

whamcloud

The logo for Whamcloud features the word "whamcloud" in a bold, lowercase, sans-serif font. A thick blue horizontal line underlines the text. On the right side, a blue graphic element resembling a stylized '3' or a cloud shape is positioned above the end of the underline, with its top curve overlapping the top of the 'd'.

Parallel Directory Operations performance tests

- Liang Zhen
Whamcloud, Inc.
liang@whamcloud.com

Testing environment

- MDS
 - 2 Intel 5650 6-core CPUs (2-HT each core)
 - Total 12 cores, 24 HTs
 - 24G RAM
 - 1 SSD external journal, Sata disk
- OSS
 - 16G RAM
 - 2 OSSs, 2 OSTs on each OSS
- 16 clients
 - Intel Xeon E5507: 4-core
- Networking
 - QDR infiniband

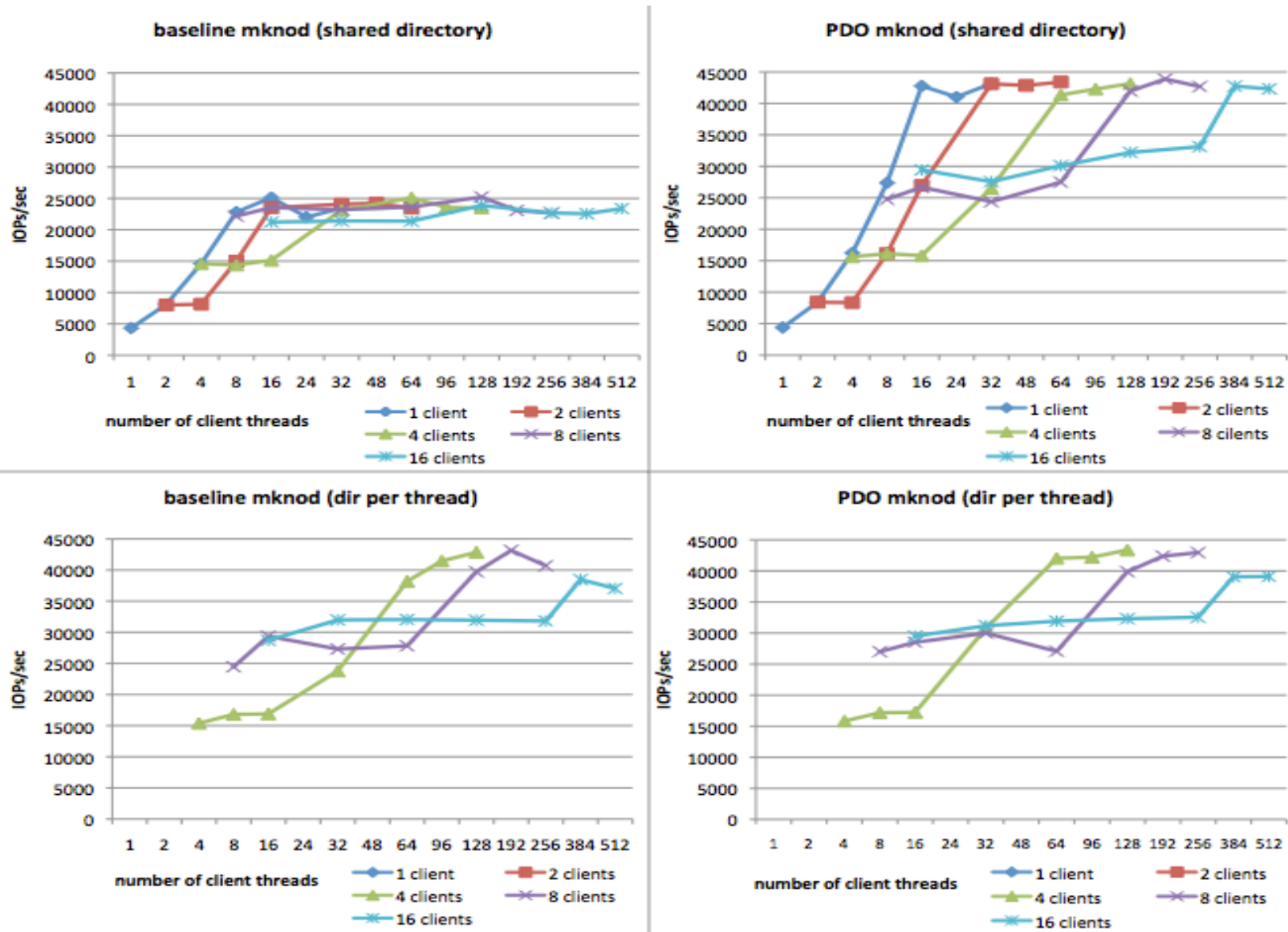
Testing tool

- mdtest with a few changes
 - shared directory tests with multi-mount
 - mknod
- Test methodology
 - 1, 2, 4, 8, 16 clients
 - 1, 2, 4, 8, 16, 24, 32 threads on each client
 - Max to 512 client threads
 - Equal to max number of service threads on server
 - mknod, opencreate (0-stripe), opencreate (1-stripe), unlink
 - Shared directory
