OpenZFS

Why libshare? WHY!

•••

George Wilson
gwilson@delphix.com
@zfsdude

Background

OpenZFS

- Utilize *sharenfs* from our applications to export NFS filesystems
- Properties are maintained with ZFS filesystem
- Allow for migration between platforms
- Utilize *libshare* functionality
  - built-in locking and caching
  - tightly coupled with NFS

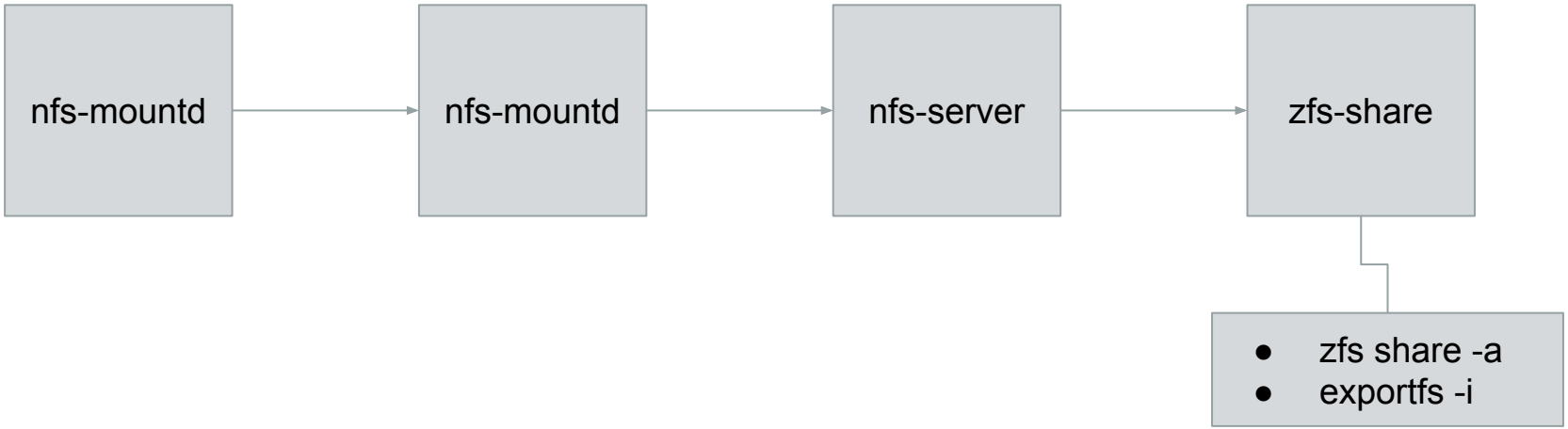
Our *sharenfs* journey on Linux

OpenZFS

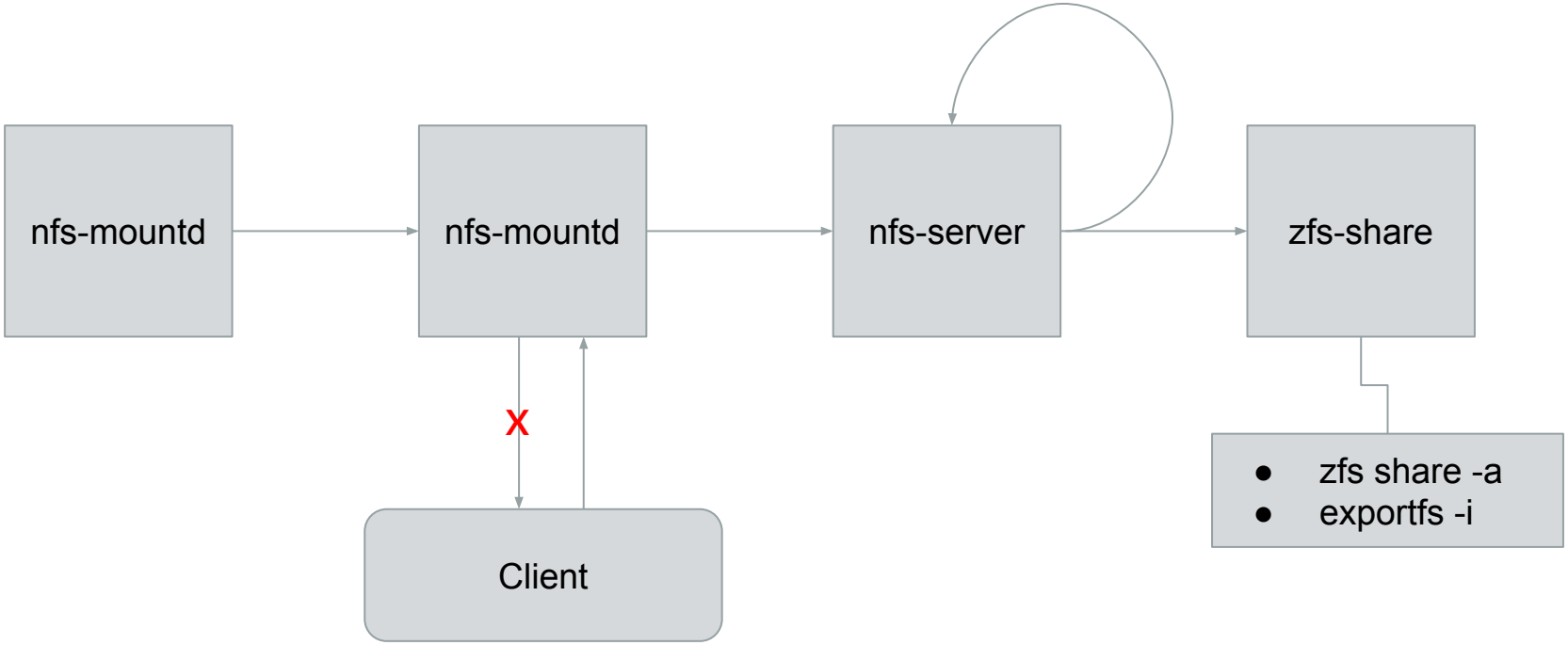
- Concurrency Issues
  - Multiple threads attempting to share/unshare a dataset
  - Added file locking as a temporary solution
- Scalability Issues
  - Constanting reading /proc/self/mounts
  - Enabled caching of file to avoid excessive number of reads
- Stale File Handles
  - Rebooting/Restarting nfs server results in mount failures

