Scaling Ceph-SMB Connections

Sachin Prabhu
IBM

Introductions

- The team
  - IBM / Red Hat
  - Ceph team - SMB service
    - https://github.com/samba-in-kubernetes

IBM

Introductions

- Ceph-SMB service
  - smb manager module
  - container - samba-container project
  - exports cephfs volumes
  - samba vfs module - vfs_ceph_new

IBM

Samba

- The Forking model
  - portability
  - switch uid/gid of running process
  - robustness

IBM

Problem

- Large number of simultaneous clients
  - large number of processes
  - each connection has its own libcephfs stack
    - own metadata and data cache
  - leads to depletion of resources for some workloads

Reproducer

- sit-test-cases - loading test
  - https://github.com/samba-in-kubernetes/sit-test-cases
  - smbprotocol python module
  - multiple threads each opening a new client connection
  - multiple files opened/closed
  - 16 M file size
- fails after 100 simultaneous connections
  - failure caused by memory pressure

Solution

- libcephfs_proxy
- design document in ceph repo
  - doc/dev/libcephfs_proxy.rst
- avoid an independant cache for each client connection
- tested with 1000+ simultaneous connections
- 2 parts
  - libcephfsd daemon process
  - libcephfs_proxy.so library

IBM

Solution

- libcephfsd daemon
  - uses actual libcephfs.so library to connect to cephfs volume
  - centralise libcephfs requests
  - listens to incoming connections from the client at unix socket
    - /run/libcephfsd.sock

IBM

Solution

- libcephfs_proxy.so library
  - provides a subset of low level cephfs API calls
  - to be used in place of libcephfs.so
  - no caching on client
  - forwards requests to libcephfsd daemon over unix socket
- Same configurations share connection
- Some calls need special handling - getcwd, chdir

Performance implications

- SPECstorage - Performance tests
  - CTDB enabled
  - cifs kernel mount
  - Ceph 19.2.0-10, Samba 4.21.0
- Higher Latency
  - SWBuild 89.708 ms vs 140.095 ms
  - VDA 75.933ms vs 97.330 ms
- Overall throughput decreased
  - SWBuild 1438.143 kb/s vs 917.124 kb/s
  - VDA 23001.164 kb/s vs 22817.778 kb/s

Future Plans

- Metadata cache on client end
  - requires synchronous invalidation callbacks through ceph
- Consider other options for connection between libcephfs_proxy.so and daemon process
- Extend low level API calls supported

IBM

Thank you

Sachin Prabhu - sprabhu@redhat.com