Stress testing of ZFS/FreeBSD port

Andriy Gapon <andriy.gapon@hybridcluster.com>

HybridCluster

OpenZFS Europe, 2014

Motivation

- Deadlocks and crashes
    - in ZPL port to FreeBSD VFS (implementation of vnode operations)
    - in DSL interaction with VFS (suspension, unmounting)
- Reactive fixing
- Proactive discovery through random interactions

Existing tools

Evaluated tools

fsstress, fsx, pjdfstest, xfstests

- Most test correctness of corner cases handling
- As well as correctness of sequences of operations
- Concurrent / random testing is very limited
    - either operations performed in different directories
    - or at a single directory level

Our tool

- Very simple but practically powerful approach
  - random operations on random paths
  - path length is random within a limited range
  - path component names are randomly chosen from a limited repertoire
  - a fixed number of processes operates on the same hierarchy
  - many variants but probability of interaction is rather high as well
- Correctness of operations is not verified
- Only resilience of the OS is tested
- Some implementation issues
  - paths can grow to unlimited depth because of directory re-parenting (move, rename)

Current implementation

- Only filesystem operations
- ZFS DSL operations are planned
- Discovered problems
  - ZFS rename operation on FreeBSD concurrent with .. (dot-dot) look-up
  - ZPL locking vs FreeBSD VFS locking: sometimes redundant, sometimes insufficient

Why rename is hard

- Rename is also a move
- Moving directories is the only operation that could break directory hierarchy rules
- Must ensure that directory does become a descendant of itself
- Must handle concurrent renames
- .. look-up is a bonus problem with FreeBSD VFS implementation

Thank you!

- Thank you for listening!
- Questions?