NFS/ZFS Systemd Dependencies

OpenZFS

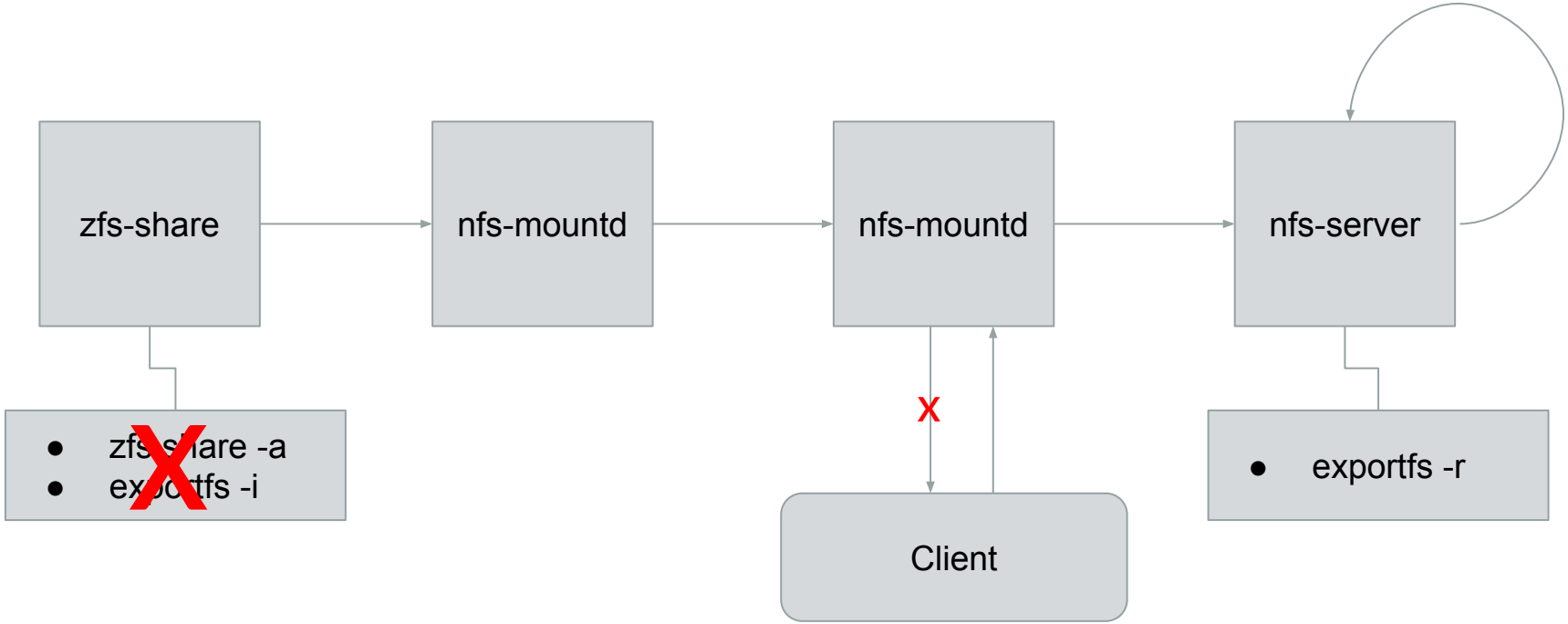


Problem with dependencies



Reorder Dependencies

OpenZFS



ZFS / NFS interactions

