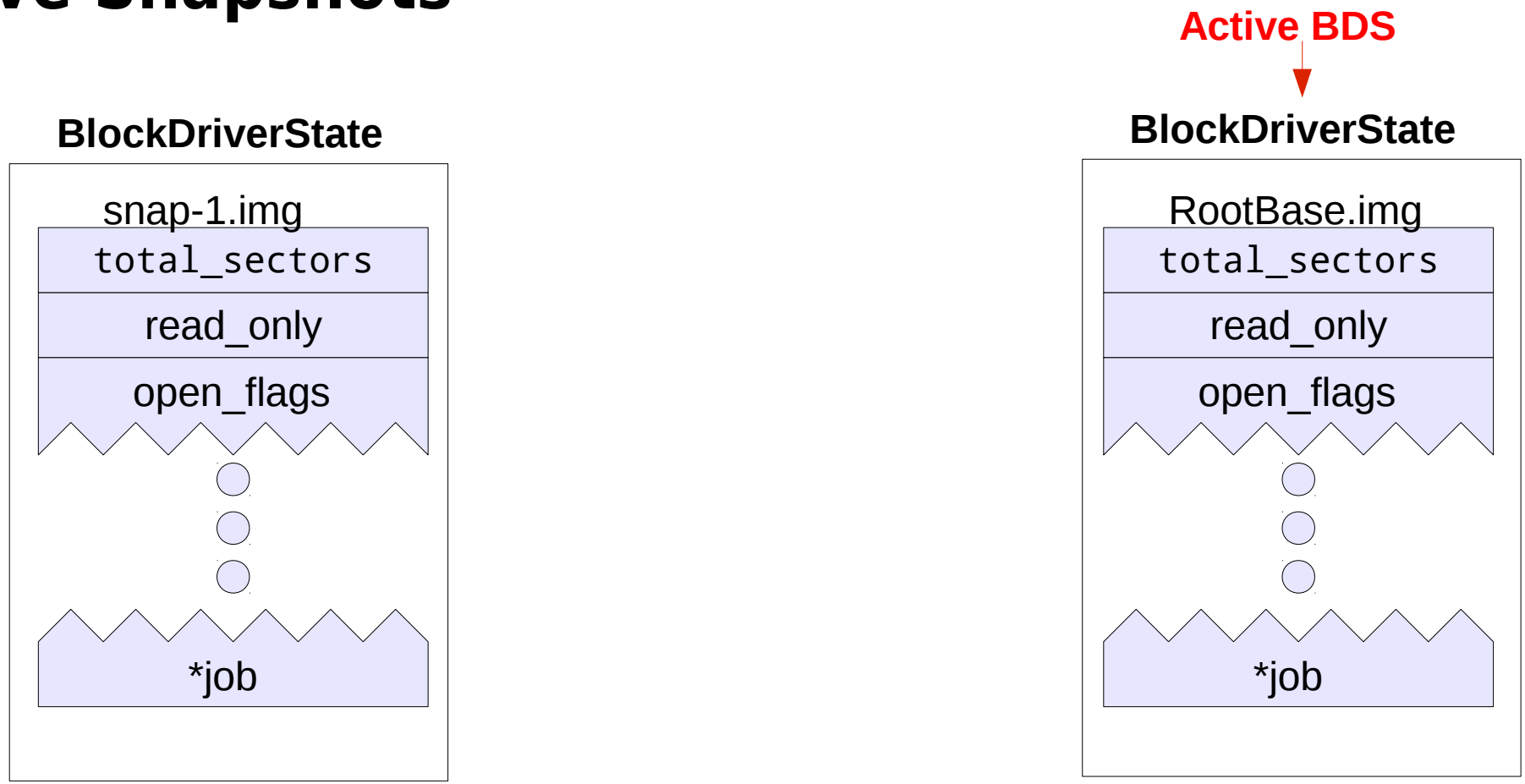
```
{ "execute": "blockdev-snapshot-sync", "arguments":
   { "device": "virtio0","snapshot-file":"/tmp/driveA-snp-1.img"} }
```

