Simulation study of a distributed metadata server PPMDS

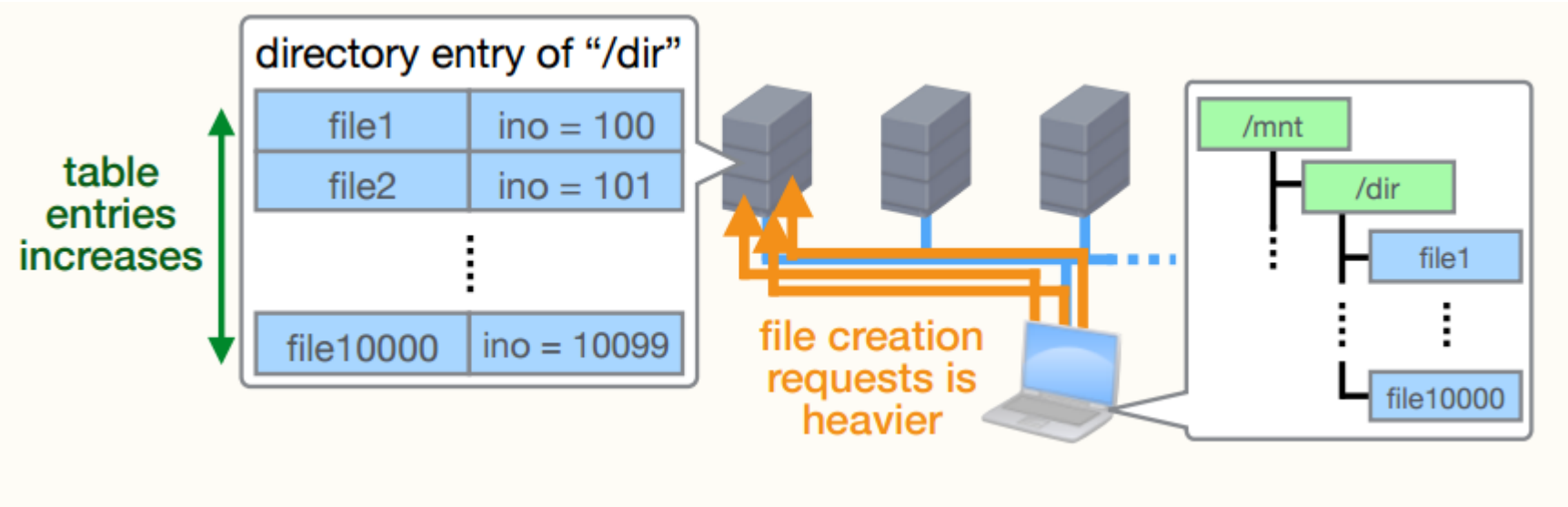
grid team, HPCS lab. University of Tsukuba, Junji Kobayashi
Center for Computational Sciences, University of Tsukuba, Osamu Tatebe

Motivation

- Simulation study of a distributed metadata server PPMDS (2012, Hiraga) using CODES/ROSS
  - Scalability and bottleneck analysis at the large scale environment
- Mdtest benchmark simulation study
  - Up to 1,024 servers, up to 48,000 clients
  - Various directory structure

Background

- Distributed file system manages metadata intensively
- There is no simple way to distribute metadata servers
  - since it manages tree based namespace



  - the performance is limited by synchronization for consistency and serialization