 - all client threads run under same target directory of different mount points, i.e: thread[0, 1, 2... 31] run under /mnt/lustre.[0, 1, 2... 31]/testdir
 - Directory per thread
 - Verify no performance drop for “directory per thread” case

Mknod performance

- Total 1 million files
- Share directory
 - [1, 2, ..., 16] clients * [1, 2, ..., 32] threads
- Directory per thread
 - [4, 8, 16] clients * [1, 2, ..., 32] threads
- mknod is a single RPC
 - mknod performance can show PDO improvement with MDS stack

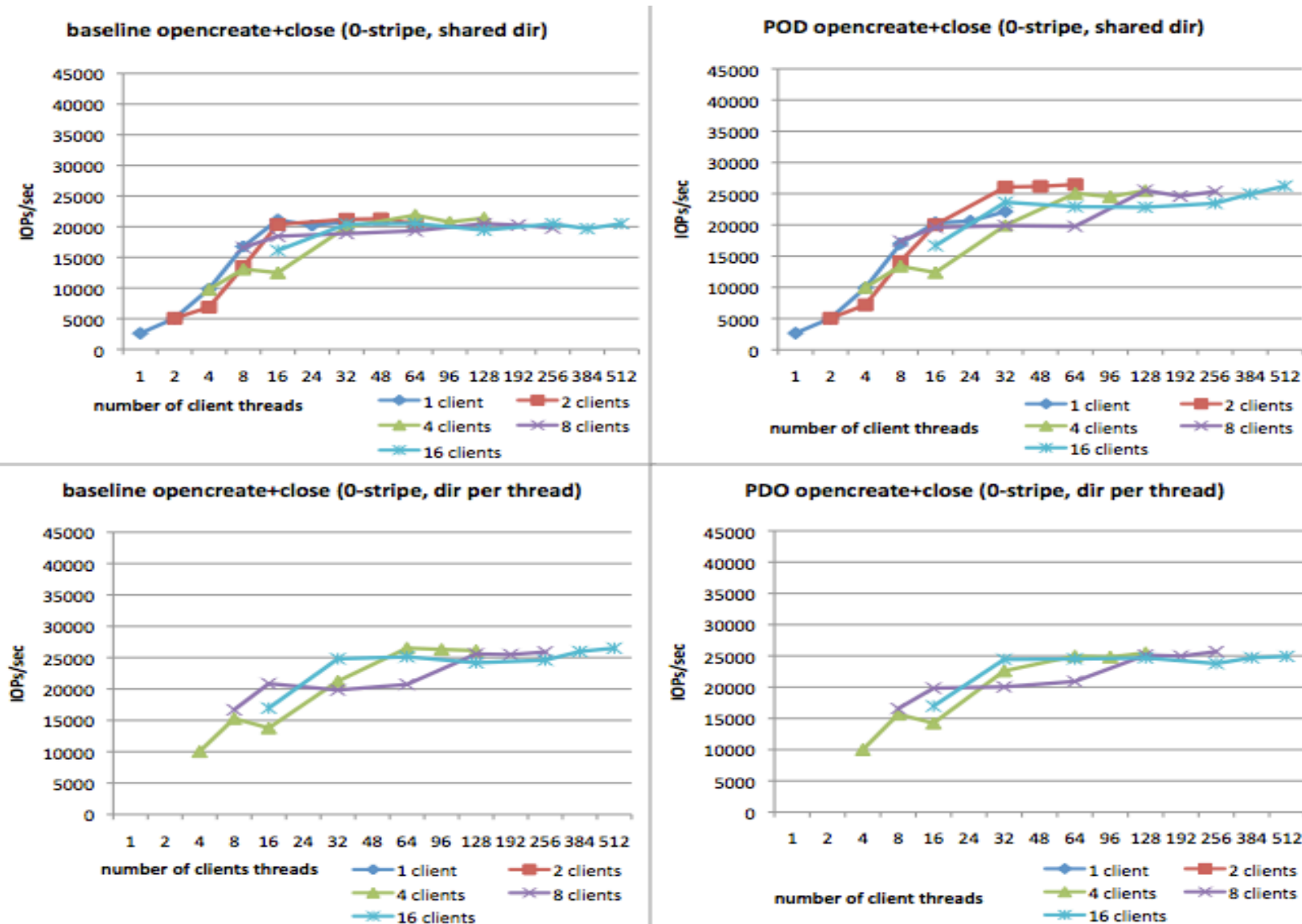
mknod performance (1 million files)