Live Snapshots

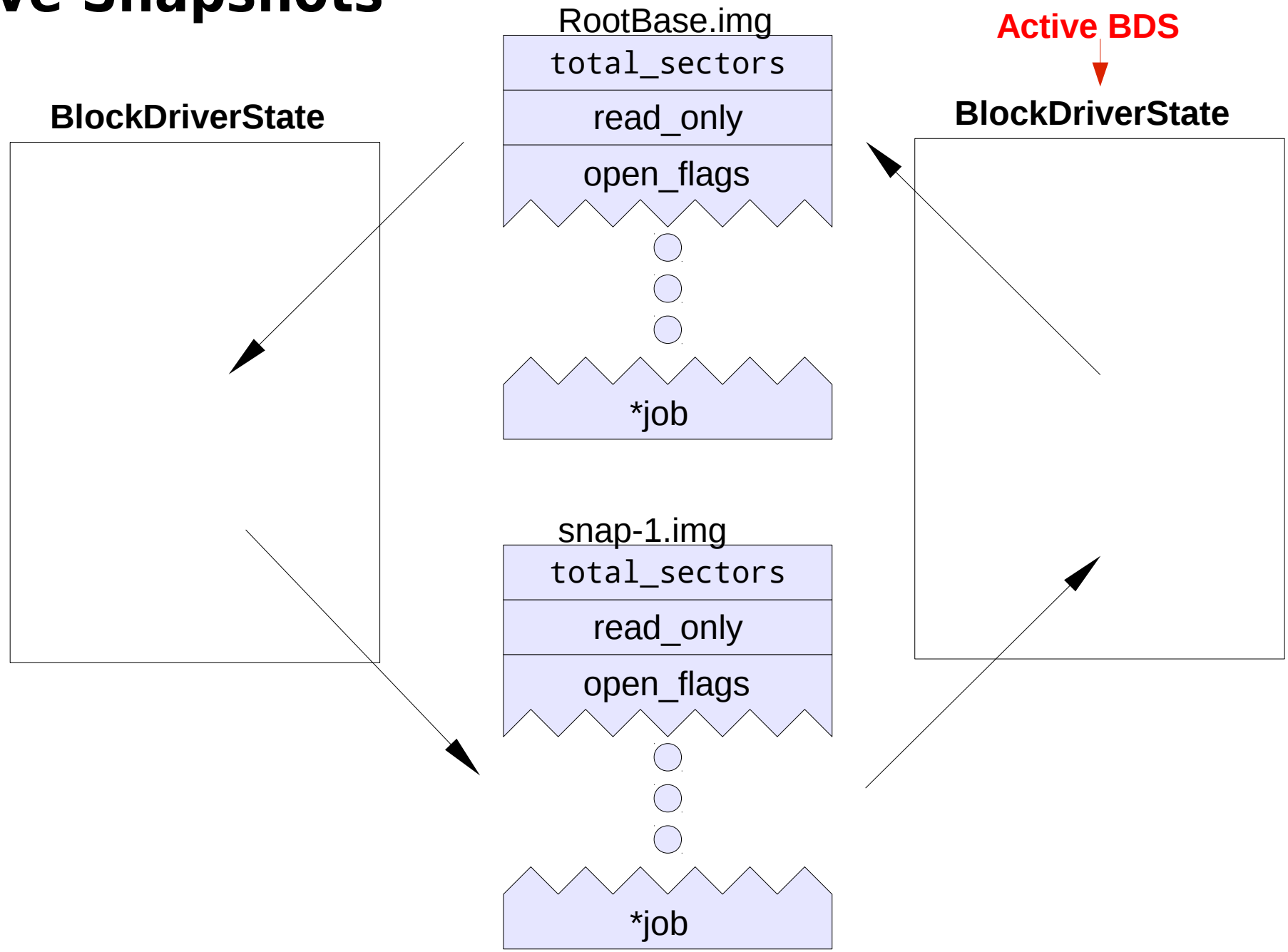



```
{ "execute": "blockdev-snapshot-sync", "arguments":
   { "device": "virtio0","snapshot-file":"snap-1.img"} }
```