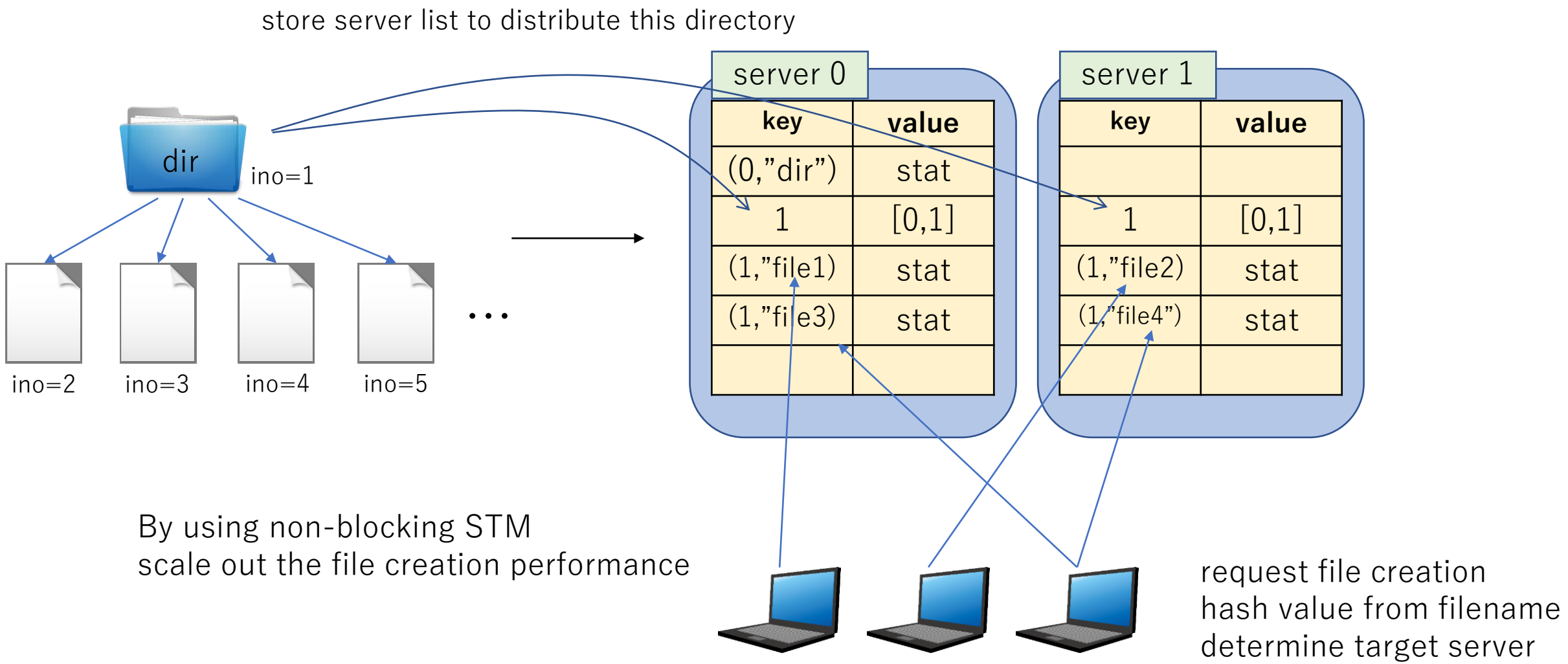
Background

- Metadata server is not easy to scale
  - handling a lot of small files scalably is difficult
- In HPC field …
  - The number of files and nodes continue to grow
  - scalable distributed metadata management server is essential to solve this issue

