Hybrid Filesystems Revisited

A tutorial on hybrid SSD/HDD XFS filesystems

Peter Edwards | Sr Systems Administrator

DMF Users Group| February 2015

CSIRO IM&T SCIENTIFIC COMPUTING
www.csiro.au

CSIRO

Hybrid Filesystems

Different types of media are appropriate for different uses:

- Solid-state drive (SSD) media is appropriate for small latency-sensitive operations
- Rotating hard-disk drive (HDD) media is appropriate for larger bandwidth- and capacity-intensive operations

A hybrid filesystem contains both types of media to accommodate both sets of requirements. Directories, most metadata (in particular, inodes) and logs (aka journals) are SSD-resident, and data blocks for files are mostly on HDD.

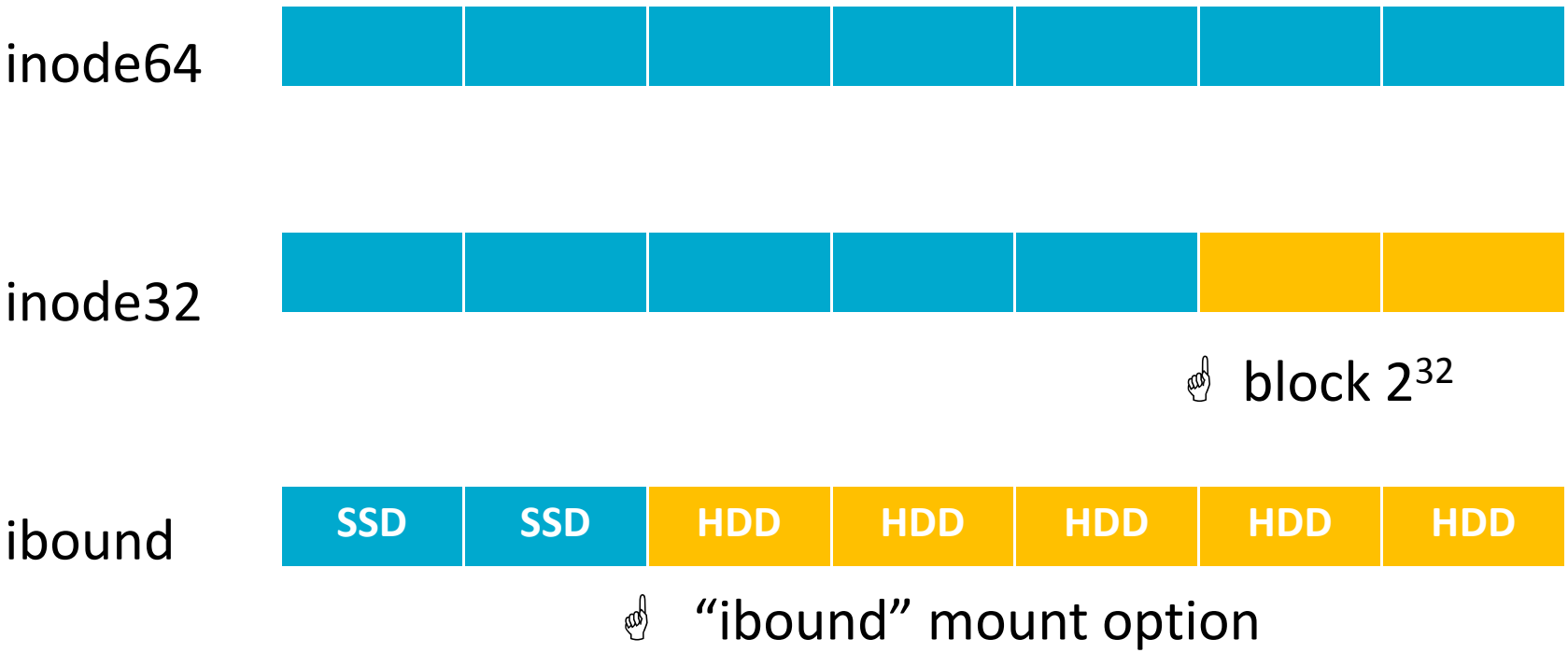
This presentation discusses their implementation in SGI's eXFS filesystem.

Types of Activity

- Inode intensive - XFS_IOC_FSBULKSTAT (“bulkstat”) ioctl
  - dmmigrate, dmfsfree, non-recursive dmscanfs, xfsdump –a
- Directory intensive
  - rsync, find, dmscanfs –r
- Log intensive
  - rm, dmput –r
- Data intensive

(Not mutually exclusive)

XFS Allocation Groups



Directories and most metadata are confined to the blue AG's.

External logs are outside this "data subvol".

Inode64 has a different allocation strategy from the others; not discussed here.