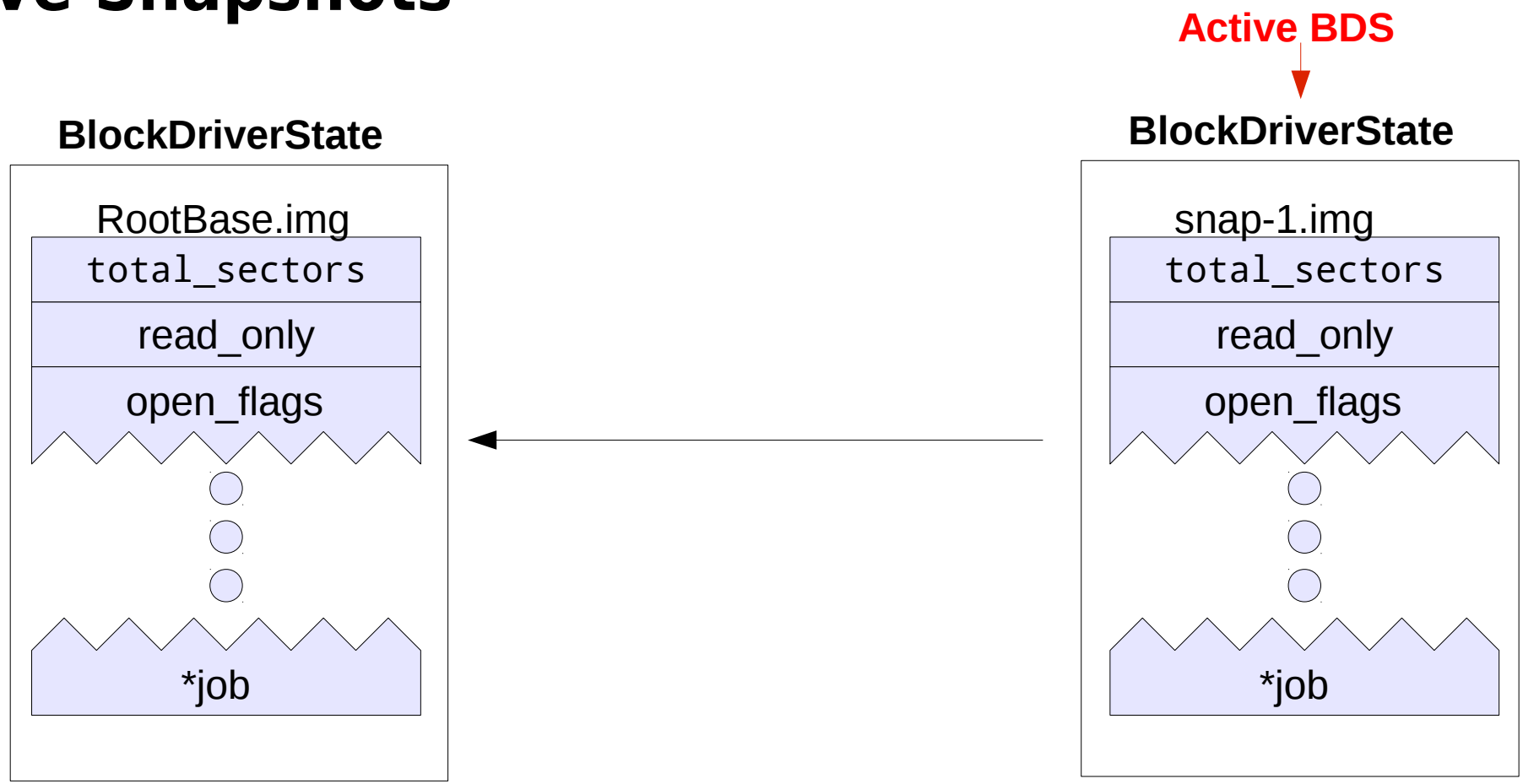
Live Snapshots



Live Snapshots



Live Snapshots



Live Snapshot – multiple devices

- Safe and atomic
- Each image created before live QEMU image chain is modified
- If any image file creation fails, operation is abandoned without touch the image chain
- On failure, will leave 'mouse droppings'

Live Merge

Live Merge

There is no “Live Merge” command!

Instead, we have two commands:

- block-stream
- block-commit

Block Stream and Block Commit

- Asynchronous; run as a block job while guest is live.
- Issues BLOCK_JOB_COMPLETED event on completion, with type 'stream' and 'commit'.

Block Commit and Stream

QAPI Commands:

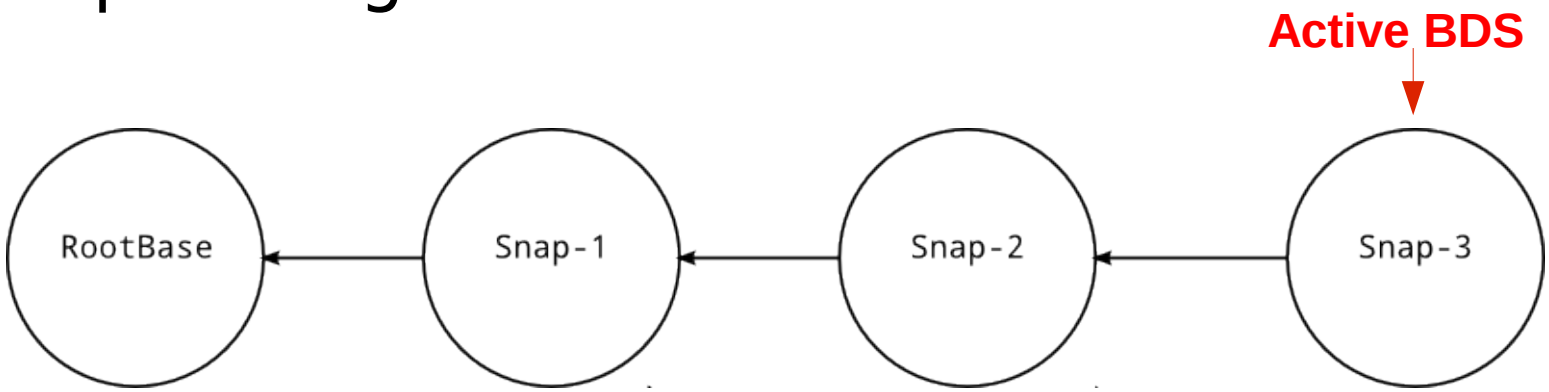
```
{ 'command': 'block-commit',
  'data': { 'device': 'str',
            '*base': 'str',
            'top': 'str',
            '*speed': 'int' } }


{ 'command': 'block-stream',
  'data': { 'device': 'str',
            '*base': 'str',
            '*speed': 'int',
            '*on-error': 'BlockdevOnError' } }
```

