Job Safety:
Blockers in the Block Layer

Jeff Cody | jcody@redhat.com
Düsseldorf, Germany
October 15, 2014
QEMU Block Jobs
QEMU Block Jobs

- Started via QMP or HMP
  - block-commit
  - block-stream
  - etc...
- Runs in a coroutine, while guest runs
  - Asynchronous
  - Modifies block image(s) in some way
Why are Blockers Needed?

- bs->job already prohibits multiple jobs
- Does not just block other block jobs

- Blocks some synchronous commands
  - eject
  - hmp commit
  - drive-del
  - etc..
History of Block Job Safety
Job Safety, circa v2.0.0

- `bs->in_use`
  - v0.15.0 up to v2.1.0
    - Boolean
      - Blocks whole BDS chain
  - Only single BDS was checked, at the top layer.
What Changed?

- Direct BDS access
  - May not reference active (top) BDS
  - Enabled by node-names
Job Safety, circa v2.1.0

- in_use flag removed
- Backing blockers added
  - Command-specific
    - BLOCK_OP_TYPE_RESIZE
    - BLOCK_OP_TYPE_STREAM
    - etc..
    - Applied automatically to backing_hd
    - Checked by various block cmds
- Improvement over in_use flag
  - Provides some granularity
  - Blockers have individual BDS in mind
Current Issues

- Still just checked on the top-level BDS
- BLOCK_OPT_TYPE_COMMIT?
  - Special exception in bdrv_set_backing_hd()
    - Means everything can only check active BDS for block-commit

```c
bdrv_op_block_all(bs->backing_hd, bs->backing_blocker);
/* Otherwise we won't be able to commit due to check in bdrv_commit */
bdrv_op_unblock(bs->backing_hd, BLOCK_OP_TYPE_COMMIT,
                 bs->backing_blocker);
```
Current Issues

- Not set or checked recursively
  - Many block jobs operate on multiple nodes
    - "Honor system"
Current Issues

- Blockers by QAPI Command
  - Per-command blockers not used yet; all-or-none
  - How to determine what blockers to check?
    - Forces QAPI handlers to be aware of how every other handler implemented
      - Does STREAM need to check for COMMIT, etc.
Current On-list Work

Patches from Benoît Canet

- Adds recursion to blockers

- We can use blockers by command BLOCK_OP_TYPE_COMMIT, yay!
- Protects chain
  - Can remove bs->job
- Still uses blockers typed by QAPI command
QEMU 2.3+
Set blocker by action

- On each BDS:
  - Read / Write
  - Graph Modify
  - Attributes

- Blockers obtained atomically
  - Set blocker on multiple BDSs
    - Check if blockers currently set
      - All-or-none
Examining block-commit

active

snap-A

snap-B

snap-C

base
Examining block-commit

snap-A into snap-C
Examining block-commit

block Reads + Writes

active

snap-A

snap-B

snap-C

base
Examining block-commit

block Write

active

snap-A

snap-B

snap-C

base
Examining block-commit

block Attribute / Metadata

active

snap-A

snap-B

snap-C

base
Examining block-commit

block Chain Manipulation

- active
  - snap-A
    - snap-B
      - snap-C
        - base
Block by action

- Issues:
  - Could be more granular
  - Need to evaluate for each node in graph
Block by action

- Advantages:
  - Only need to know self
    - e.g., block-commit only cares about itself
  - Provides more granularity than current
    - Potentially support more concurrent block jobs
More Enhancements

- Allow modifiers to action blockers
  - Applies more definition
    - e.g., GUEST_VISIBLE
  - Can be extended
The Future is Hazy...

- Define blockers in the QAPI JSON

  Relevant blockers set automatically
  - Don't need to write redundant code
  - Fewer mechanical mistakes
Questions / Discussion
THE END

Jeff Cody | Red Hat, Inc.