0-stripe Opencreate+close performance

- Total 1 million files
- Share directory
 - [1, 2, ..., 16] clients * [1, 2, ..., 32] threads
- Directory per thread
 - [4, 8, 16] clients * [1, 2, ..., 32] threads
- 2 RPCs for opencreate+close
 - PDO project can't help anything for "close"
 - Opencreate+close can't show full improvement on "create" because extra "close" RPC

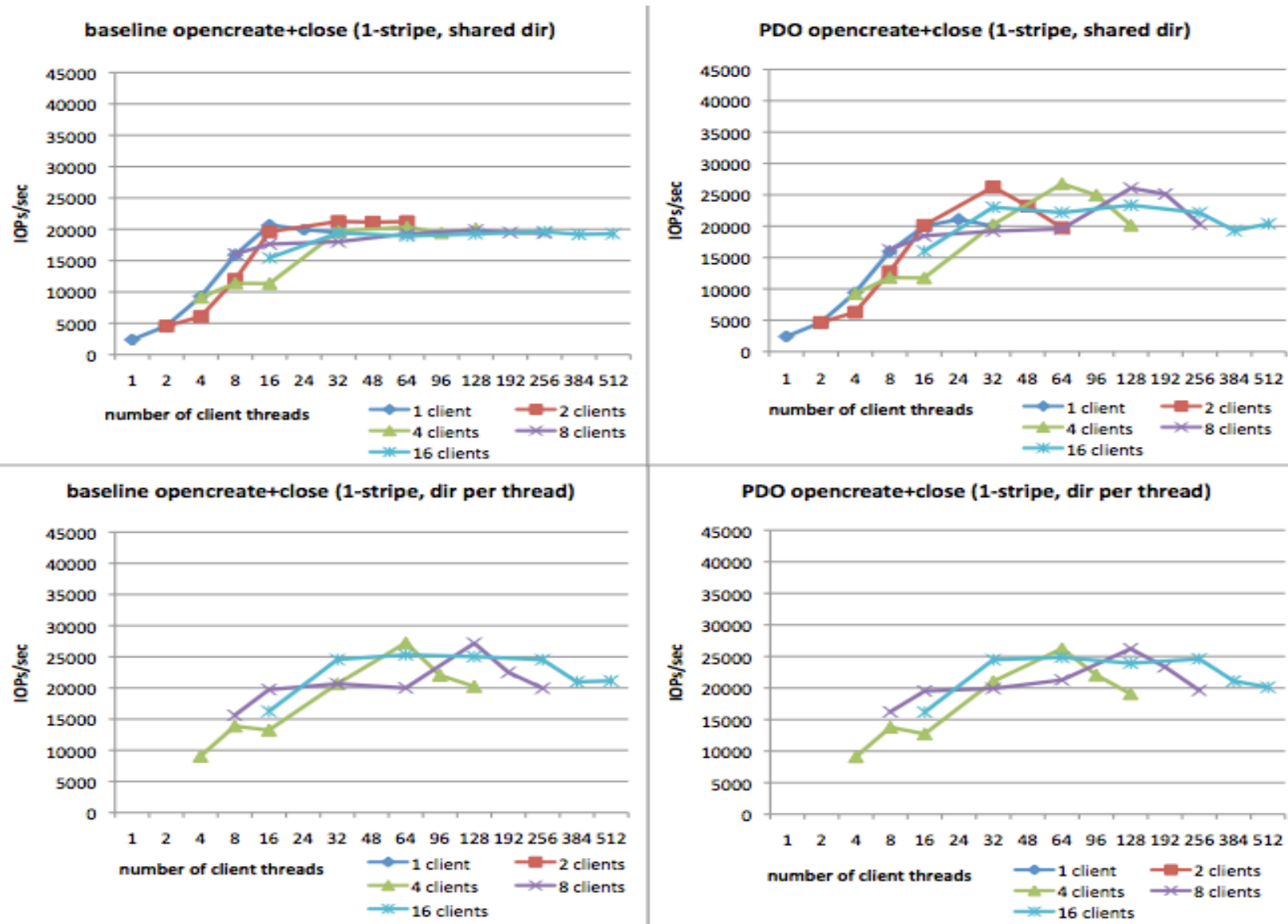
0-stripe opencreate+close performance (1 million files)



1-stripe Opencreate+close performance

- Total 1 million files
- Share directory
 - [1, 2, ..., 16] clients * [1, 2, ..., 32] threads
- Directory per thread
 - [4, 8, 16] clients * [1, 2, ..., 32] threads
- 2 RPCs for opencreate+close
 - PDO project can't help anything for "close"
 - More overhead