Block Stream and Block Commit Differences

block-stream	block-commit
• Merges towards active layer	• Merges towards base
• Intermediate images remain valid	• Intermediate images become invalid
• Merges to the active layer only	• Can commit between any intermediate images below the active layer.
• Since v1.1	• Since v1.3

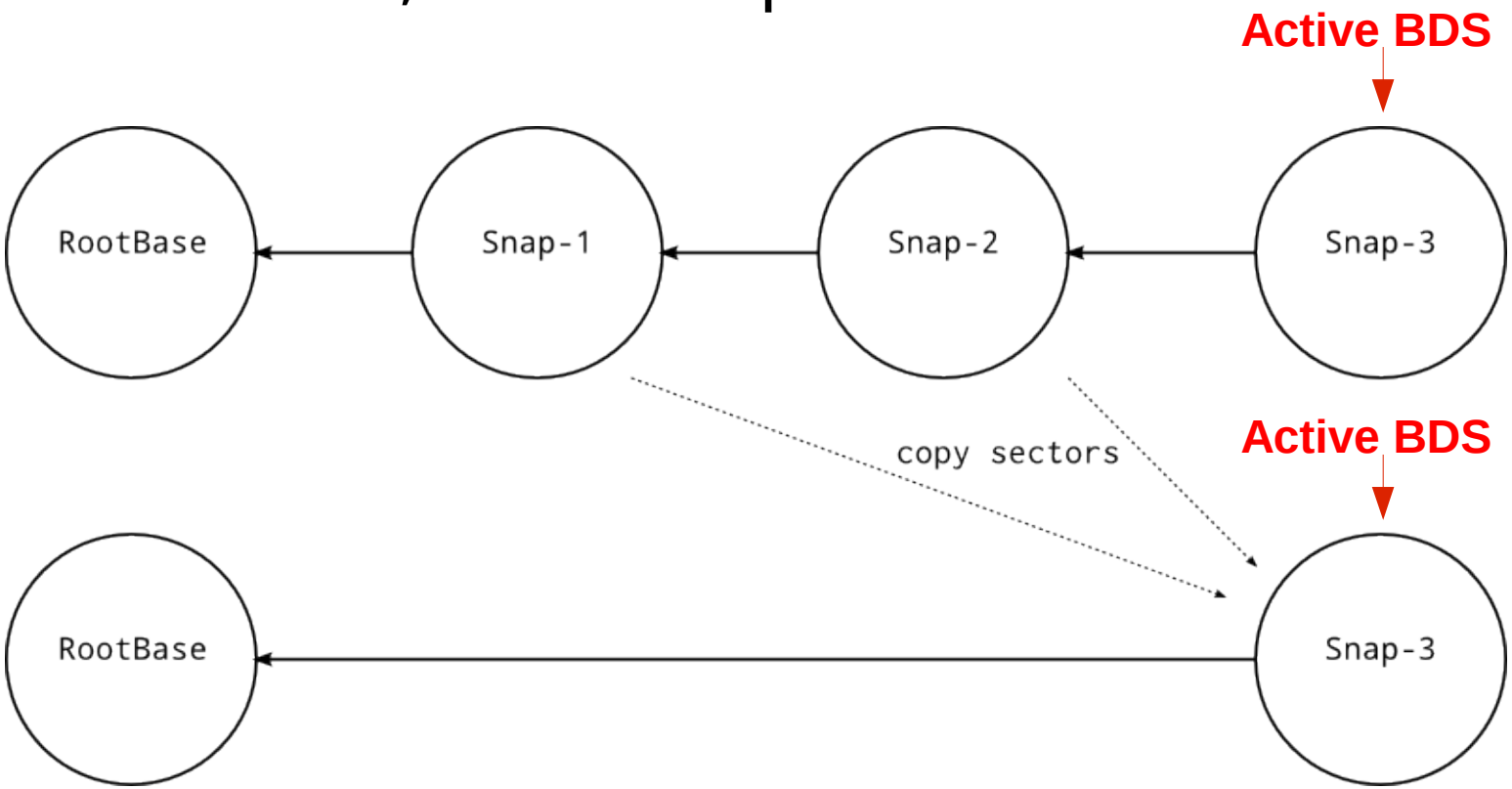
Block Stream

Sample image chain:



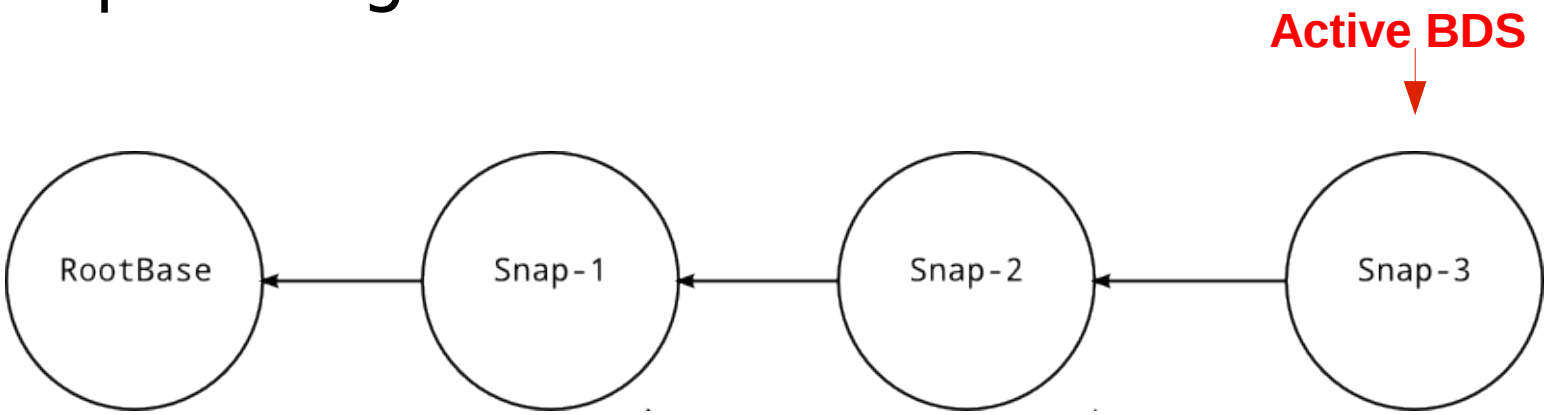
Block Stream

Block-stream, from Snap-1:



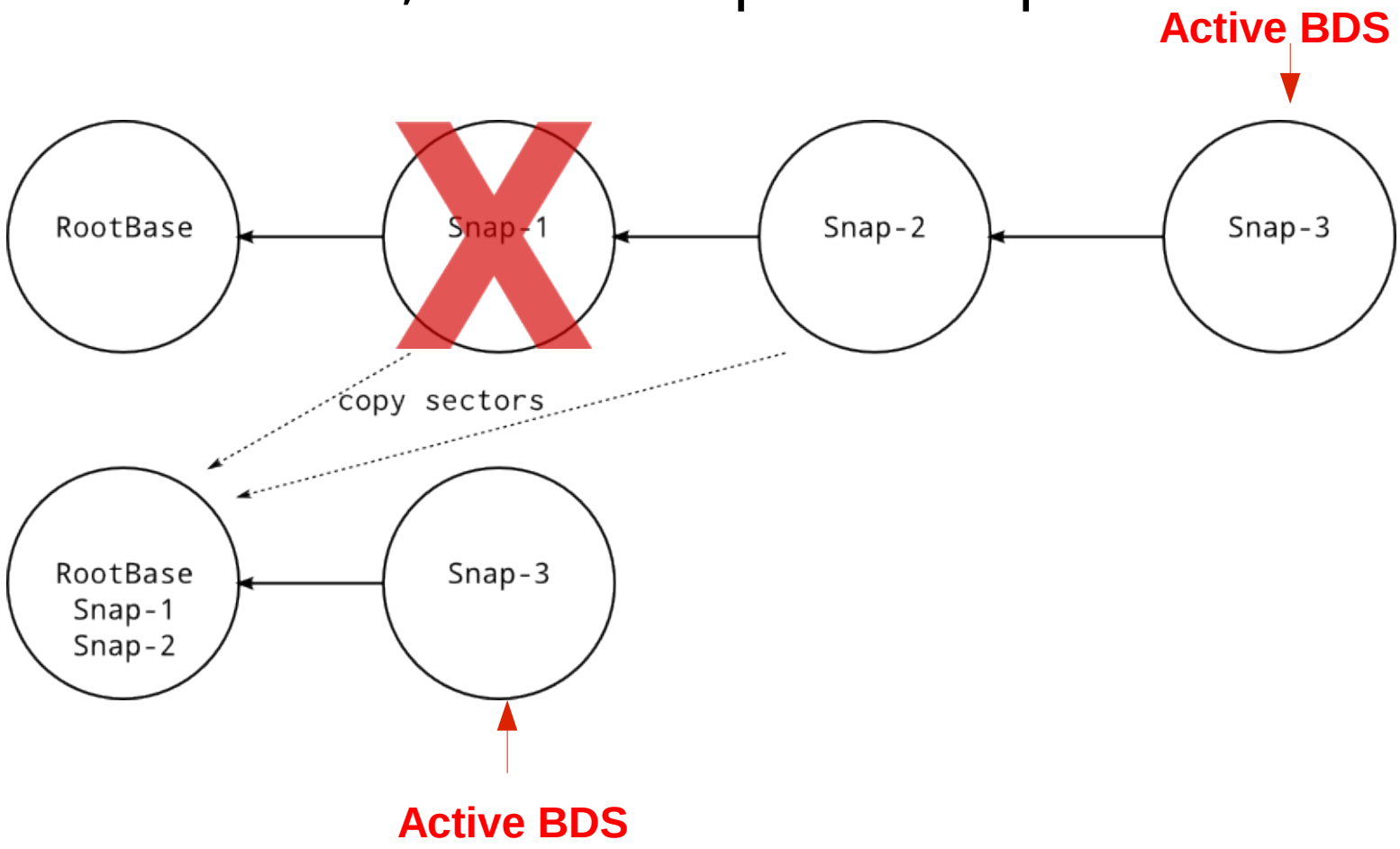
Block Commit

Sample image chain:

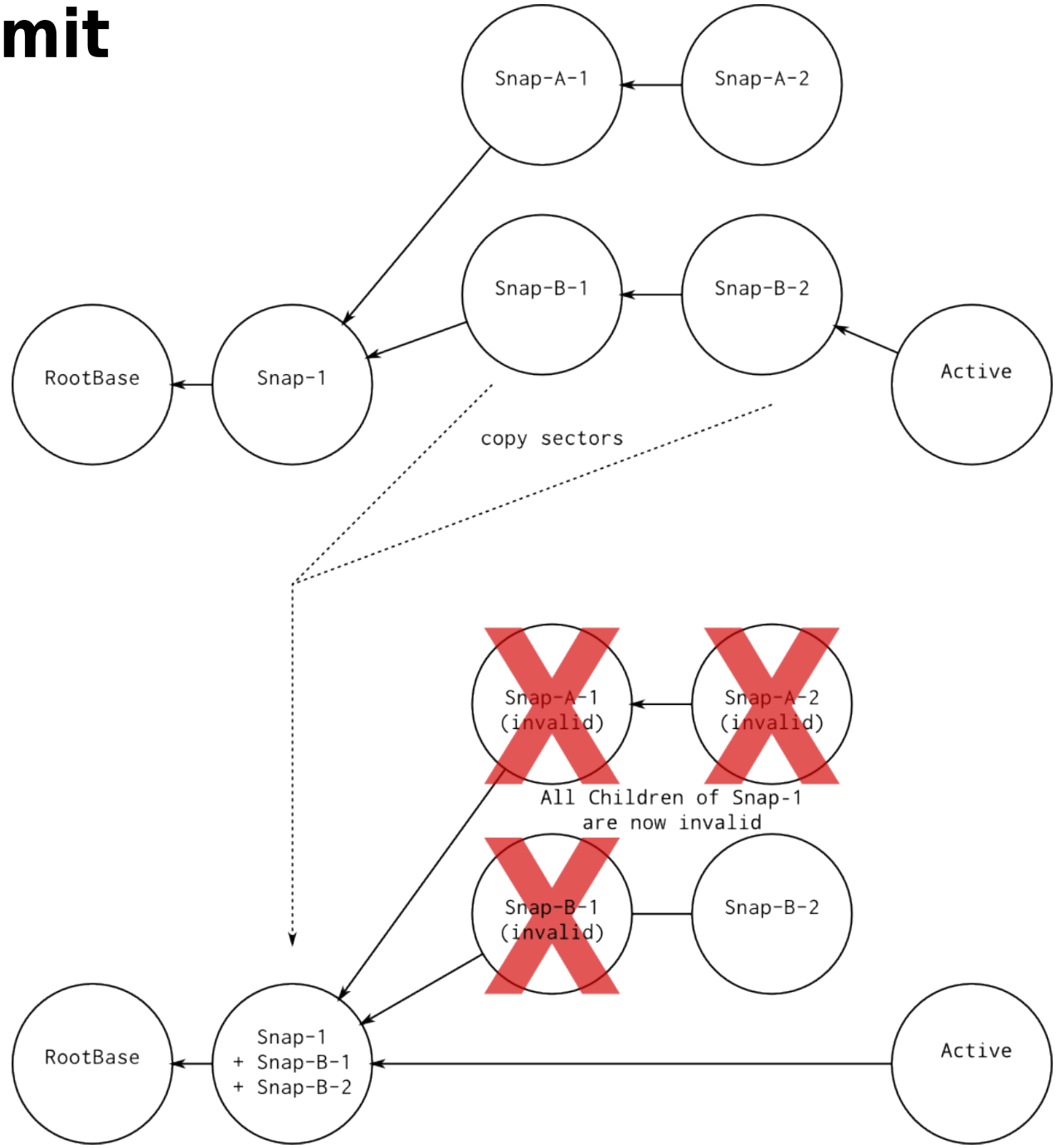


Block Commit

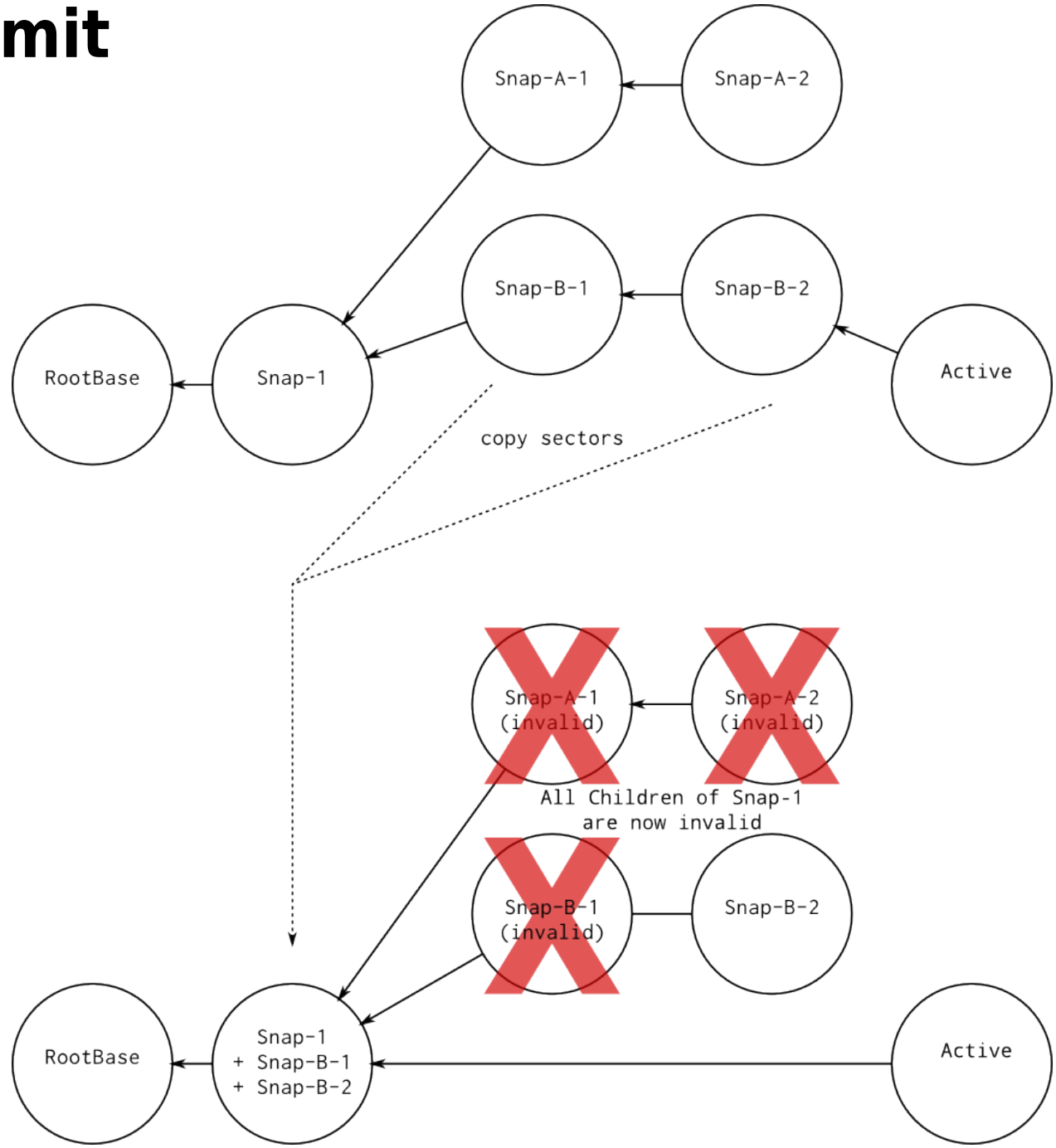
Block-commit, from Snap-2 as top:



Block Commit



Block Commit



Block Commit – Why use it?

- Potentially faster
  - Snapshots likely smaller than backing image
- Can collapse into RAW backing file

Block Commit – What happens

- Block job created
- Sectors copied in block job to 'base', above 'base' through 'top'.
- Backing file updated in the overlay of 'top'
- Intermediate images dropped from chain

Block Stream – What happens

- Block job created
- Sectors copied in block job to active layer, between 'base' and active layer.
- Backing file updated in active layer
- Unused images closed

Block Commit – What's Next

- Commit of active layer
  - Guest still writing to image
- Commit intermediate images in order

Block Stream – What's Next

- Stream to intermediate images, not just active layer.