OpenZFS

REQUIREMENTS

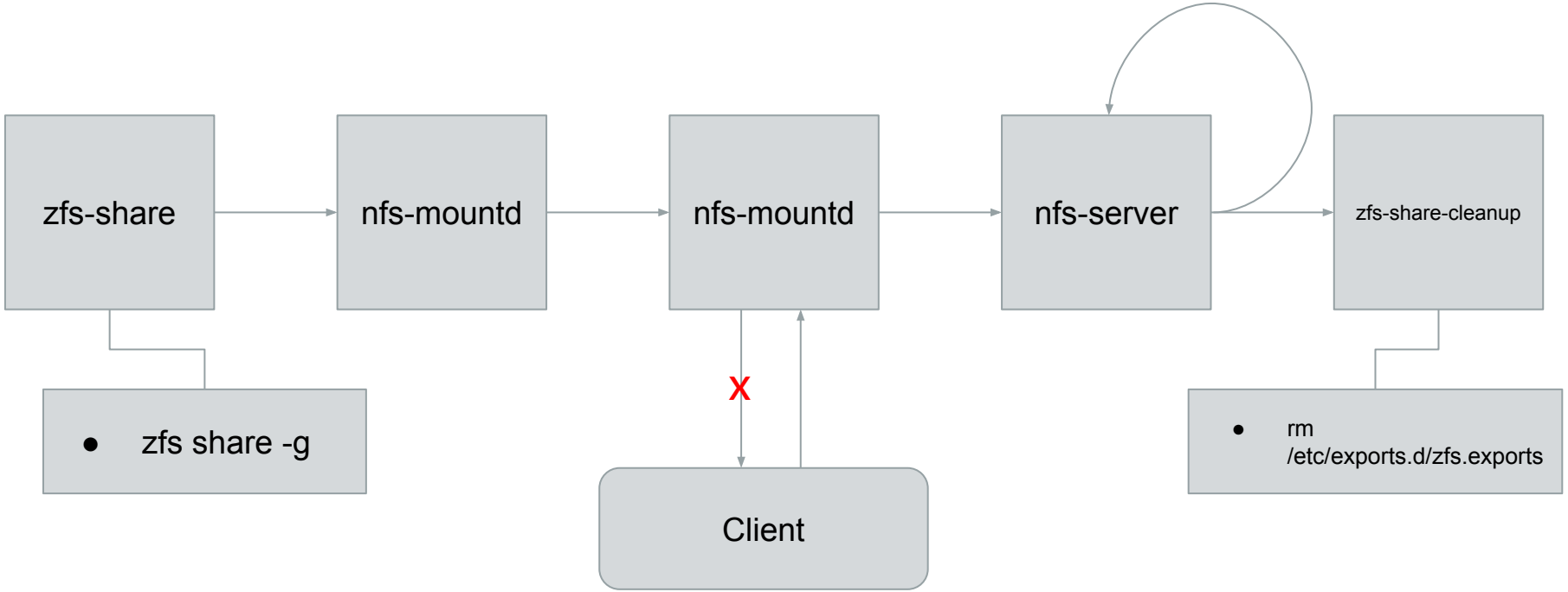
- ZFS sharenfs logic needs to happen before mountd/nfsd
- Ideally should have close ties to nfs service