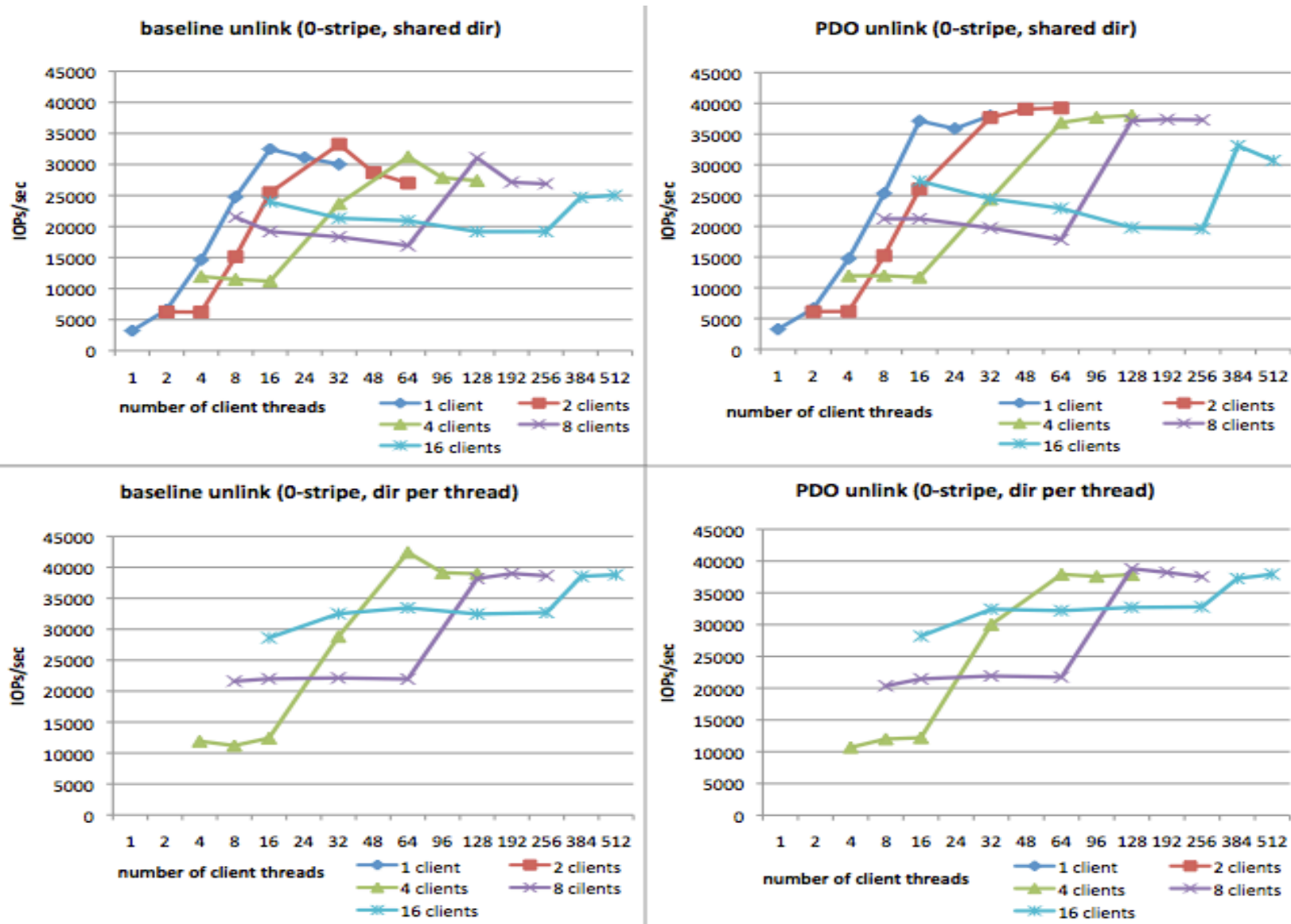
1-stripe opencreate+close performance (1 million files)



0-stripe file unlink performance

- Total 1 million files
- Share directory
 - [1, 2, ..., 16] clients * [1, 2, ..., 32] threads
- Directory per thread
 - [4, 8, 16] clients * [1, 2, ..., 32] threads

