



BACKING UP MYSQL USING FILE SYSTEM SNAPSHOTS

Lenz Grimmer

Community Relations Manager EMEA
Sun Microsystems

Overview

- File system snapshots provide a complete, physical copy of the actual MySQL database/table files
- Use cases:
 - > Full backups
 - > Disaster recovery
 - > Setting up database slaves



Picture taken by Travis Gray (flickr)

General Procedure

- FLUSH TABLES
- FLUSH TABLES WITH READ LOCK
- Create the snapshot
- SHOW MASTER/SLAVE STATUS
- UNLOCK TABLES
- Mount snapshot, perform backup
- Unmount/Discard the snapshot

Benefits of MySQL Snapshot Backups

- “Almost hot” (no downtime)
- Supports all storage engines
- Fast, low overhead
- Easy integration
- Can be combined with log recovery
- Fast recovery
- (Usually) Free

Snapshot Backup Caveats

- Not incremental
- InnoDB ignores FLUSH TABLES WITH READ LOCK
- FLUSH TABLES performance impact
- Possible I/O performance impact while snapshot is active (Linux LVM)
- Handling data spread on multiple volumes (DB logs on separate LV or DBs spread across multiple LVs)

Workarounds / Tricks

- “Clean” InnoDB snapshots
 - > Shut down the server
 - > Use a replication slave
 - > Start a second MySQL instance on the snapshot to perform the log recovery
- Logical backups: start a second MySQL instance on the snapshot and use mysqldump

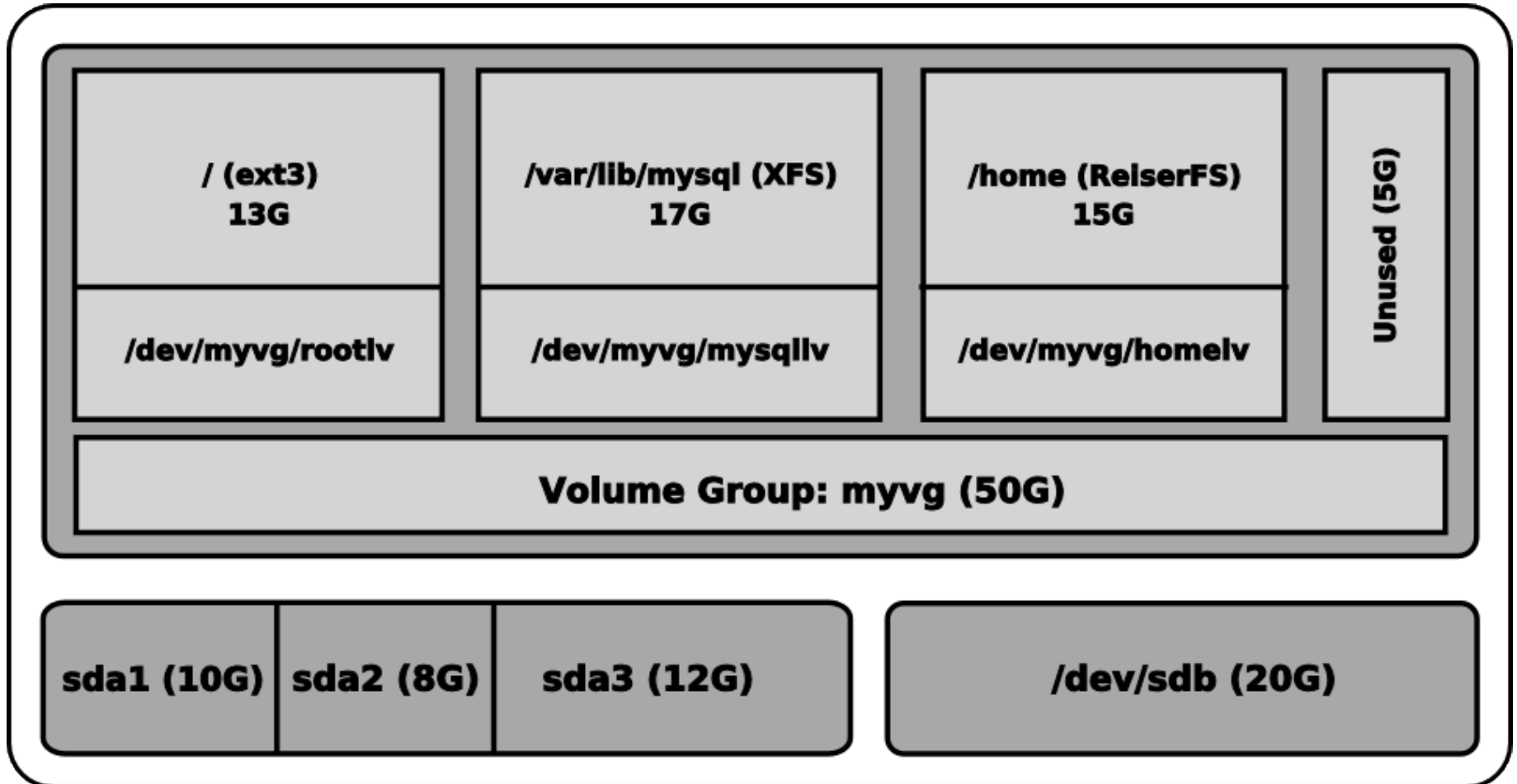
Linux LVM

- Storage Virtualization / Abstraction for Linux
- Initially developed by Sistina (now Red Hat)
- More flexible than partitioned disks:
 - > Online adding/removing of volumes
 - > Online resizing/moving of volumes
 - > Grouping of volumes
- Beneficial for both large and small storage systems

