ZFS send and receive, performance issues and improvements

BSDCan 2018

Rod Grimes
rgrimes@freebsd.org
Encryption, pipes and context switches need to go!

1) The local use of zfs send | zfs receive.
2) The remote use of zfs send | ssh zfs receive, and zfs send | nc
3) A new option to zfs send and receive, socket.
The local use of zfs send | zfs receive

Context switch per buffer
The local use of zfs send | zfs receive

Context switch per buffer
- Copyin to kernel
- Copyout to user
The local use of zfs send | zfs receive

Context switch per buffer
Pipe buffer size
The local use of zfs send | zfs receive

Context switch per buffer

Pipe buffer size

- Ancient 512 bytes
- Increased to 4k but static
- Increased to 4k with dynamic growth
  - Kva pool used to restrict
- Increased to dynamic size with dynamic growth and shrink
The local use of zfs send | zfs receive

Pipe buffer size
- No consideration of cache size
- No considerations of NUMA
The local use of zfs send | zfs receive

Context switch per buffer

Pipe buffer size

Copyin and Copyout
  – Mtx and lock
  – Uiomove aka slow, not page flipped
The local use of zfs send | zfs receive

Pipe concurrency and locking

- Single buffered
- Single flag and a mutex are the locking
- Dragonfly has made some improvements
The remote use of zfs send | zfs receive

zfs send | ssh zfs receive

- Encryption can become a bottleneck
- Ssh hacks
The remote use of zfs send | zfs receive

zfs send | nc   ssh nc | zfs receive

- So we eliminate ssh
- Ending up with 2 pipes
- One on each end
A new option to zfs send and receive, socket

zfs send -S ip:port
zfs receive -S ip:port
A new option to zfs send and receive, socket

zfs send uses an fd to pass STDOUT

zfs recv uses an fd to pass STDIN

Kernel just expects fd's!!!
POC

Add getopt processing
Connect a socket
Pass to zfs in place of STDIN/OUT
POC Benefits

No context switches
No copyin or copyout
No locking needed
Direct from zfs buffers to mbuf via write(2)
Direct from mbufs to zfs via read(2)
Fewer running processes
  – Zfs user process is sitting blocked on both ends
POC diff

177 line context diff to zfs_main.c

Http://people.freebsd.org/~rgrimes/zfs_send_socket.diff
Future Work

Pipes and cache size
Page flipping pipe
kevent/kqueue
Security Concerns

Not addressed, relies on other mechanisms
Questions?
Thank You

rgrimes@freebsd.org