PPMDS: A distributed metadata management system

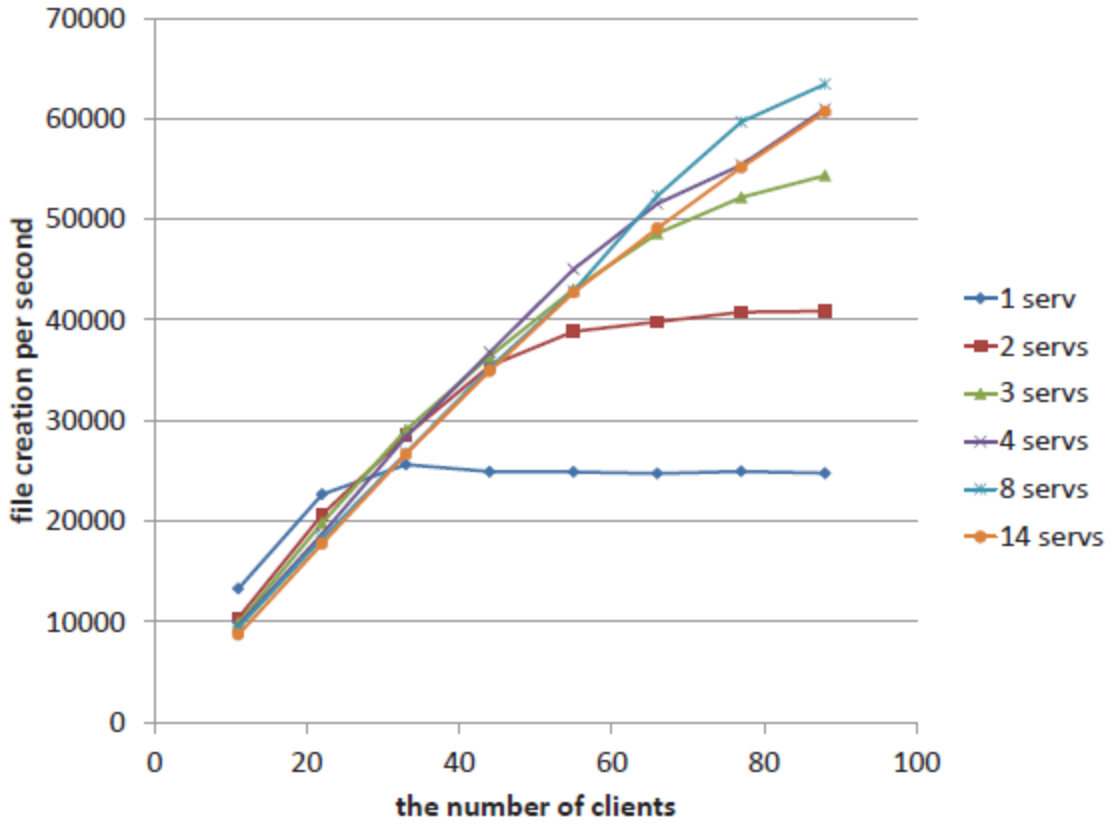
- PPMDS(2012, Hiraga) is a distributed metadata server
- Feature
  - **scalable**
    - shared nothing Key-value stores
  - **consistent**
    - distributed transaction based on non-blocking STM
- These features have enabled
  - highly parallel read/write/delete accesses to a single directory

PPMDS: A distributed metadata management system



PPMDS: A distributed metadata management system

- result of file creation performance at a single directory using 1~14 metadata servers and 1~88 clients



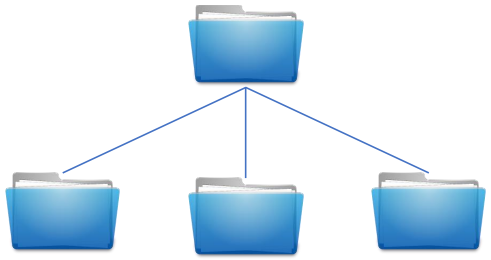
mdtest: metadata benckmark test

- mdtest is an MPI-coordinated metadata benchmark test
  - reports the performance of crate/stat/remove operations on files and directories

- Tasks in mdtest create directory trees and each task creates files and directories in the directory tree.

mdtest: metadata benckmark test

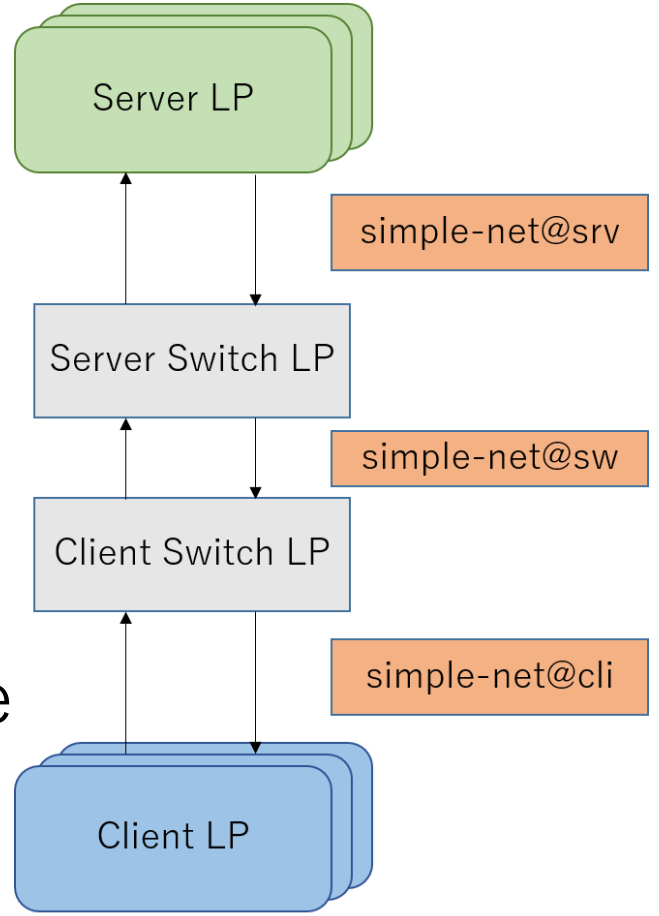
- how mdtest operates
  - $ mpirun –np $CNT mdtest –z 1 –b 3 –I 100 -u –d /tmp/testing
  - Each task creates a directory tree in /tmp/testing
  - Each tree has a depth of 1 and a branching factor of 3
  - 100 files/dirs are operated upon in each node of each tree.