A simple XVM hybrid volume

```
cherax#  xvm  show  -top  vol/is55b_backup1
vol/is55b_backup1                                    0  online,open,accessible
    subvol/is55b_backup1/data                 8657010688  online,open,accessible
        concat/concat0                        8657010688  online,open,accessible
            slice/is4600_ssd_meta94s0           67092480  online,open,accessible
            slice/is5500b_sas_23s0            8589918208  online,open,accessible
     subvol/is55b_backup1/log                      245760  online,open,accessible
        slice/is4600_ssd_log75s0                   245760  online,open,accessible
```

Note the external log (journal) partition – much better than internal log.

Maximum log size is 2038 MiB – wasteful if on HDD but easy on SSD.

A more complex one

```
vol/datastore                                  0  online,open,accessible
  subvol/datastore/data                56463360000  online,open,accessible
    concat/concat1                     56463360000  online,open,accessible
      slice/is4600_ssd_meta92s0          268468224  online,open,accessible
      stripe/files_datastore          56194891776  online,open,accessible
        slice/is4600_fc_40s0           7024361472  online,open,accessible
        slice/is4600_fc_41s0           7024361472  online,open,accessible
        slice/is4600_fc_42s0           7024361472  online,open,accessible
        slice/is4600_fc_43s0           7024361472  online,open,accessible
        slice/is4600_fc_44s0           7024361472  online,open,accessible
        slice/is4600_fc_45s0           7024361472  online,open,accessible
        slice/is4600_fc_46s0           7024361472  online,open,accessible
        slice/is4600_fc_47s0           7024361472  online,open,accessible
  subvol/datastore/log                     245760  online,open,accessible
    slice/is4600_ssd_log72s0               245760  online,open,accessible
```

How to set it up?

- Build XVM volume
  - SSD volume element must be the first member of the concat
    - (assuming here that it's just a single slice)
  - See p.70 of XFS manual for an example
  - Max log size is now 2038 MiB in recent eXFS versions

- Observe reported size of SSD slice in 512 byte sectors
  - xvm show –top vol/the_vol | grep –m 1 slice
    - slice/ssd_meta92s0 10000063 online,open,accessible

- Choose number of AG's to be in SSD
  - Powers of 2
  - Performance of 1, 2 & 4 is poor
  - 8 is good, 16 & 32 probably the sweet spots
  - > 32 – little extra benefit, but extra CPU cost to manage due to fragmentation
  - Following example assumes 16

How to set it up? (cont'd)

- Calculate AG size
    - Assume SSD size is 10000063 sectors

  a) Divide SSD size by chosen number of AGs and truncate
    - 625,003 (assuming 16 AGs)

  b) Convert to filesystem blocks by dividing by 8 and truncate
    - 78,125

  c) Convert back to sectors by multiplying by 8
    - 625,000

- Build filesystem with mkfs.xfs
  - mkfs.xfs –d agsize=625000**s** –l logdev=/dev/lxvm/the_log /dev/lxvm/the_vol

- Mount with ibound
  - mount –o ibound=10000063,logdev=/dev/lxvm/the_log /dev/lxvm/the_vol /mtpnt (ignoring DMAPI and other necessary parameters)

- Update /etc/fstab

How much faster?

Approximate speed-up due to hybrid filesystem:

- dmaudit 6x
- dmscanfs -r 2-4x
- xfsdump -l0 2.5x
- xfsdump -l9 10x
- xfsdump -l9 (just phase 1&2) 20x
- dmfind 25x
- xfsrestore of 30M mostly-offline files 1 hour

NB: apples and oranges comparisons! Different underlying HDD drives.

Gotchas

- “df –i” is unaware that inodes are being constrained to the SSD slice. If it fills, processes will fail with ENOSPC even though “df” shows plenty of space. Keep an eye on usage with “ssd_utilization” script.

- Files can have data blocks assigned in SSD AG’s unfortunately. Run “ssd_intruders” script to reclaim SSD space by freeing dual-state files.

- The configuration rules for filesystems using ibound may result in a filesystem with thousands of AG’s in total, which will cause XFS to consume more CPU searching for free space in a nearly full filesystem. For best performance, ensure that the filesystem is less than ~90% full.

References

- XFS for Linux Administration - Chapter 7: Enhanced XFS Extensions
  - /usr/share/doc/packages/sgi-issp-3.0/LX_XFS_AG/pdf/LX_XFS_AG.pdf
  - http://techpubs.sgi.com/library/manuals/4000/007-4273-006/pdf/007-4273-006.pdf

  Some sections of text from there were “borrowed” for this presentation

- An introduction to SSDs, presented by Jeremy Higdon (SGI) to the SGI Users Group:
  - http://hpsc.csiro.au/users/dmfug/shared/SGI_USER_Group/Jeremy_Higdon-SSD.ppt

- Local CSIRO scripts:
  - http://hpsc.csiro.au/users/dmfug/Meeting_Feb2015/Presentations/CSIRO_SC_Hybrid_FS_scripts/

- Cover picture:
  - http://en.wikipedia.org/wiki/Platypus

Thank you

CSIRO IM&T Scientific Computing
Peter Edwards
Sr Systems Administrator

t +61 3 8601 3812
e peter.edwards@csiro.au
w https://wiki.csiro.au/display/ASC/Scientific+Computing+Homepage