Drive Mirror

Drive Mirror

- Mirrors the writes of a block device to new target
- 3 Sync Modes
  - Top
    - Copies the topmost image data to target, plus all new writes
  - Full
    - Copies all image data in the chain to target, plus all new writes
  - None
    - Copies only new writes to target

```
{ 'command': 'drive-mirror',
  'data': { 'device': 'str', 'target': 'str', '*format': 'str',
            'sync': 'MirrorSyncMode', '*mode': 'NewImageMode',
            '*speed': 'int', '*on-source-error': 'BlockdevOnError',
            '*on-target-error': 'BlockdevOnError' } }
```

Drive Mirror

- Uses dirty bitmap to know which sectors to copy over
- If copying 'static' data (Top and Full sync modes), dirty bitmap is first initialized.

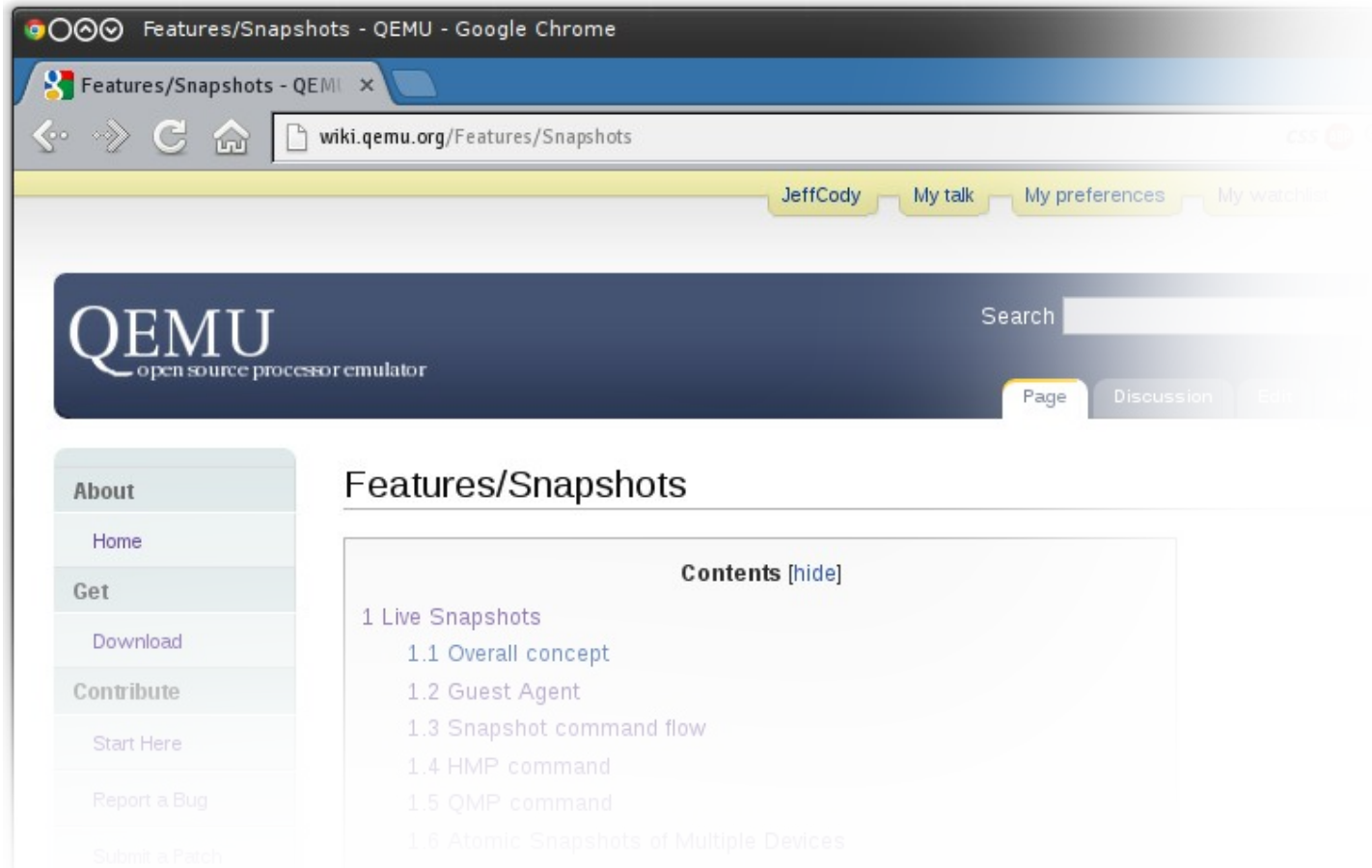
Any Questions?

"Judge a man by his questions rather than his answers"

- Voltaire*

More information:

http://wiki.qemu.org/Features/Snapshots



IRC: jtc on OFTC (#qemu)
Email: jcody@redhat.com

Any Questions?

"Judge a man by his questions rather than his answers"

- Voltaire*

* OK, that is actually an incorrect quote – the real quote is:

It is easier to judge the mind of a man
by his questions rather than his answers

- Pierre-Marc-Gaston