Hierarchical directory structure (tree)
depth=1, branching factor=3

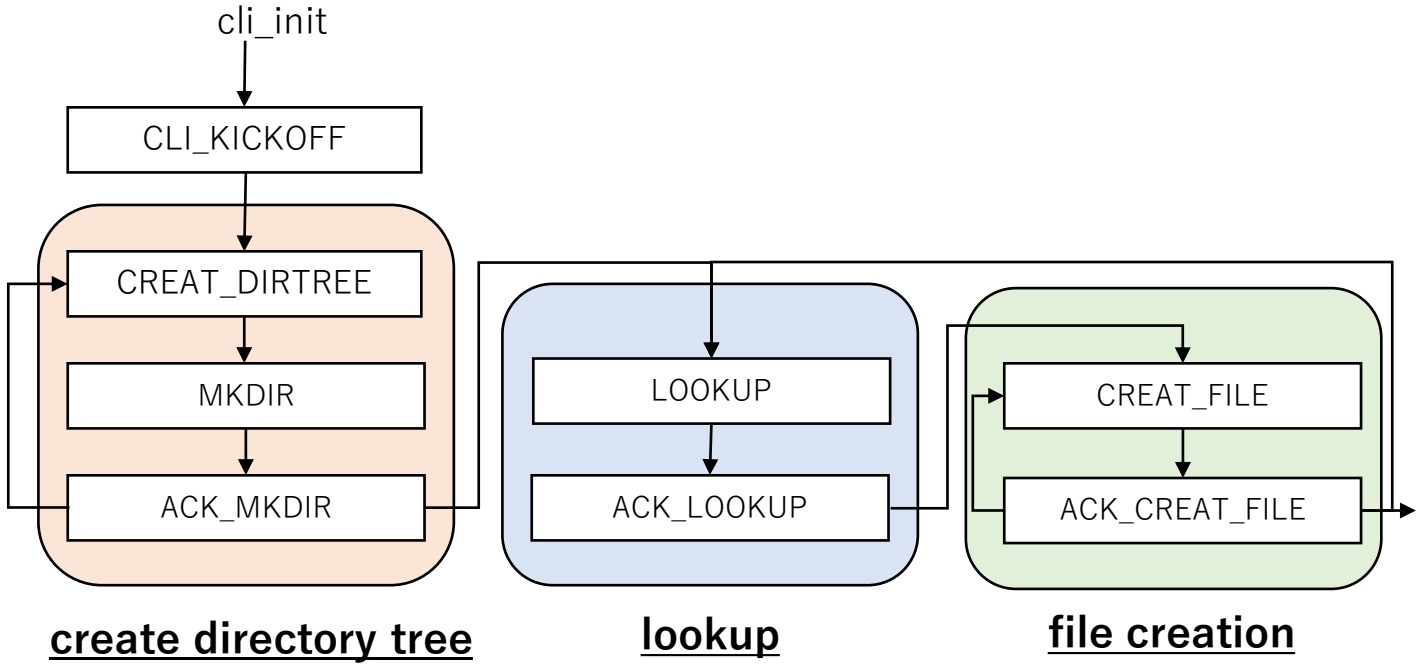
Simulation of a benchmark for PPMDS

- I implement simulation of benchmark for PPMDS
- each Client LP create directory tree like mdtest
- Client LP request Server LP to create/stat/remove file to the directory tree



