LessFS and ZFS

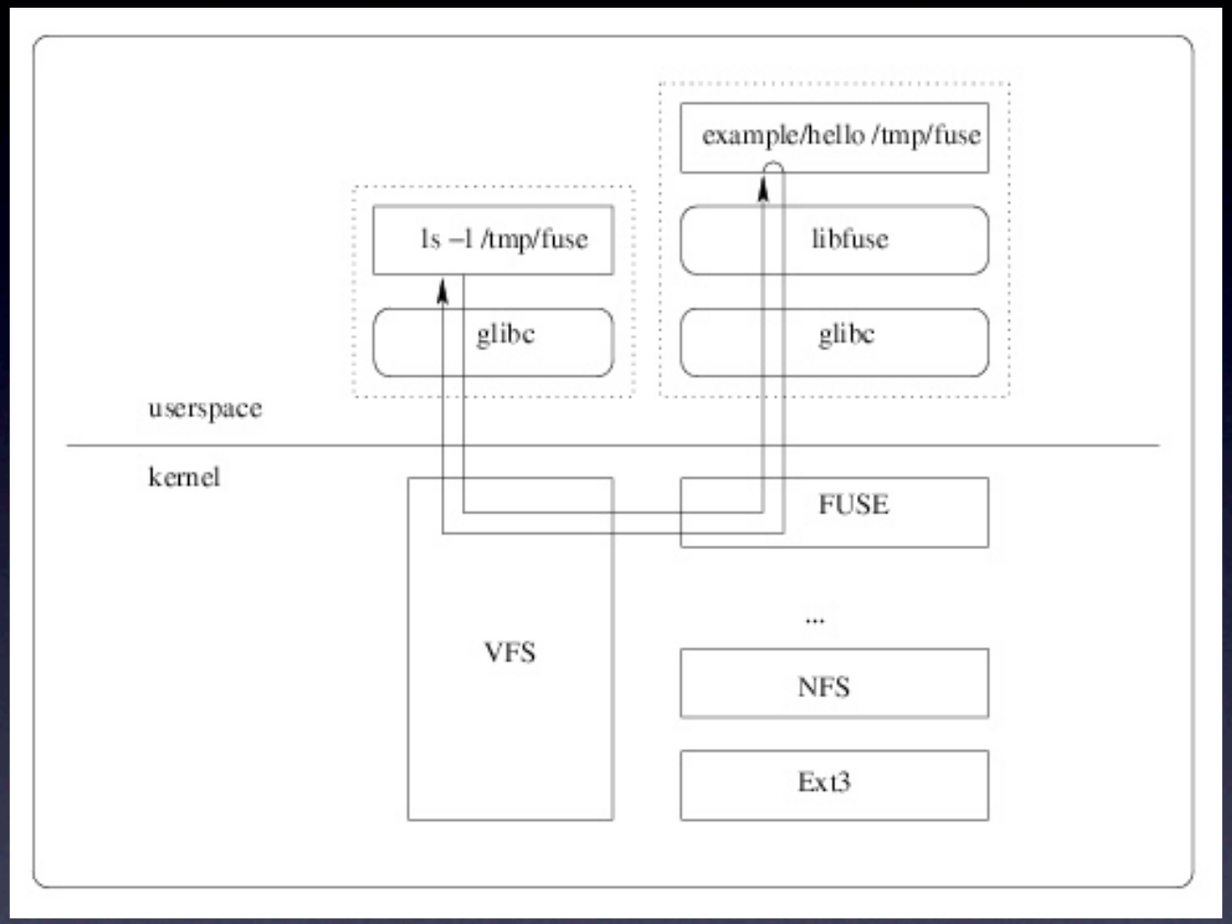
A Tale Of FUSE Filesystems

It always starts with a problem.

- Need to backup VM images and PC backup images and store them efficiently

What is FUSE?

- **F**ilesystem in **USE**r space
- Not to be confused with Free UNIX Spectrum Emulator
- Ports for Linux, BSD and OS X
- Initially developed for AVFS



FUSE Architecture

FUSE Implementations

- Rapid prototyping can be done under fuse and high level languages with little knowledge of kernel programming
- FUSE filesystems can be written in C, C++, Perl, Ruby and Python

Notable FUSE Filesystems

- SSHFS
- NTFS-3G
- CryptoFS
- LessFS
- ZFS

What is LessFS?