Terminology

- Logical Volume (LV)
- Logical Extent (LE)
- Volume Group (VG)
- Physical Volume (PV)
- Physical Extent (PE)

Overview



Tools

- Commandline
 - > PVs: pvcreate, pvdisplay, pvmove, pvresize ...
 - > VGs: vgcreate, vgdisplay, vgextend, vgreduce ...
 - > LVs: lcreate, lvdisplay, lvextend, lvreduce...
- YaST2 (openSUSE/SLES)
- Disk Setup & system-config-lvm (Fedora/RHEL)
- IBM EVMS tools (GUI, CLI)
- LVM GUI (Java)

Linux LVM Snapshots

- Atomic, instant & exact copy of another LV
- Low disk space requirements (COW)
- LVM2 provides read & write access on snapshots
 - > Useful for testing purposes (e.g. software updates)
 - > Or cloning Xen DomU instances
 - > Or starting another MySQL instance

LVM Snapshot Caveats

- Requires data to actually reside on a LV
- Needs free space in the VG for the backing store
- Active snapshots degrade I/O performance
- <http://tinyurl.com/lvm-performance>

mylvmbackup

- <http://www.lenzg.net/mylvmbackup/>
- Perl script
- GPLv2
- Automates the process of creating MySQL snapshots and taking backups
- Creates tar.gz archives or directory copies (using rsync/rsnap)
- Can perform InnoDB log recovery on the snapshot prior to backup (LVM2)

LVM Hints & Recommendations

- File system recommendations
 - > XFS, ReiserFS (support online resizing of LVs)
- Snapshot size considerations
 - > Snapshots are automatically discarded when they exceed the size given at creation time
 - > Check the output of `lv`s after taking the backup to determine the percentage of space used in the backing store
- Discard snapshots after successful backup

ZFS

- 128bit File System
- Solaris/OpenSolaris, FreeBSD, Linux (zfs-fuse), Mac OS X
- Simple administration
- Pooled storage (no partitions/volumes)
- Copy-on-write transactions

ZFS (2)

- Checksums, self-healing (no silent data corruption)
- Striping / mirroring / RAID / Compression
- ZFS Volumes (iSCSI)
- CIFS / NFS

ZFS Snapshots

- Read-only, point-in-time copy of the filesystem
- Instantaneous creation
- (Virtually) unlimited number of snapshots
- Initially, no additional space used
- Writable copies (Clones)
- Incremental replication (zfs send/receive)
- Snapshots are simple & cheap to create!
`zfs snapshot fsname@snapname`

ZFS Hints & Recommendations

- Set ZFS record size to match InnoDB block size (before creating the DB):
`zfs set recordsize=16K <filesystem>`
- Avoid double-buffering: set InnoDB buffer pool size to a low value
- Consider disabling file-level prefetching
- Consider enabling compression

<http://dev.mysql.com/tech-resources/articles/mysql-zfs.html>

http://www.solarisinternals.com/wiki/index.php/ZFS_Evil_Tuning_Guide

<http://en.oreilly.com/mysql2009/public/schedule/detail/7121>

Zumastor

- <http://zumastor.org/>
- Better snapshots and remote replication (asynchronous)
- Requires Linux LVM
- One snapshot store for all snapshots
- Up to 64 snapshots per volume
- GPLv2

btrfs

- <http://btrfs.wiki.kernel.org/>
- A new COW file system for Linux
- Supports writable snapshots
- Actively developed by Oracle (Chris Mason), other Vendors (HP, IBM) and Linux distributors (SuSE/Novell, Red Hat)
- GPLv2
- Work in progress (on-disk format changes), not ready for production yet (2009-02)

Zmanda Recovery Manager (ZRM)

- <http://www.zmanda.com/backup-mysql.html>
- Commercial backup solution with an Open Source version (no GUI, limited functionality)
- Supports various snapshotting technologies
- Supports various backup techniques

R1Soft Linux Hot Copy

- <http://www.r1soft.com/products/linux-hot-copy/>
- Command line utility (hcp) and kernel module (block device driver)
- Creates read-/writable snapshots of any Linux Block device using COW
- No dedicated snapshot device or storage: changed blocks are stored in free space of the block device
- Does not require LVM
- **Not** Open Source (Freeware)

Questions / Comments?



BACKING UP MYSQL USING FILE SYSTEM SNAPSHOTS

Lenz Grimmer

Community Relations Manager EMEA
Sun Microsystems