Simulation of a benchmark for PPMDS

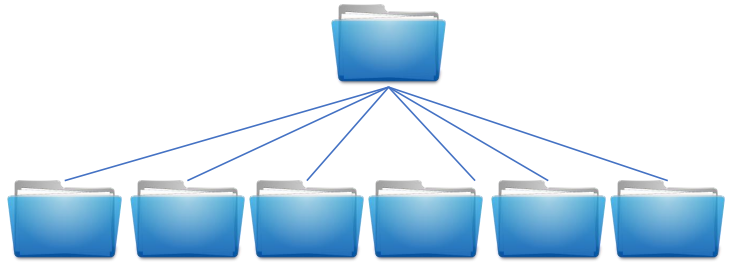
- Simulate file creation
- create directory tree
  - depth/width defined by the value in the configuration file.
- lookup target directory
  - cache inode
- request file creation to target Server LP



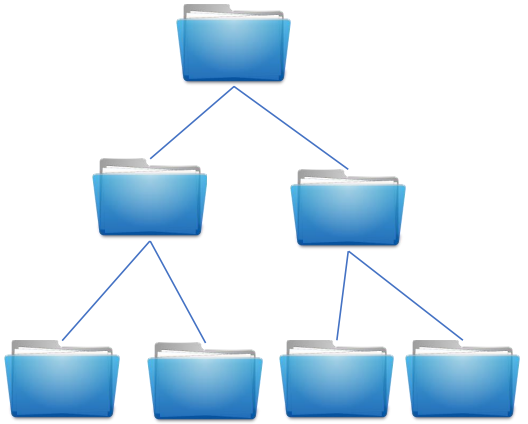
Simulation result

- Evaluate file creation maximam performance of PPMDS
  when the number of servers is 1 ~ 1,024
  increasing the number of clients 1~48,000

- Clients create directory tree
  (depth=2, width=2, 7 directories)
  (depth=1, width=6, 7 directories)



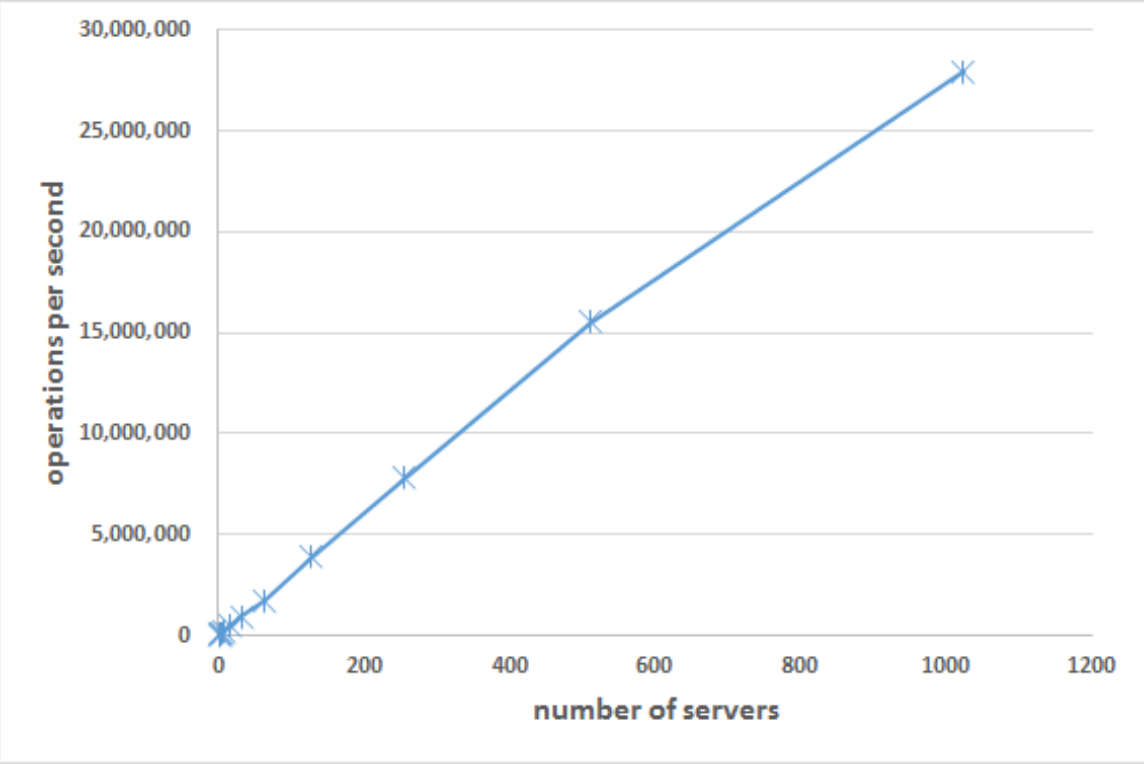
Hierarchical directory structure (tree)
depth=1, branching factor=6