- "A high performance inline data deduplicating filesystem for Linux."
- Mostly POSIX compliant
- Built-in compression and strong encryption
- Choice of "back-end" systems
- Supports master/slave replication

Uses for LessFS

- MAID
- Archival backups
- Offline VM images/backups
- Backing up SAN/iSCSI devices/Oracle RDM
- Tier 2 “near-line” File Storage

LessFS Requirements

- FUSE 2.8.x
- Tokyo Cabinet 1.4.x
- ZLIB, BZLIB, OpenSSL

LessFS Performance

- On modern hardware with 12 low end SATA drives lessfs can reach speeds up to 130MB/s for an 100% unique dataset. It will reach speeds up to 170MB/s for previously stored data blocks.

LessFS Configuration

- Install FUSE 2.8.x, Tokyo Cabinet 1.4.46
- ./configure (fix deps); make ; checkinstall
- The default config file provided is not rational. 64G of physical space, 64G dedupe space.
- lessfsconfig.sh will help you get to a configuration for /etc/lessfs.cfg, but won't hand it to you
- Use 64K or 128K blocks for best performance
- `BLOCKUSAGE_BS = ((blockdata_size*1024*1024*1024)/(block_size*1024))*2`
- `FILEBLOCK_BS = ((fileblock_size*1024*1024*1024)/(block_size*1024))*2`

LessFS Configuration

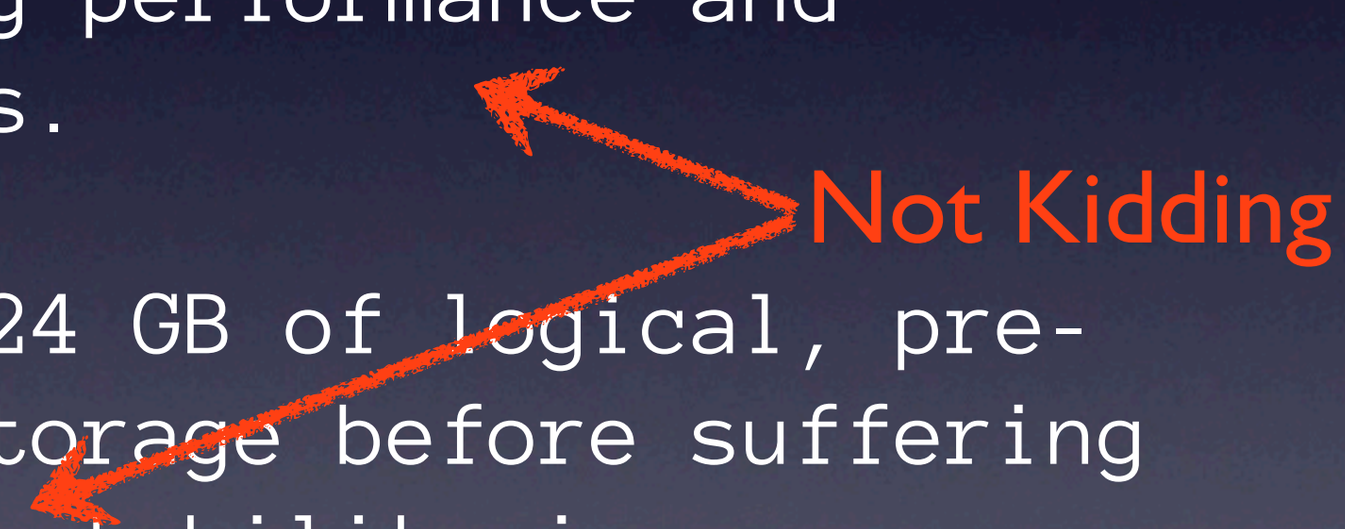
```
root@lessfs:~/lessfs-1.1.9.6# ./lessfsconfig.sh --check /etc/
lessfs.cfg

Statistics for: /etc/lessfs.cfg

This configuration will need 1048 MB of available RAM to operate well.
  512 MB for cache
  4 MB for for blockdata database buckets
  16 MB for for fileblock database buckets
  4 MB for for metadata database buckets
  512 MB for block write buffer

This configuration will:
  Use up to 128 GB of physical space before suffering performance and
stability issues.
  Supply up to 1024 GB of logical, pre-deduplicated, storage before
suffering performance and stability issues.
```

LessFS Configuration

This configuration will:

Use up to 128 GB of physical space before suffering performance and stability issues.

Supply up to 1024 GB of logical, pre-deduplicated, storage before suffering performance and stability issues.



