Multi-thread-epoll patch results Ben England, 9/4/14

- Quick-read on (default) improves reads by 17%
- Open-behind on (default) improves reads by 169%
- Patch 3842 improves reads by 13%
  - Bottlenecked by SSD!
- Patch 3842 improves creates by 41%
  - Still bottlenecked in e-poll threads
- 4 epoll threads improves creates by 87%
- 7 epoll threads no better than 4
  - CPU utilization increasing to 60-70% of server cores

Methodology

git clone git://forge.gluster.org/glusterfs-core/glusterfs.git

- Create two copies, mtep and nomtep
- Apply multi-thread-epoll patch to “mtep” copy
- ./configure –prefix=/usr/local/[no]mtep
- Make ; make install
- Create tarball from each and copy them to servers
- Remove any gluster rpms from servers
- Untar into install directory on servers
- Install /sbin/mount.glusterfs and make it point to correct tree
- Set PATH environment to point to right tree

Baseline configuration

- 1 SSD brick with XFS filesystem on each server
  - mkfs -t xfs -f -i size=512 -n size=8192 -d agcount=16
  - Mount -t xfs -o noatime,inode64 /dev/sdg /mnt/ssd
- 10-GbE, jumbo frames (MTU=9000)
- Gluster v create v replica 2 gprfc08{3,4}-10ge:/mnt/ssd/brick1
- Gluster v set v io-cache off
- 8 glusterfs mountpoints per server, /mnt/mtep{1-to-8}

Common workload

- 16 smallfile threads per server
- 10240 files per thread
- File size 4 KB
- Drop cache on both servers before read tests