Hierarchical directory structure
depth=2, branching factor=2

Simulation result

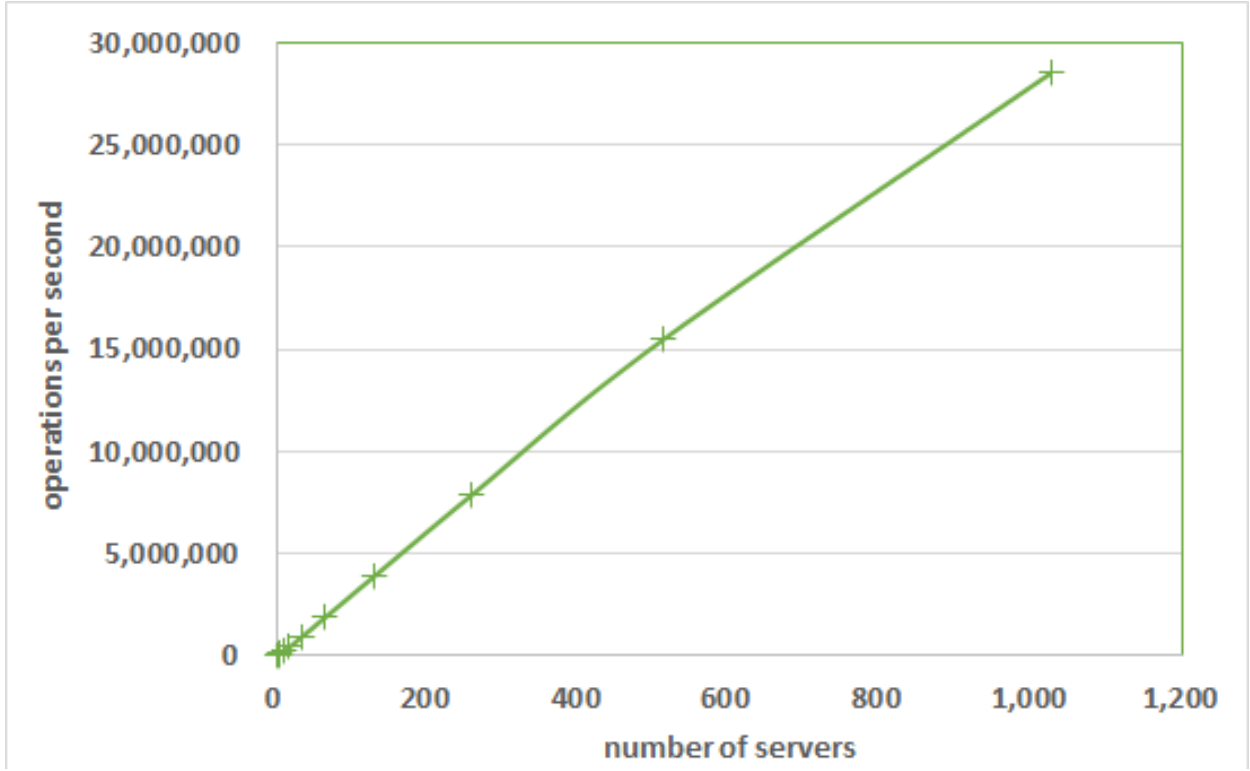
- Each Client LP create files to this directory tree like mdtest.
  - mdtest –z 2 –b 2 …
  - mdtest –z 1 –b 6 …
- Running the simulator in optimistic mode



simulation result of file creation (depth=2, width=2)
1~1024 servers, 1~48000 clients.