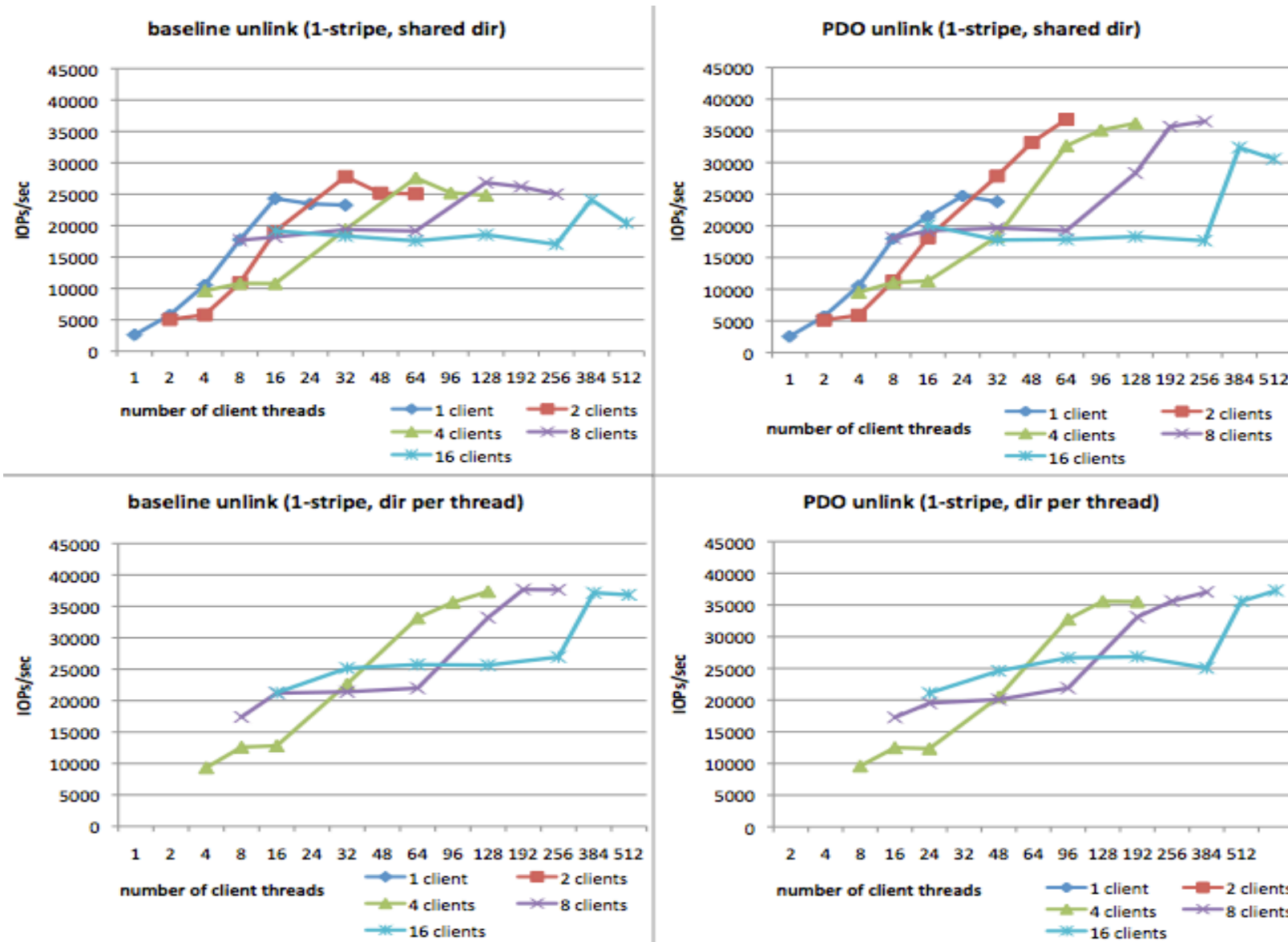
0-stripe file unlink performance (1 million files)



