Whamcloud

Parallelize Filesystem Check for Exascale Storage

Wang Shilong

DDN® STORAGE

What is filesystem check?

Whamcloud

- The system utility **fsck** (*file system consistency check*) is a tool for checking the consistency of a file system in Unix and Unix-like operating systems(Wiki)
- Online
  - Lustre lfsck, ZFS/btrfs scrub
- Offline
  - extN e2fsck, btrfs check, xfs_repair...

Why we need filesystem check?

Whamcloud

- Useful for both developers and administrators
  - Run fsck after every testing to make sure bug free(Lustre tests, xfstest)
- Regular healthy check
  - Scrub running every few weeks
  - e2fsck recommended every 20 mount.
- Repair, isolate errors to make system available
  - Repair inconsistency, lfsck could fix Lustre MDT file owners differ with OST owners.
  - e2fsck fix wrong quota accouting..
  - badblock due to hardware problems..
  - Fsck is a must to restore service or isolate errors..

Lustre ext4 based filesystem and its challenge?

Whamcloud

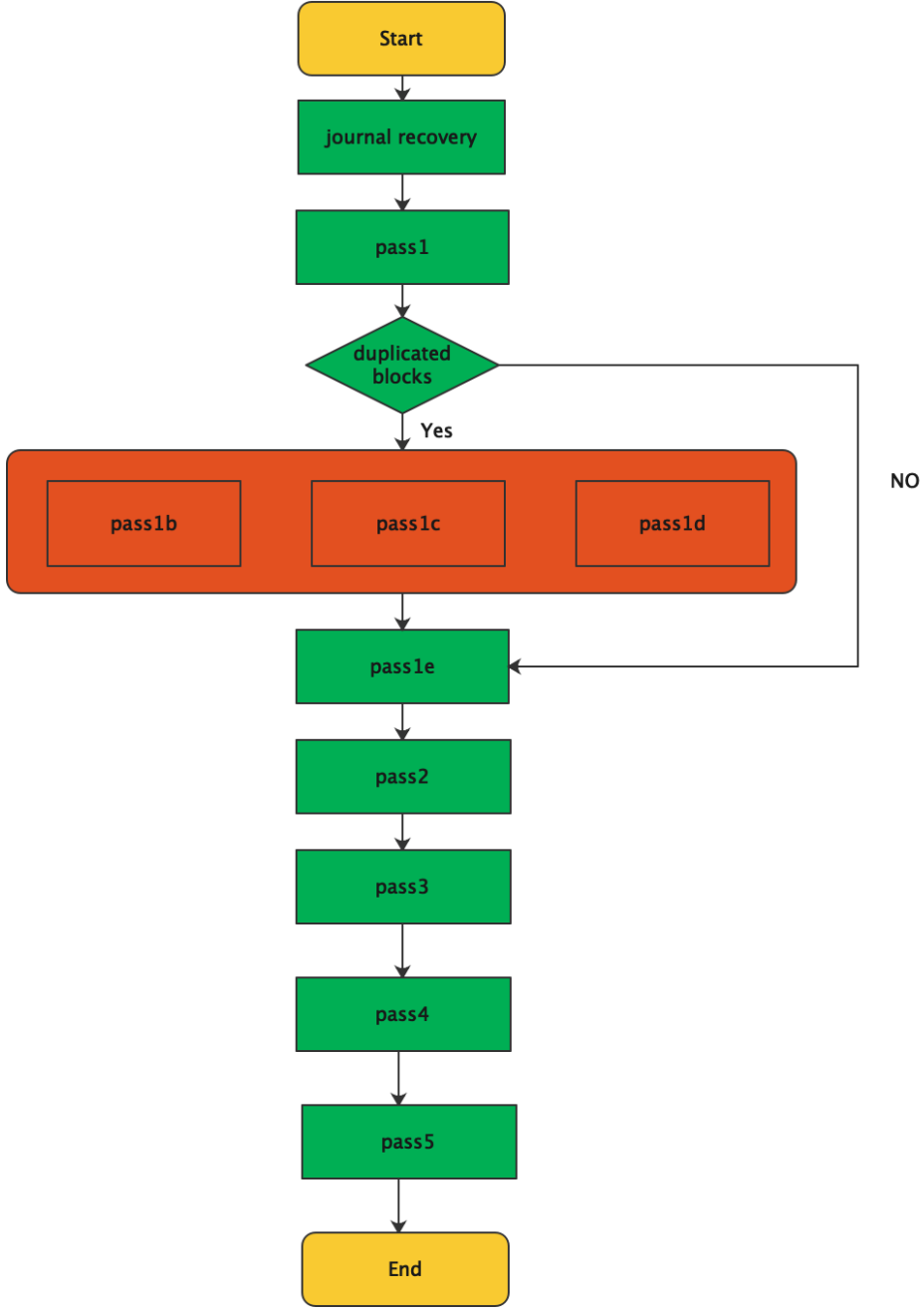
- Single HDD could be 16TiB

- One Lustre OST could be more than 1PiB with LUN System

- But one time check could take more than 5 hours
  - Nightmare because maintenance usually take during night or weekend
- Scale it for next 5 or 10 years?
  - Gurantee scalability for future extentions.

e2fsck process

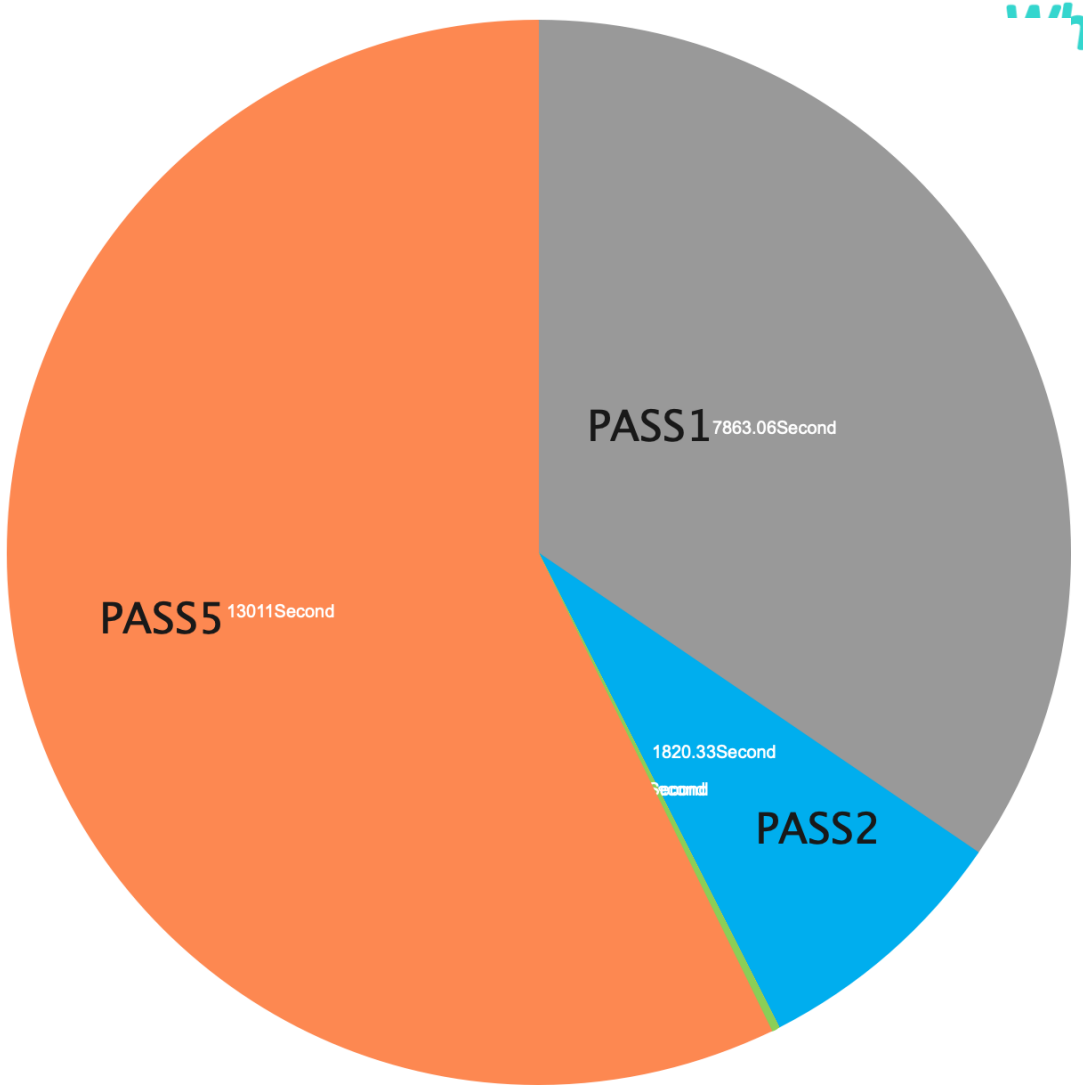
- Pass1
  - inode, xattr, extents
- Pass1[b-d](optional)
  - Fix duplicated blocks errors
- Pass1e
  - Rebuild extents if necessary
- Pass2
  - Check directory
- Pass3
  - Check full path for each directory
- Pass4
  - Check full path for regular files
- Pass5
  - Check block/inode bitmaps



