

MySQL University

MySQL and ZFS

Martin MC Brown
Technical Writer
Database and Infrastructure Group,
Sun Microsystems

Overview

- ZFS Overview
- Using MySQL on ZFS
- Increasing Performance
- ZFS Tricks

Understanding ZFS

- Filesystem/volume management in software
- Makes use of faster CPUs often sitting idle
- Flexible support compared to hardware RAID
- Enhanced range of features
- 128-bit throughput
- Integrated fault management, check-summing and copy-on-write

Pools

- Logical collection of physical disks
- Everything in software
- Correctly sets disk metrics, caches
- Transparent use of SSDs
- Transparent use of different disks

Pool Types: Stripe (RAID 0)

- Default pool type
- `zpool create MYPOOL c0t0d0`
- `zpool create MYPOOL c0t1d0 c0t2d0 c0t3d0`
- Extend the pool:
 - > `zpool add MYPOOL c0t5d0`

Mirrored (RAID 1)

- Mirrored (writes to N+1 disks)
- `zpool create MYPOOL mirror c0t1d0 c0t2d0`
- To extend a mirror, you must add another mirror set:
 - > `zpool add MYPOOL mirror c0t3d0 c0t4d0`

Striped and Mirrored (RAID 0+1)

- Create a mirror/stripe one go:
 - > `zpool create MYPOOL mirror c0t1d0 c0t2d0 mirror c0t3d0 c0t4d0`

Striped with parity, RAIDZ (RAID5)

- RAIDZ
 - > Like RAID5, but more flexible
 - > Easy to extend
 - > 1 or 2 disk parity
- 1 parity disk (2 disks or more)
 - > `zpool create MYPOOL raidz c0t0d0 c0t1d0`
- 2 parity disks (3 disks or more)
 - > `zpool create MYPOOL raidz2 c0t0d0 c0t1d0 c0t2d0`

RAIDZ Extension

- Extension possible
- Can't use a single disk
- Extend using same geometry:
 - > `zpool add MYPOOL raidz c0t2d0 c0t3d0`
 - > `zpool add MYPOOL raidz2 c0t3d0 c0t4d0 c0t5d0`

Adding a hot spare

- Hot spares can be used if a disk fails
- Not used until a failure is located
- But employed automatically when it does
- Best employed with a mirror or RAIDZ
- Add one or more spares
 - > `zpool add MYPOOL spare c0t4d0`

Filesystems

- One Filesystem can have only One Pool
- But One Pool can have multiple filesystems
- Filesystem created automatically when you create pool
- All filesystems share the storage of the pool
- Each filesystem supports individual settings, even if they share the same pool

ZFS Intent Log

- ZFS Intent Log supports synchronous transaction support
- Designed to be used to support immediate writes/`fsync()`
- Allocate the log to different device for speed:
 - > `zpool create MYPOOL c0t1d0 c0t2d0 log c0t3d0`
- Specify multiple devices
- Can be mirrored
- Great for 10K/15K RPM disks or SSD

ZFS Cache

- Caches provide extra layer between RAM and Disk for read-only ops
- ZFS supports exclusive device for cache:
 - > `zpool create MYPOOL c0t1d0 c0t2d0 cache c0t3d0`
- Specify multiple devices to increase cache size
- Cannot be mirrored/striped
- Cache disk is considered volatile
- Great for 10K/15K RPM disks or SSD

MySQL: Reducing Admin

- Running out of space?
- Moving data is a pain point
- InnoDB:
 - > Old method:
 - > `innodb_data_file_path = /disk1/ibdata1:10G;/disk2/ibdata2:10G;/disk3/ibdata3:10G:autoextend`
 - > New method:
 - > `innodb_data_file_path = /dbzpool/data/ibdatafile:10G:autoextend`

MySQL: Increasing Performance

- On InnoDB switch off double-write buffer
- Match the ZFS block size and InnoDB block size for data
 - > zfs set recordsize=8K MYPPOOL
- Use a separate zpool for log files
- To increase RAM for MySQL, limit ARC:
 - > set zfs:zfs_arc_max = #bytes in /etc/system

Compression

- Handled in software
- Filesystem level
- Sounds Bad
- However, using compression:
 - > Reduces read and write times
 - > More systems have idle CPU but high disk contention
 - > Takes up less space

Using Multiple ZFS Filesystems

- Provide greater control
- Create a top-level pool for a MySQL database
- Create a ZFS filesystem for each database
- Set compression/quotas on each

ZFS Snapshots

- Backups are a pain point
- Snapshots are quick and efficient
 - > `zfs snapshot MYPOOL@snap1`
- Save a snapshot to a file or tape:
 - > `zfs send MYPOOL@snap1 >mypool.backup`
- Or over a network:
 - > `zfs send MYPOOL@snap1 | ssh mybackuphost cat ->snapshot`
- Restore into a new pool:

ZFS Replication

- Not really replication
- Regular snapshots/restores between systems
- See coalface.mcslp.com—zfs-replication-for-mysql-data <<http://coalface.mcslp.com/2008/11/09/zfs-replication-for-mysql-data/>>

Questions?

Martin MC Brown

mc.brown@sun.com