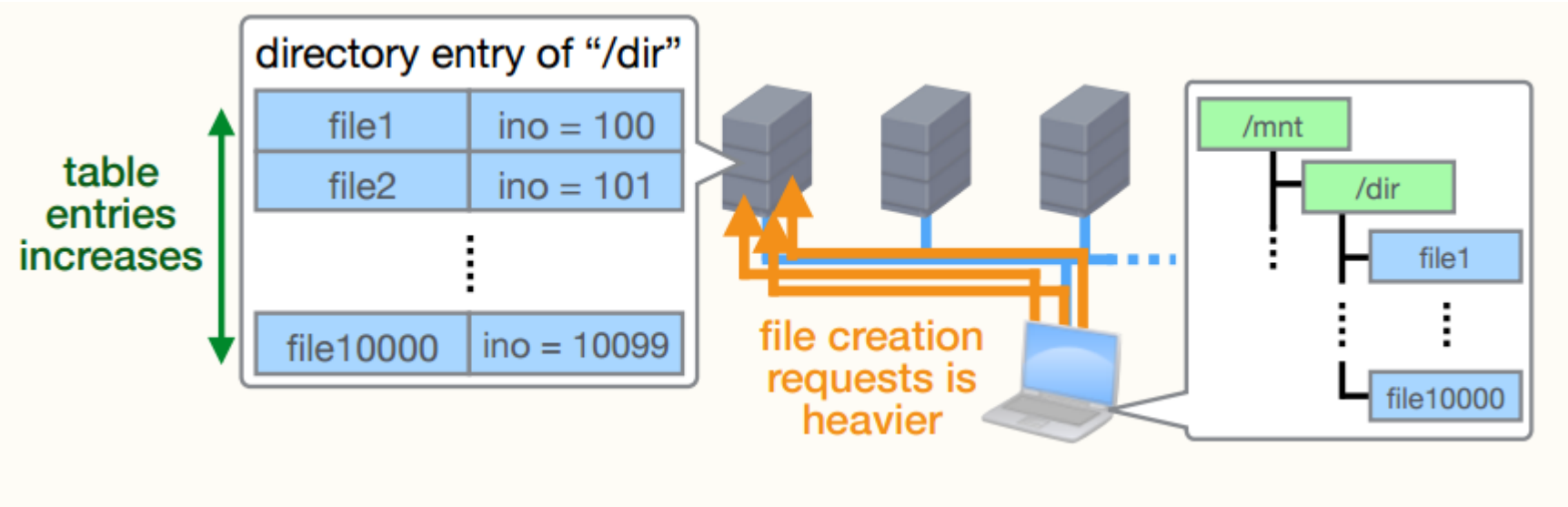
Simulation study of distributed metadata server

Junji Kobayashi, Osamu Tatebe
University of Tsukuba
HPCS Lab.

