ZFS – SYNC vs. ASYNC I/O

Robert Milkowski
Senior Systems Analyst
TalkTalk Group

http://milek.blogspot.com

ZFS Synchronous vs. Asynchronous I/O

ZFS I/O

- Asynchronous
  - I/Os are buffered in memory for 5-30s
- Synchronous
  - fsync(), O_DSYNC, O_SYNC, ...
  - Written immediatelly to ZIL
    - slog or inside a pool
  - NFS does synchronous I/O for all meta-data operations

How To Override

- Sometimes it is useful to override sync/async
  - zil_disable (all I/O to ZFS turns into ASYNC)
- New zfs sync property
  - On-the-fly changes with immediate effect
  - Applies both to ZFS datasets and zvols
    - Overrides zvol's WCE flag
    - dataset/zvol granularity
  - Inheritable

SYNC Property Syntax

- sync=standard

  This is the default option. POSIX compliant behaviour.

- sync=disabled

  Synchronous requests are disabled.

- sync=always

  Every file system transaction is written and flushed

  to stable storage by a system call return.

ZFS Synchronous vs. Asynchronous I/O

Usage Example

```
# zfs create rpool/test
# zfs get sync rpool/test
NAME        PROPERTY  VALUE     SOURCE
rpool/test  sync      standard  default

# ptime ./sync_file_create_loop /rpool/test/f1 1000
real       11.284050371

# zfs set sync=disabled rpool/test
# zfs get sync rpool/test
NAME        PROPERTY  VALUE     SOURCE
rpool/test  sync      disabled  local

# ptime ./sync_file_create_loop /rpool/test/f1 1000
real        0.041377999
```

280x improvement!

On-Disk Consistency

- sync=disabled
  - Does **NOT** affect ZFS on-disk consistency
  - Might affect data consistency from an application p.o.v. (only if OS reboots/crashes without syncing data)
  - All I/O is commited when a TXG commits
    Currently between 5-30s by default
  - All I/O is commited in the same order as submitted
  - sync(1M) still sync's all filesystems before it returns

ZFS Synchronous vs. Asynchronous I/O

NFS Server

- Synchronous I/O
  - NFSv3 WRITE with FILE_SYNC or DATA_SYNC flag set
  - COMMIT operation (also commit on close by default)
  - NFSv3 server does synchronous I/O for meta-data ops

    SETATTR, CREATE, MKDIR, SYMLINK, MKNOD
    REMOVE, RMDIR, RENAME, LINK
- sync=disable
  - Data might be corrupted from an application p.o.v.
    if server crashed, no impact on zfs on-disk consistency

Integration Details

- PSARC/2010/108 zil synchronicity
  Platform Software Architecture Review Committee
  - Bug id: 6280630
  - zil_disable removed
- Integrated into snv_140
  - No, it won't make it into OpenSolaris 2010.06

Contributing to Open Solaris

- Write code, test it, cstyle, webrev, submit
- Sun Contributor Agreement
  - Gives Sun and a contributor joint copyright
  - http://hub.opensolaris.org/bin/view/Main/sun_contributor_agreement
- Request a sponsor

  e-mail to request-sponsor@opensolaris.org

  - Code reviews, testing, doc changes, ...
  - PSARC, RTI, ...

Webrev

- HTML-based code reviews
  - Cdiff, Udiff, Wdiff, Sdiff, PDF, patch, ...
- http://cr.opensolaris.org
  - Allows to publish and share webrevs

```
webrev -U -o onnv.6280630.14
```

- Delivered in developer/build/onbld package

```
pkg install onbld
```

Development Environment

- VirtualBox
  - Good for kernel panic's
  - VM snapshots
  - Limits any harm to a VM only
- Open Solaris Boot Environments
  - Quick&Easy software updates
  - Fast reboots into a BE

ZFS Synchronous vs. Asynchronous I/O

Useful links

http://milek.blogspot.com/2010/05/zfs-synchronous-vs-asynchronous-io.html
http://arc.opensolaris.org/caselog/PSARC/2010/108/
http://bugs.opensolaris.org/bugdatabase/view_bug.do?bug_id=6280630

http://milek.blogspot.com/2010/02/zvols-write-cache.html

http://blogs.sun.com/roch/entry/nfs_and_zfs_a_fine

http://hub.opensolaris.org/bin/view/Main/sun_contributor_agreement

http://opensolaris.org/os/community/zfs/

ZFS Synchronous vs. Asynchronous I/O

ZFS – SYNC vs. ASYNC I/O

Robert Milkowski
Senior Systems Analyst
TalkTalk Group

http://milek.blogspot.com