SOLUTION

- Utilize /etc/exports
  - Automatically consumed by nfs-service
- Need a way to generate all exports
  - Replace “zfs share -a” with “zfs share -g”
  - Generates nfs-service consumable output
  - Much faster than “zfs share -a”
- Introduce cleanup service

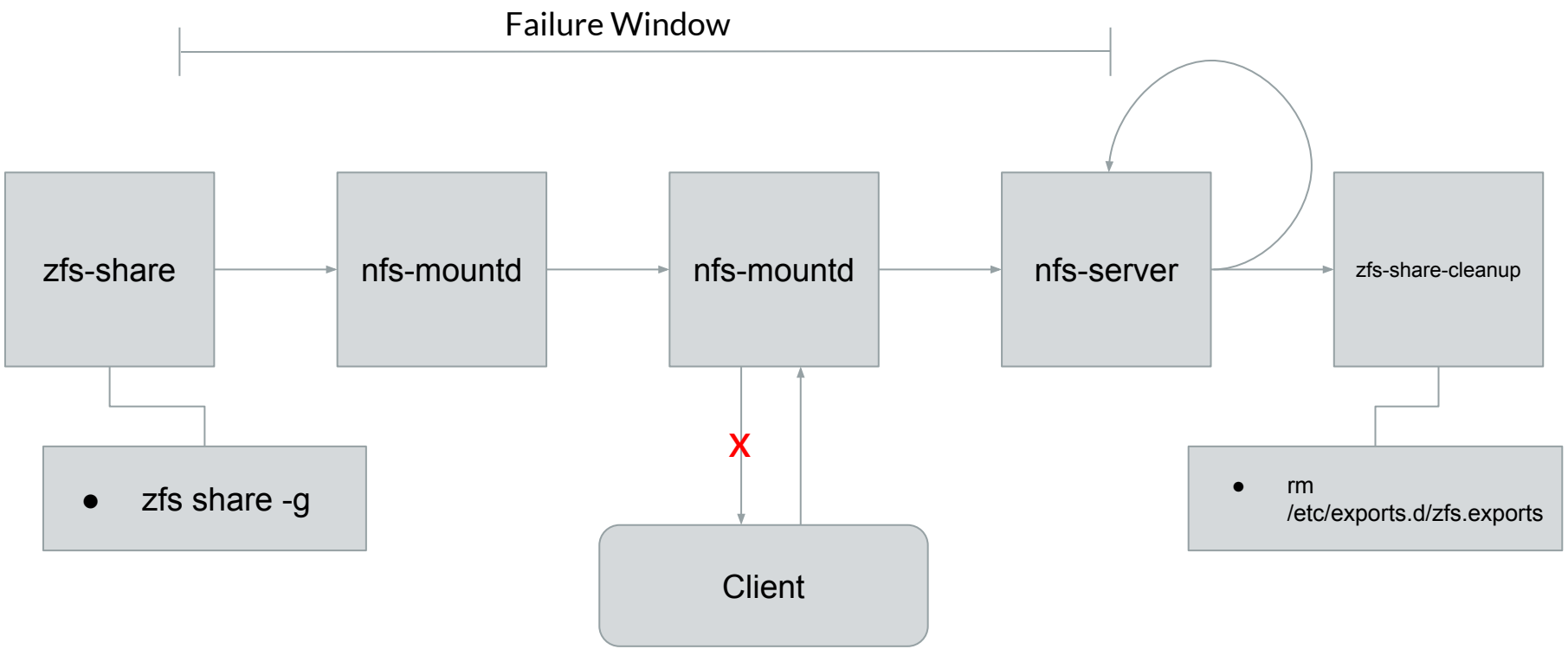
New Dependencies

OpenZFS