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

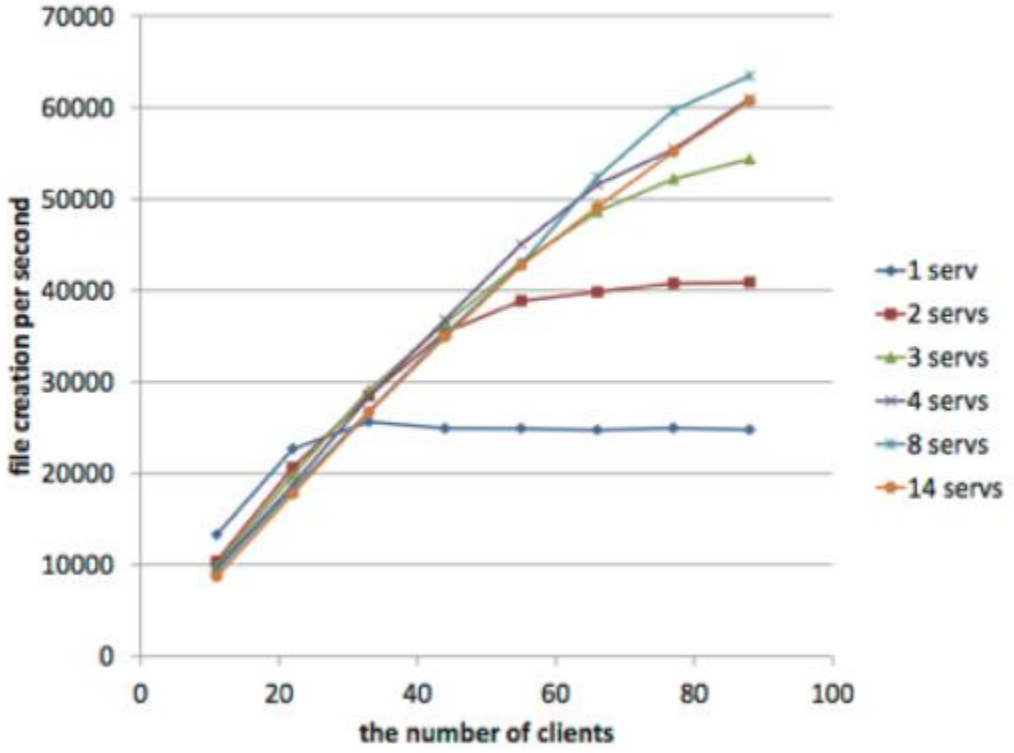
- Metadata server is not easy to scale
  - handling a lot of small files scalably is difficult
- In HPC field …
  - The number of files and nodes continue to grow
  - scalable distributed metadata management server is essential to solve this issue

PPMDS: A distributed metadata management system

- 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

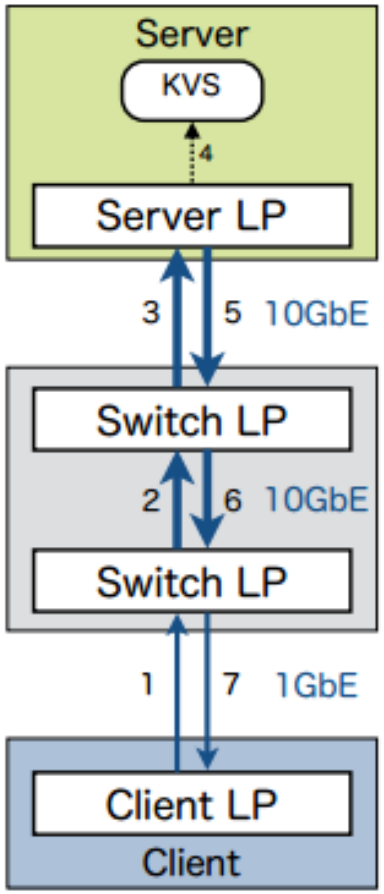
PPMDS: A distributed metadata management system

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



Simulation of PPMDS: PPMDSsim

- Simulate clients & servers using CODES/ROSS
- Client LP supports *file_create,file_stat,file_removal*
- Server LP represents server nodes. Each Server LP has Key-Value-Store and supports load/store inode entry
- Switch LP represents network switches