Mounting LessFS

```
lessfs /etc/lessfs.cfg /media/lessfs \
-o use_ino,readdir_ino, default_permissions, allow_other,\
big_writes,max_read=131072,max_write=131072
```

LessFS Configuration

- PLAN AHEAD!
- **Extremely** sensitive to configuration values that are poorly documented
- CPU and RAM intensive
- Turn on background delete, use threads
- It isn't magic compression

Real Example

```
root@lessfs:~# du -h -s /media/dedupe
6.8G    /media/dedupe

root@lessfs:~# du -h -s /media/lessfs
40G     /media/lessfs

root@lessfs:~# df -h | egrep "(dedupe|lessfs)"
/dev/mapper/vg00-lv03  48G  7.0G   38G  16% /media/dedupe
lessfs                 48G  7.0G   38G  16% /media/lessfs
```

What is ZFS?

- "ZFS is the most advanced file system ever invented"
- ZFS can address 256 quadrillion zettabytes of storage, handle a maximum file size of 16 exabytes
- Provable integrity - it checksums all data (and meta-data)
- Atomic updates
- Instantaneous snapshots and clones
- Built-in (optional) compression and deduplication
- High scalability
- Pooled storage model
- Built-in stripes (RAID-0), mirrors (RAID-1) and RAID-Z

Uses for ZFS

- NAS Storage
- Home Directories
- Taking combining different disk sizes to one pool

ZFS Requirements

- FUSE
- zfs-fuse
- Debian varients have it already
- Storage

RAID-what?

- The problem with RAID-5
- RAID-1
- RAID-Z{1,2,3}
- Combinations

Advanced ZFS

- Mirrored RAID-Z
- Copying, migrating and moving data
- Cache devices
- Log devices

ZFS Filesystem Creation

```
root@zfs:~# zpool create test raidz1 /
zfs/disk1 /zfs/disk2 /zfs/disk3
```

ZFS Configuration

```
root@zfs:~# zfs set dedup=on test
root@zfs:~# zfs set compression=gzip test
root@zfs:~# zfs set copies=3 test

root@zfs:/zfs# zpool status test
  pool: test
 state: ONLINE
 scrub: none requested
config:

        NAME             STATE     READ WRITE CKSUM
        test             ONLINE       0     0     0
          raidz1-0       ONLINE       0     0     0
            /zfs/disk1   ONLINE       0     0     0
            /zfs/disk2   ONLINE       0     0     0
            /zfs/disk3   ONLINE       0     0     0

errors: No known data errors
```