Whamcloud

e2fsck Runtime

- Configurations
- 8 CPU cores, 150GB RAM, 1 x IB-HDR100
- 1 x DDN AI400, 162 x 10TB NL-SAS HDD. DCR (declustering raid) POOL for 1.2P OST
- 400 M 0-byte files on single 1.2 PiB OST
- e2fsck -fn -v -tt /dev/sdr
- Conclusions
- Total 6.3 hours to finish!
- PASS5 took 3.6 hours(~57%) PASS1 took 2.2 hours(~35%)



Need to improve Pass1 Step

Whamcloud

- Pass1 takes more than 35% of the e2fsck time
- Why Pass1 is slow

- Walk through the entire inode table
- On each inode

a. Read and check the inode atributes(IO Bound)
b. Check the blocks used by each inode(IO bound)
c. A lot of inserting and searching of data structures(CPU bound)

- How to improve

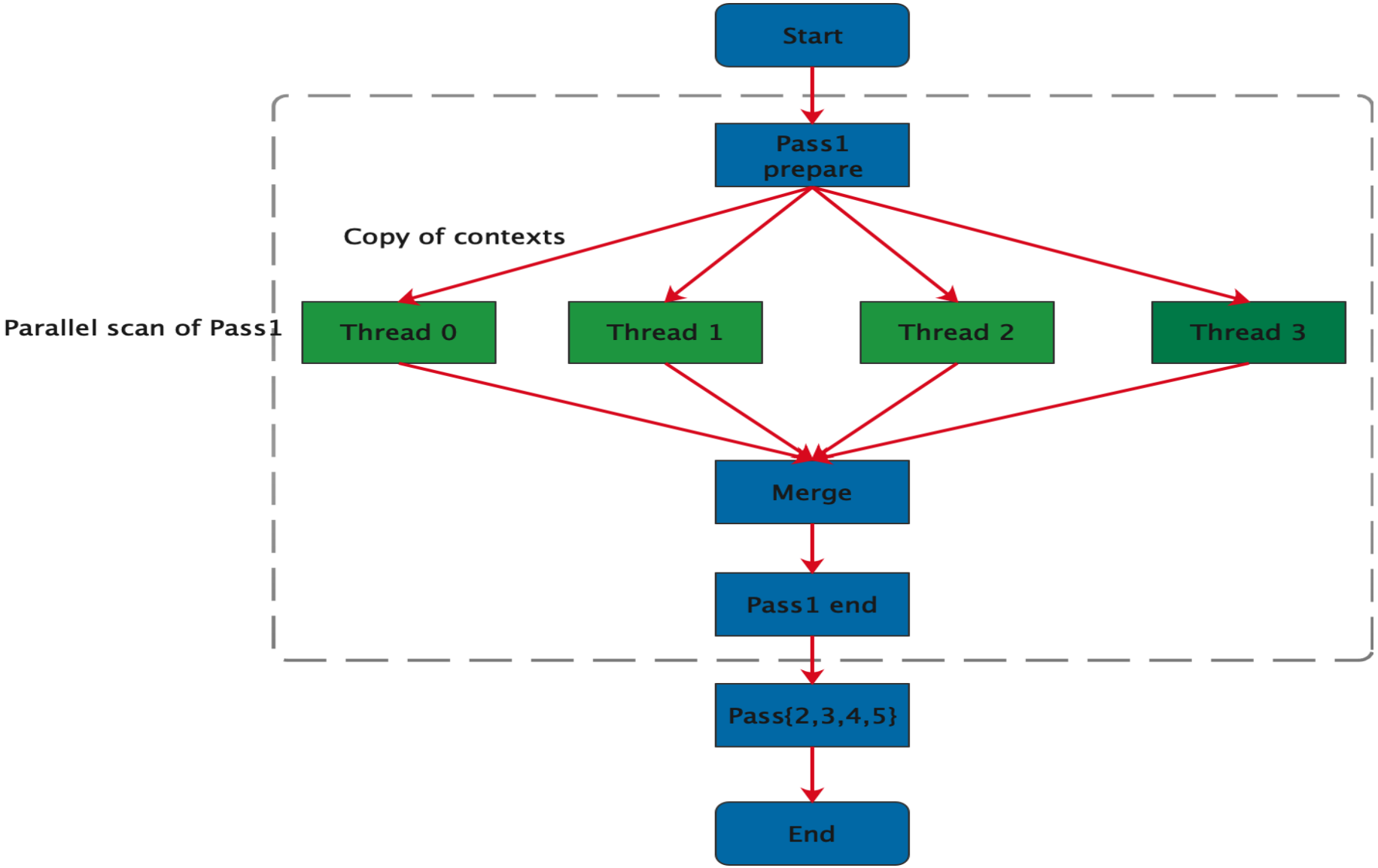
- Readahead for inodes(continuous inside the same block group)
- Fortunately, the check of each inode is almost independente
- Different threads can check diferente inodes in parallel

Challenges and Solutions

Whamcloud

- The result of Pass1 will be used by Pass2/3/4/5 too
  - Share inode_{used,dir,reg,bb,imagic}_map etc, use global locks to serialize operations
  - Separate per thread, but merge them to globally after threads finish.
- Synchronization will be needed between threads
  - 99% of fsck is READ only operations(this need be parallel)
  - Fix operation need be serialized.
  - Some operations like create resize inode could not be parallel.
- Correctness is very hard to confirm
  - Wrong e2fsck would cause/escalate data corruptions
  - Need to pass all existed regression test of e2fsprogs
  - Run single/multiple threads test on same corrupted fs and compare results.
  - Large OST(PiB Size) performance testing and Strict review