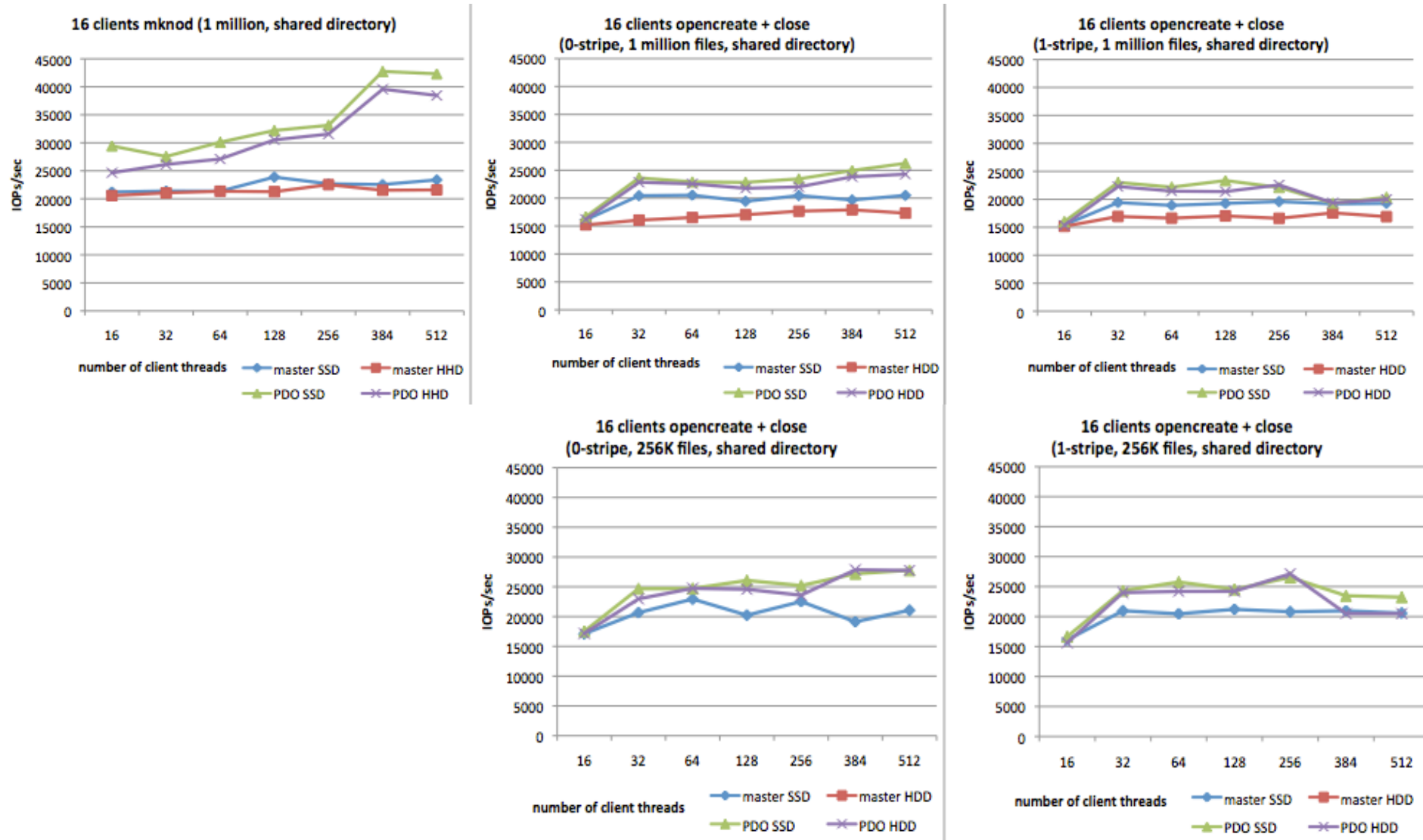
1-stripe file unlink performance

- Total 1 million files
- Share directory
 - [1, 2, ..., 16] clients * [1, 2, ..., 32] threads
- Directory per thread
 - [4, 8, 16] clients * [1, 2, ..., 32] threads

1-stripe file unlink performance (1 million files)



More comparisons: 16 clients creation performance



Summary

- PDO project improved performance of shared directory operations
 - mknod & unlink performance are significantly improved
 - Although there are some strange performance values in unlink tests, which need more survey
 - Opencreate+close is improved somehow
 - Can't show full improvement because of extra "close" RPC
 - Could have some performance issues in striped file, but it shouldn't be in scope of this project.
 - No performance drop for "directory per thread" case
 - Results could be better with SMP improvements



Thank You

- Liang Zhen
Whamcloud, Inc.
liang@whamcloud.com