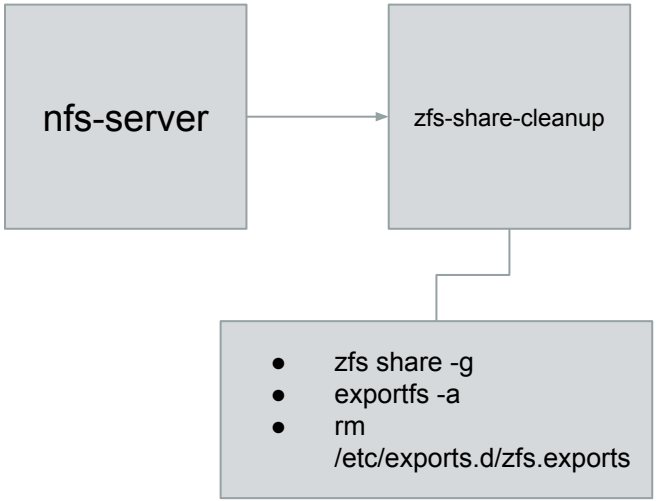
Failure Window

OpenZFS

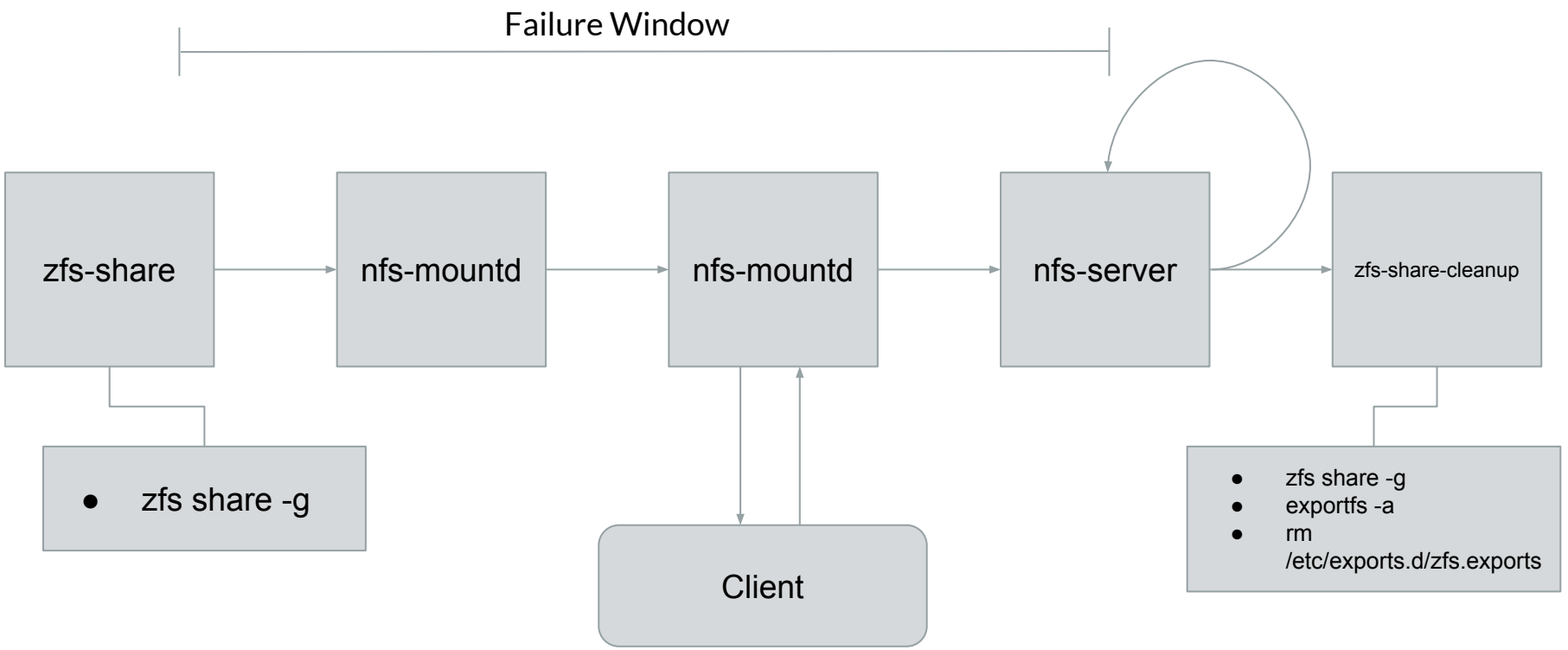


Closing the window (mostly)

- Extend cleanup service
- Regenerate any exports file
- Re-export all filesystems



Temporary Solution

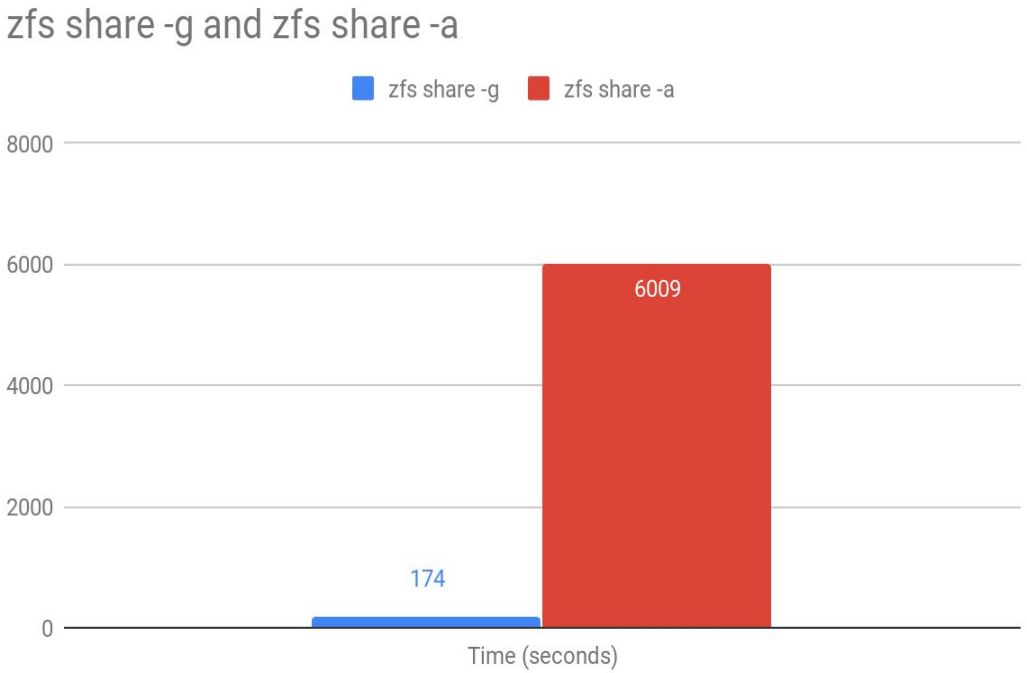


Performance Impact

OpenZFS

- VM with 64GB of RAM
- 60K+ filesystem
- 8 vCPUs

Over 3000% Improvement



Future Work

OpenZFS

- Redefine libshare
  - Define simplified API for loosely coupled platforms
- Rework Linux libshare
  - Update /etc/exports or /etc/export.d directly
  - Interact with systemd as required
- First phase, focus only on NFS sharing
  - SMB shares need more investigation

Questions?