ZFS Disk Failure

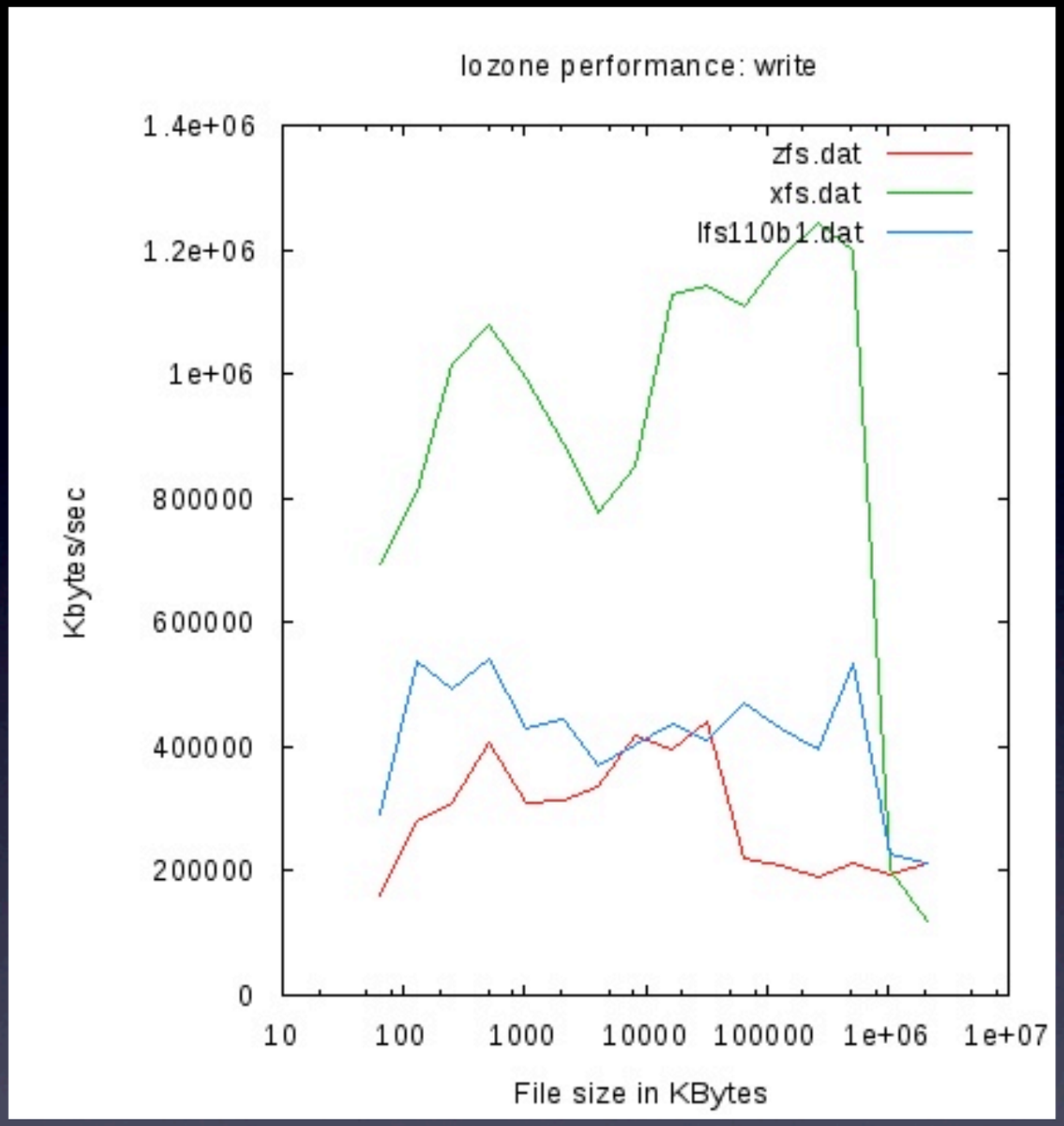
```
root@zfs:~# dd if=/dev/random of=/zfs/disk3 bs=1024 count=2048000

root@zfs:~# zpool status
  pool: test
 state: ONLINE
status: One or more devices could not be used because the label is missing or
        invalid.  Sufficient replicas exist for the pool to continue
        functioning in a degraded state.
action: Replace the device using 'zpool replace'.
   see: http://www.sun.com/msg/ZFS-8000-4J
 scrub: none requested
config:

        NAME            STATE     READ WRITE CKSUM
        test            ONLINE       0     0     0
          raidz1-0      ONLINE       0     0     0
            /zfs/disk1  ONLINE       0     0     0
            /zfs/disk2  ONLINE       0     0     0
            /zfs/disk3  UNAVAIL      0     0     0  corrupted data

errors: No known data errors
```

ZFS Healing

```
root@zfs:~# zpool replace test /zfs/disk3
root@zfs:~# zpool status test
  pool: test
 state: DEGRADED
status: One or more devices is currently being resilvered.  The pool will
        continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
 scrub: resilver in progress for 0h0m, 14.38% done, 0h1m to go
config:

        NAME                    STATE     READ WRITE CKSUM
        test                    DEGRADED     0     0     0
          raidz1-0              DEGRADED     0     0     0
            /zfs/disk1          ONLINE       0     0     0
            /zfs/disk2          ONLINE       0     0     0
            replacing-2         DEGRADED     0     0     0
              /zfs/disk3/old    UNAVAIL      0     0     0  cannot open
              /zfs/disk3        ONLINE       0     0     0  48.6M resilvered

errors: No known data errors
```

ZFS Native

- Native port of ZFS to a kernel module
- Not distributable with Linux
- Work in progress



Relative FUSE Performance

The Future?

- LessFS won't be included in distributions for a while
- ZFS is hampered by non-GPL licenses (and Oracle?)
- BTRFS is the going to be the native kernel equivalent
- Slated to have a comparable feature set as ZFS
- In testing in 2.26.29+, available in RHEL6, Ubuntu 10.10, Debian Squeeze

The Solution

- LessFS with EXT4 backing FS
- FUSE on FUSE didn't seem prudent
- What we really wanted: LessFS on ZFS pools with expansion and data protection

Links

- FUSE: http://fuse.sourceforge.net/
- LessFS: http://www.lessfs.com
- LessFSConfig: http://stashbox.org/900747/lessfsconfig-0.0.2.tgz
- ZFS FUSE: http://zfs-fuse.net/
- ZFS Native: http://zfsonlinux.org/

Questions?

- Slides available from CPOSC.org
- gorkab@mysterons.org