Pass1 Design

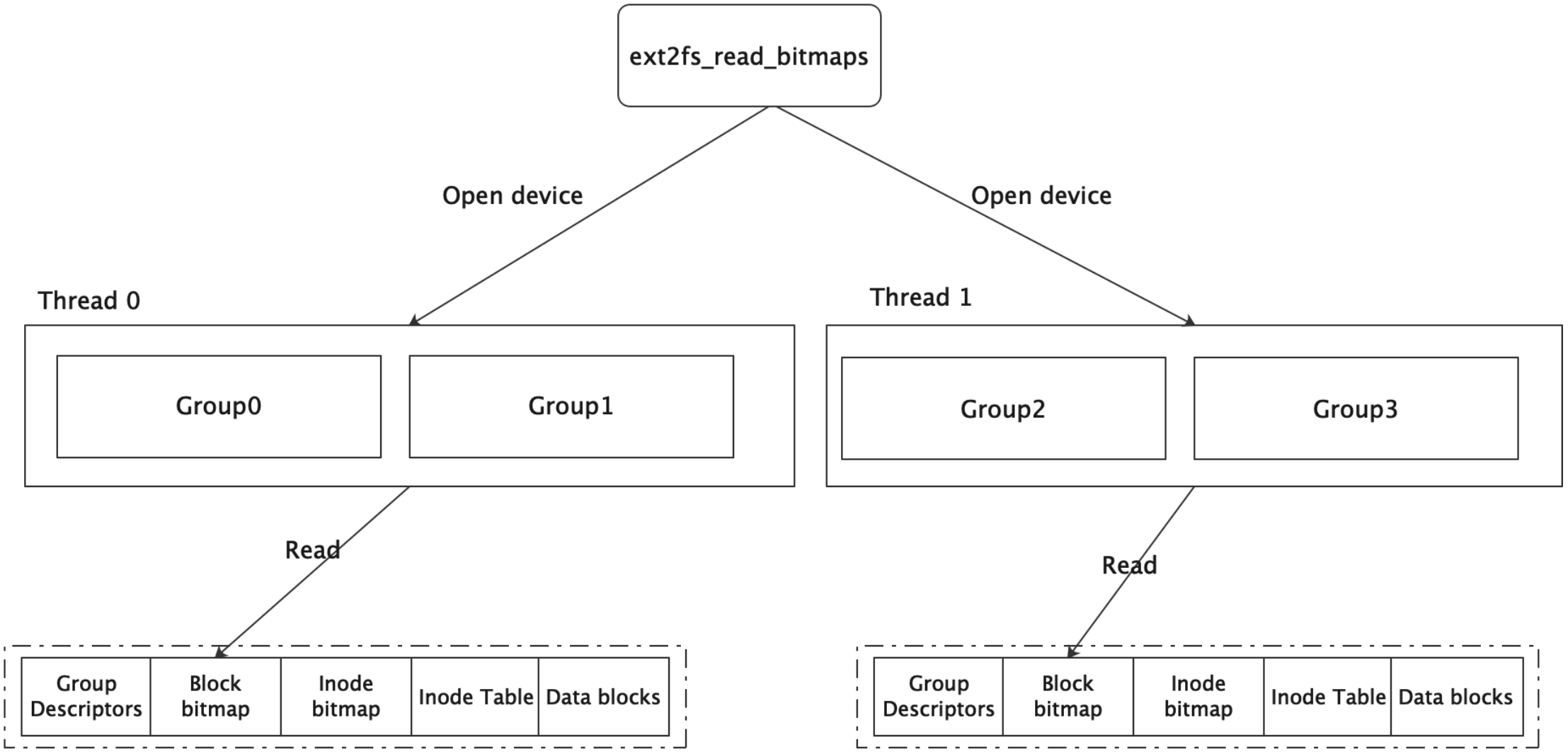


Need to improve Pass5 Step

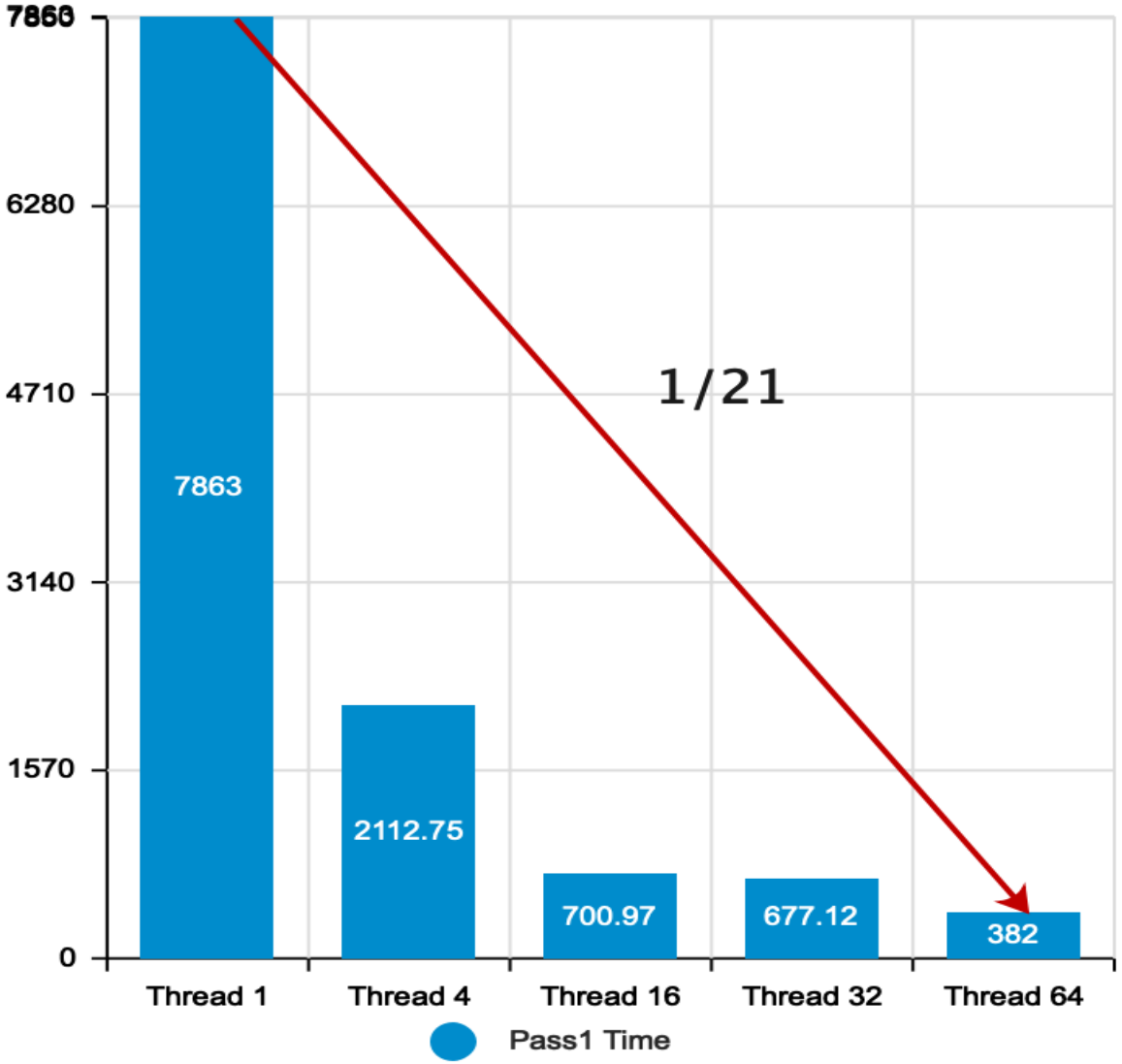
Whamcloud

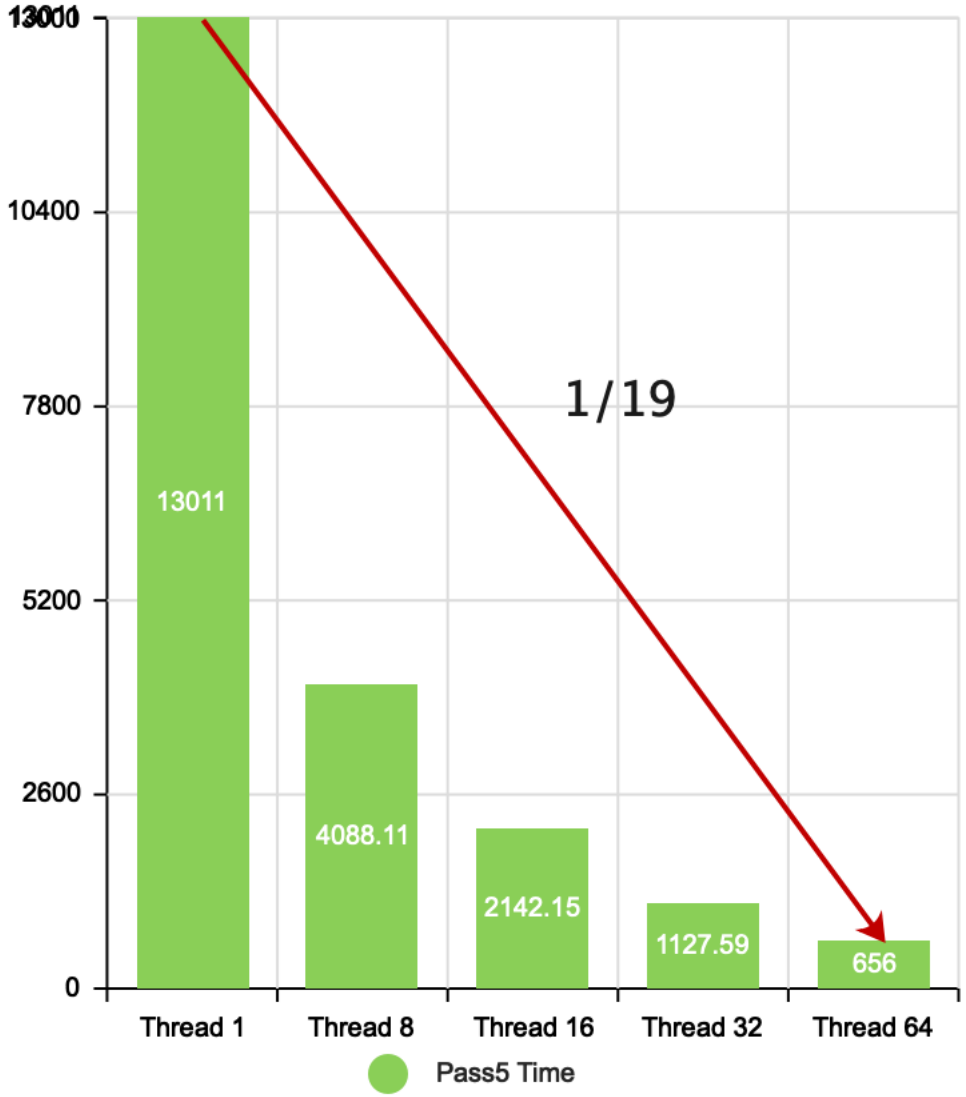
- Pass1 takes more than 57% of the e2fsck time
- Why Pass5 is slow?
  - Load all inode and block bitmaps(small IO 99.5% of total time)
  - Verify inode/block bitmaps and checksum
  - Expand inode size if needed
- Improve Pass5
  - Parallel ext2fs function ext2fs_read_bitmaps()
  - Much simpler than pass1 as this is READ only operation.
  - Parallel other checking in the future(TODO)