Simulation result

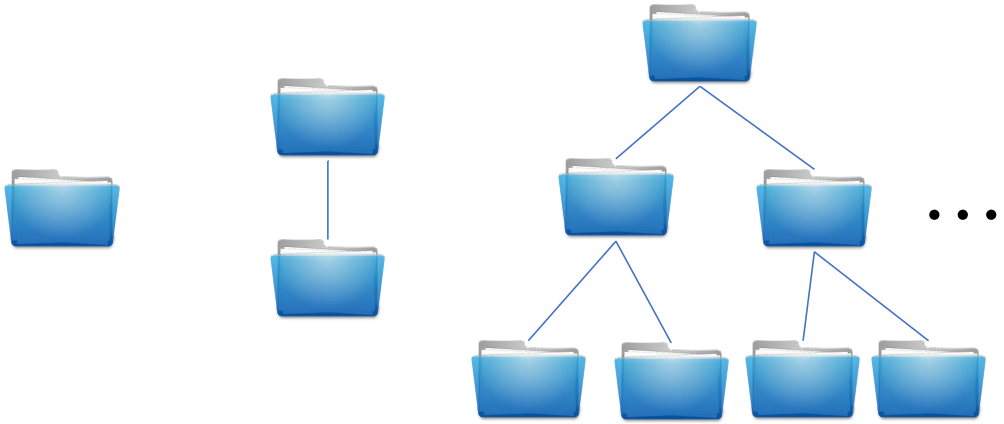
- Both simulation results show good scalability on file creation operation per second regardless of directory structure
- From this result, it appears that PPMDS is scalable regardless of directory structure



simulation result of file creation (depth=1, width=6)
1~1024 servers, 1~48000 clients

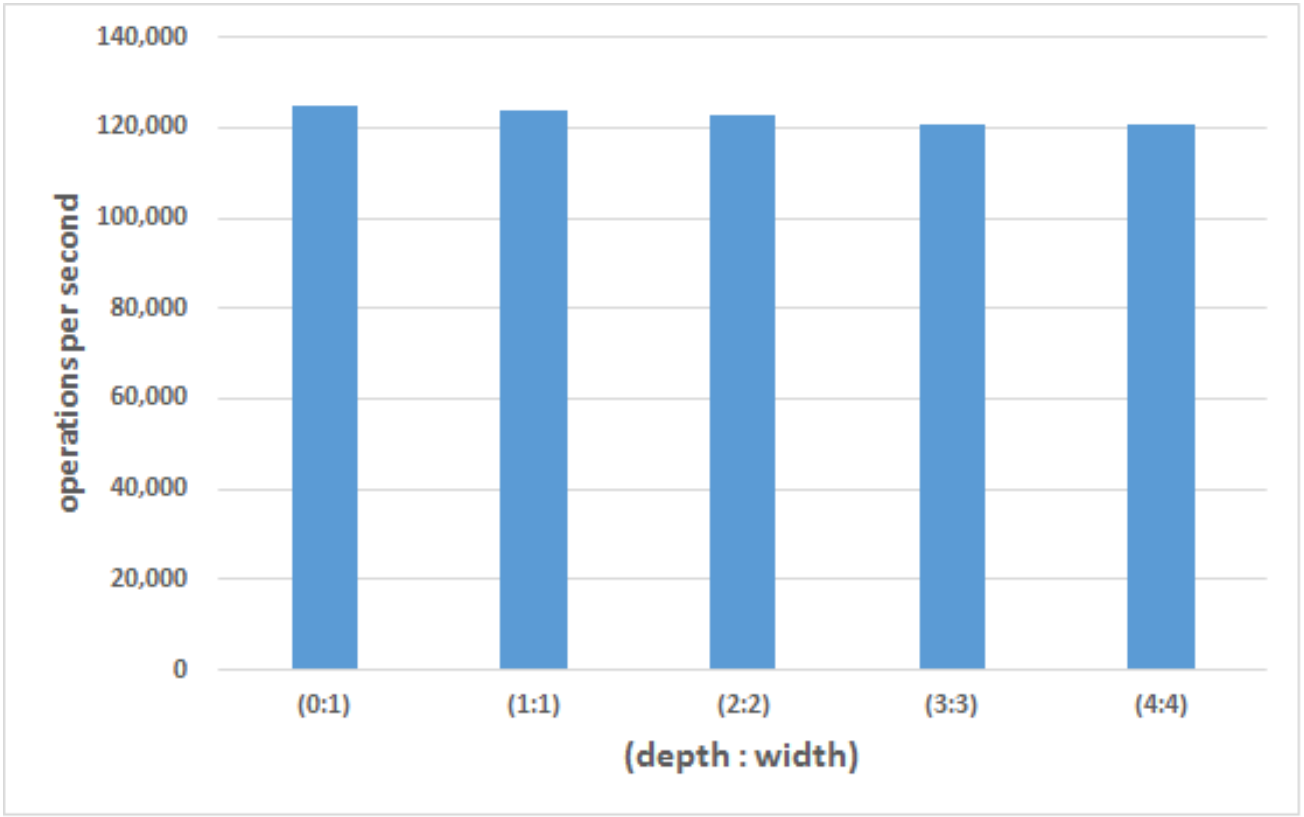
Simulation result

- Simulate file creation performance to various directory structures (16 servers, 88 clients)



