Shared Snapshots

Mikuláš Patočka

Red Hat Czech, s.r.o.
Snapshots

– Snapshot is a fixed image of a volume taken at a specific time
– Most common uses:
  – Online backup
  – Preserving data before change
  – Large sparse device (snapshot of a zero volume)
Two approaches to snapshots

In logical volume manager (LVM2)
  + Works with any filesystem (or raw volume)
  - Less space efficient, tries to preserve even unallocated blocks

In filesystem driver (OpenVMS Spiralog, FreeBSD FFS, Solaris ZFS, Linux Btrfs)
  - Requires filesystem support
  + More space efficient
  - May fragment the filesystem
LVM2 snapshots

- Separate logical volume for each snapshot
- On write to the origin, copies previous data to every snapshot
- Performance degradation with multiple snapshots
Shared snapshots

- One volume holding all the snapshots
- Filesystem-like structure inside
- Snapshots share common blocks
- Efficient with many snapshots
Possible uses

– Monitoring system activity
  – Take snapshot every few minutes
  – Record system activity
– Multiple volumes with most common blocks
  – Images of virtual machines
– Thin-provisioning
  – Allocate space on demand
– Snapshots-of-snapshots are supported
  (but snapshots-of-snapshots-of-snapshots aren’t)
Implementation

– Snapshot store keeps data in the units of chunks. Chunk size is configurable.
– 64-bit ID. High 32 bits are snapshots ID, low 32 bits are subsnapshot ID.
– New snapshots get growing IDs. IDs are never reused.
– B+tree keyed by (block number, starting+ending ID)
– Log-structured format for crash recovery
B+tree

- Chunk number in the origin
- Starting snapshot ID
- Ending snapshot ID
- Chunk number in the snapshot store
- Flags

Leaf B+tree entry

B+tree key
Writing to the origin

- Snapshot ranges in B+tree
- Snapshot ranges in memory
- Ranges that must be copied when writing to the origin

An unused entry
It may exist for some time but background scan will eventually delete it

If the entry has DM_MULTISNAP_PREVIOUS_COVERED flag, it is guaranteed that all previous ranges are reallocated

- Snapshot ranges in B+tree
- Snapshot ranges in memory
- Ranges that will be copied when writing to the origin
Using

- Create the shared snapshot store
  lvcreate -s --sharedstore mikulas -c 64k
  -L 64G vg/volume

- Create individual snapshots
  lvcreate -s -n snap vg/volume

- Create snapshot of snapshot
  lvcreate -s -n snapofsnap vg/snap
How does it look

# lvs

LV   VG  Attr   LSize  Origin   Snap%
snap vg  swi-a-  48,00g  volume
snapofsnap vg  swi-a-  48,00g  volume
volume vg  owi-a-  48,00g
[volume-shared] vg  swi---  64,00g  volume 0,00

lvresize – Resize the individual snapshots or the whole store
lvconvert --merge – Merge the snapshot to the origin
lvremove – Remove the snapshot or the store
Sequential write rate depending on a number of snapshots
Random write rate depending on a number of snapshots
Sequential write rate depending on a chunk size
Random write rate depending on a chunk size
Where to get it

http://people.redhat.com/msnitzer/patches/multisnap/lvm2/

http://people.redhat.com/mpatocka/patches/kernel/new-snapshots/

http://people.redhat.com/mpatocka/patches/userspace/new-snapshots/

Ask me: mpatocka@redhat.com