ZFS @ DESY and IN2P3

Martin Gasthuber / DESY

Loic Tortay / IN2P3

ZFS basics

HAMBURG • ZEUTHEN

- pooled storage
  - building stripes/mirrors/single-parity/dual-parity pools out of disks/stripes/mirrors/single-parity/dual-parity objects of any depths
- VM like approach - ins. SIMMS/malloc+free
- no artificial abstraction layer between fs and block interface - zfs knows all !
- end to end data integrity - transactional ops.
  - memory > FS/Drv/Ctrl/Disk > memory
  - self healing / resilvering
- open: http://www.opensolaris.org/os/community/zfs
  - lots of infos/papers in blogs and admin-sites

further features - shortform

- snapshot/rollback - clone
- import/export - handling endianness
- ACLs - NFSv4/NT form
- volume emulator - iSCSI/raw/swap
- hierarchy of filesystems (inheritance)
- prefetch + dynamic striping
- dynamic configuration (not many knobs)
- scrubbing - check all ‘used’ blocks

usage

- just 2 (two) commands
  - *zpool*
    - create/delete/manage pools
      - zpool create mypool mirror c0t0d0 c1t0d0
      - zpool add mypool mirror c2t0d0 c3t0d0
  - *zfs*
    - create/delete/manage zfs instances on pools
      - zfs create mypool/home
      - zfs set mountpoint=/export/home mypool/home
      - zfs create mypool/home/hepixuser
      - zfs set compression=on mypool
      - zfs set quota=100g mypool/home/hepixuser
      - zfs set reservation=100g mypool/home/hepixuser

known issues with other FS/Raid

- unrecoverable ‘read errors’ - *silent* corruption
  - Raid5 is out - insufficient redundancy for large disks
  - Raid6 - not forever !
  - read check summing - rare ! (changing ...)
  - see talk @ Rome-Hepix ‘Disk Storage, Interconnects and Protocols
- Trends
  - software based raidX can handle problems far better - detection + healing
  - enough CPU (cores) available
  - extra HW (costs/reliability)

why use zfs - @DESY and @IN2P3

HAMBURG • ZEUTHEN

- handle large installations
- fast - basically ;-)
- transactional, copy-on-write -> no fsck
- loves to run on JBODs

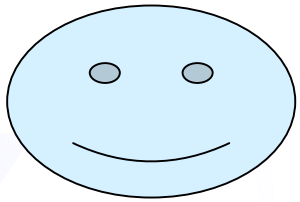
- END to END data integrity
  - today we can/do only checksum complete file on the application level (i.e. dCache) - not on partial reads (what usually happened)

installations

- IN2P3
  - ~ 800 TB (thumper based) since November 2006
  - mostly dCache, also Xrootd and SRB as ‘upper layer’ storage service
- DESY
  - ~ 170 TB starting in December 2006
    - mostly thumper
    - FC JBOD+Raid and SAS/SATA JBOD configs
  - dCache pool
- others: SLAC ... (raise your hands)

observations running zfs

- on FS timescale - zfs is still hot (very young)
- plain FS usage ! no ACL, snapshot etc.
- sometimes slow sync, readdir+stat, *df* info inaccurate
  - see zfs discussion forums
- in general



platforms/OS

- Solaris 10 / OpenSolaris
  - iSCSI, FC-Target, OSD/OST
- MacOSX 10.5
- FreeBSD
- Fuse (userland) / demo ?

- active (hot) discussion inside Linux & OpenSolaris community about license issues building ZFS/Linux - community demands solution !

fits perfect as ‘foundation FS’ for ...

HAMBURG • ZEUTHEN

aggregation/distribution
‘*upper layer*’ storage system

dCache, Lustre (OSD), ...