Pass5 Design



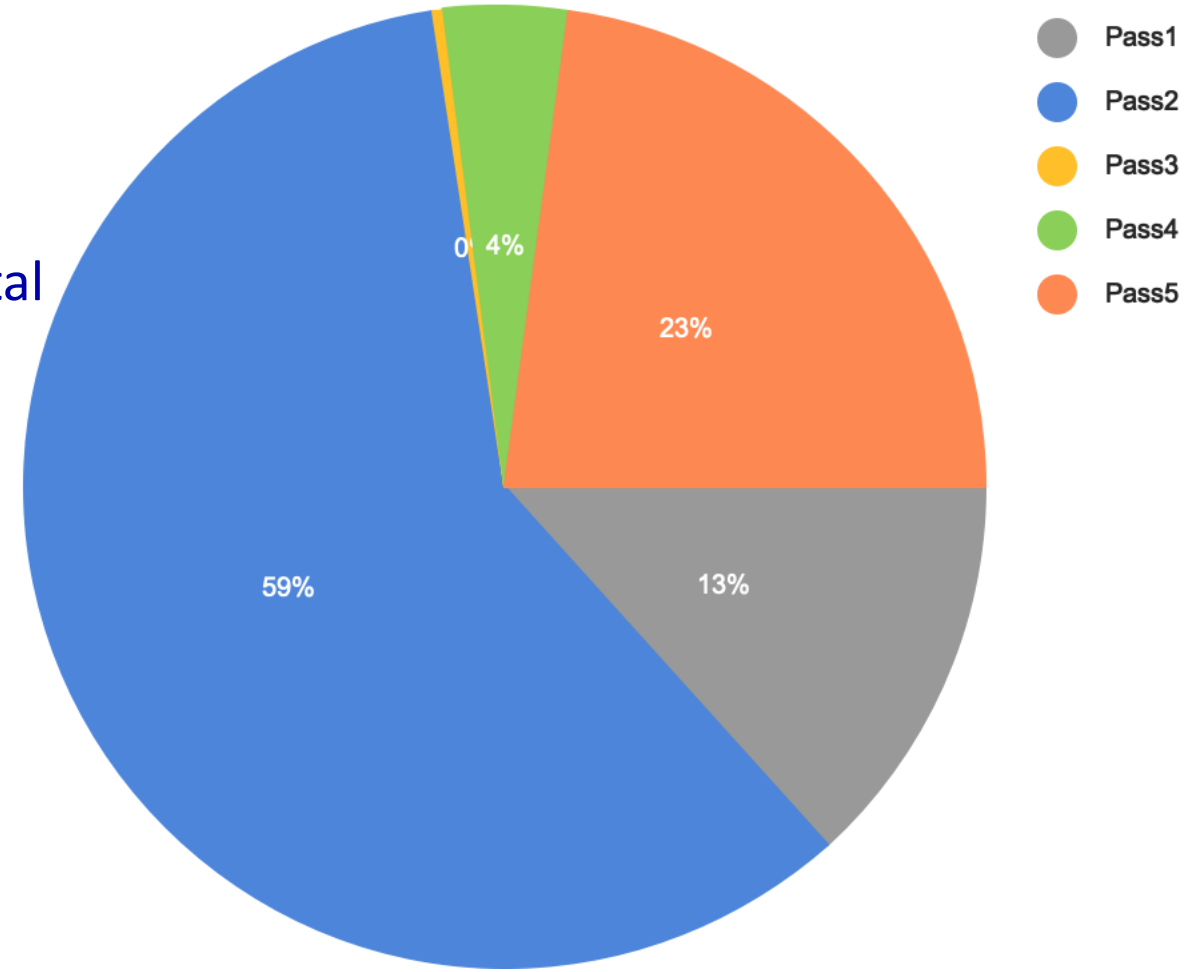
Benchmarking for improvements





Benchmarking for improvements

- About 40 minutes now
  - Total time is 1/14 of original single thread
  - Pass2 cost ~24 minutes(59%)
  - Once multiple thread for Pass2 is done total time will be less than 20 minutes for PiB fs



Current Status

Whamcloud

- 50+ patches pushed on both Lustre gerrit and linux ext4 community
  - https://review.whamcloud.com/#/c/39874/
- Acceptable fsck time now for PiB OST with minimum engineering efforts
  - Speed up ~8X times, from 22792.64 seconds to 2914.98 seconds
- Testing make sure reasonable stability
  - Stability should be most important for fsck
  - Default behavior for fsck is still single thread
  - Pass all existed testing and dozens of times corruption testing.

Future Work

Whamcloud

- More Speed up is possible in the future
  - Pass2 is still single thread, important for big MDT device.
  - Better load balance policy, dynamical thread pool.
- Need verify total memory usage with number OSTs fsck running.
  - pfsck itself cost little memory
  - Even single thread fsck memory cost could be challenging for the large OST.
- large scale testing on NVME device is missing
  - Much faster even single thread
  - Different challenges VS HDD based

Whamcloud

Thank You!

DDN® STORAGE