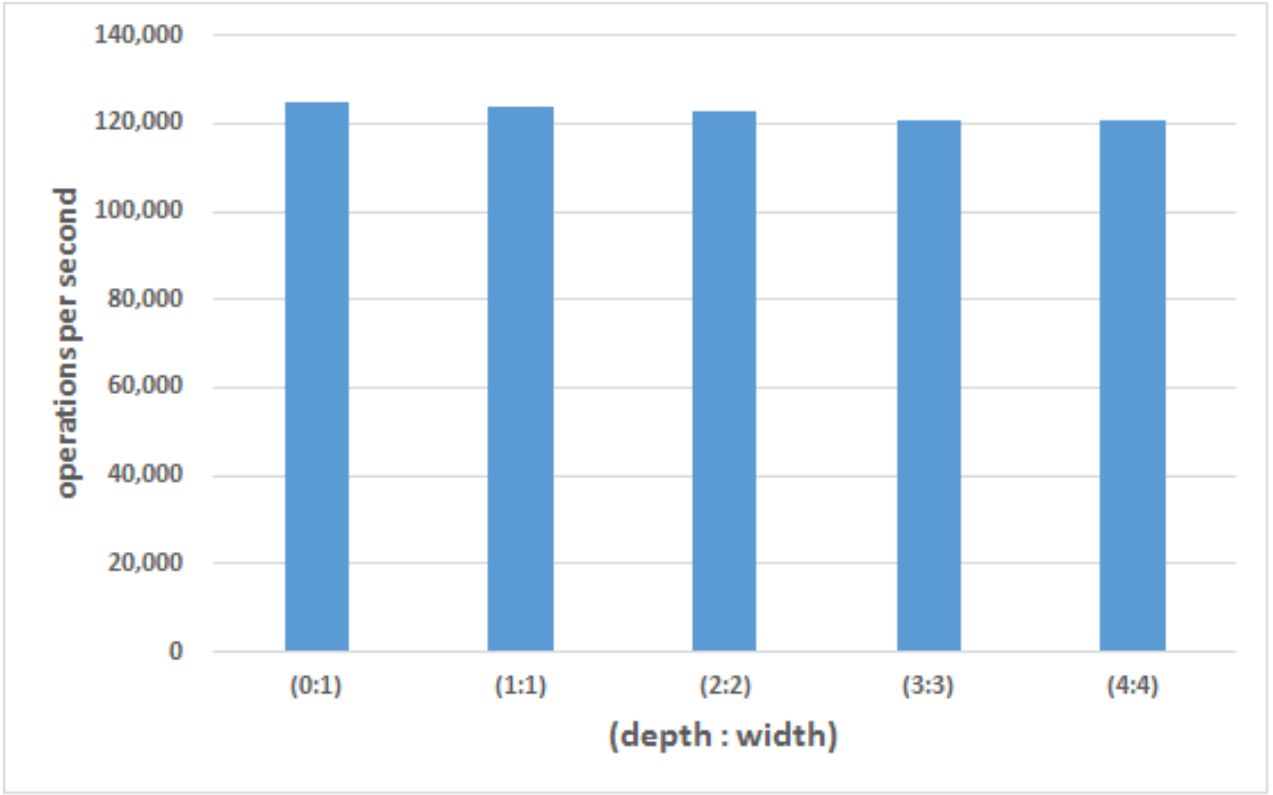
(0:1) (1:1) (2:2)

(depth: width)



Simulation result

- From this result, it appears that file creation performance in PPMDS not affected by directory structure.



Next Steps

- PPMDS simulator supports only metadata operations,
  It can not simulate as storage server.

- In order to evaluate as storage server, it is required to add
  Storege Server LP to PPMDS simulator.