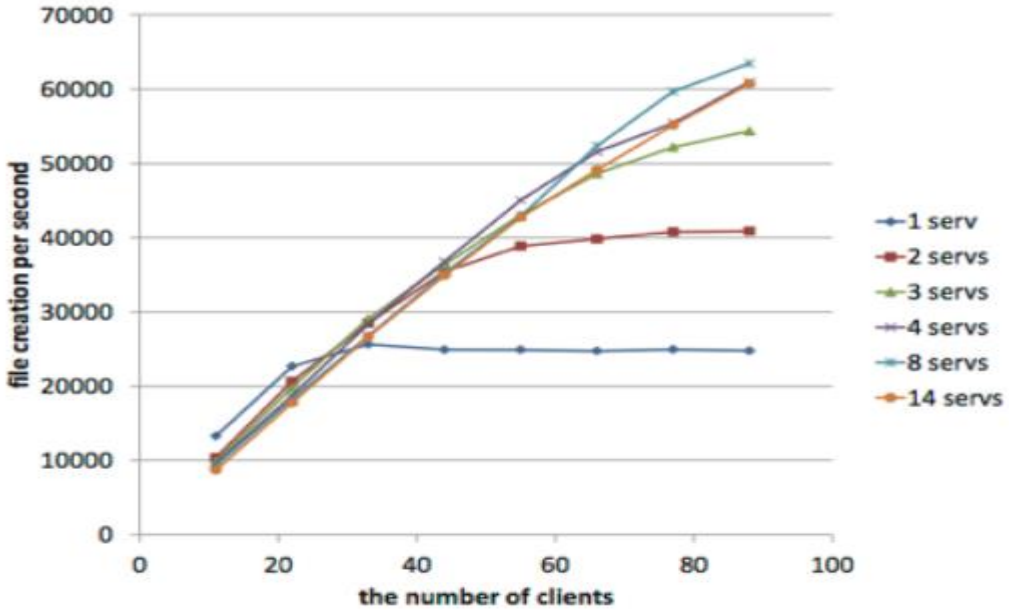
Simulation of PPMDS: PPMDSsim

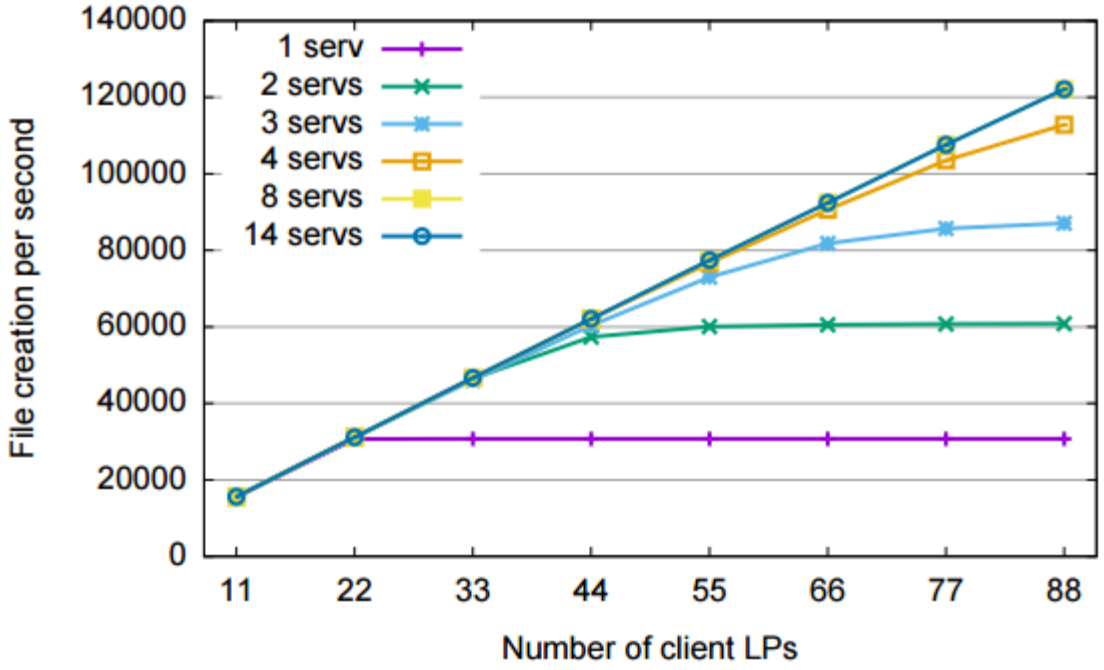
- Configurable parameters in PPMDSsim

parameter	**value**	
packet_size	1,500	network packet size
net_startup_ns	32,000	network startup latency
net_bw_mbps@cli	125	network bandwidth between client-switch(MB/sec)
net_bw_mbps@srv	1250	network bandwidth between server-switch(MB/sec)
net_bw_mbps@sw	1250	network bandwidth between switch-switch(MB/sec)
payload_size	512	message transmission payload size (byte)
store_inode_event_latency	500,000	transmission latency of the message that inode entry is stored (ns)

Simulation result of parallel file creation to a single directory

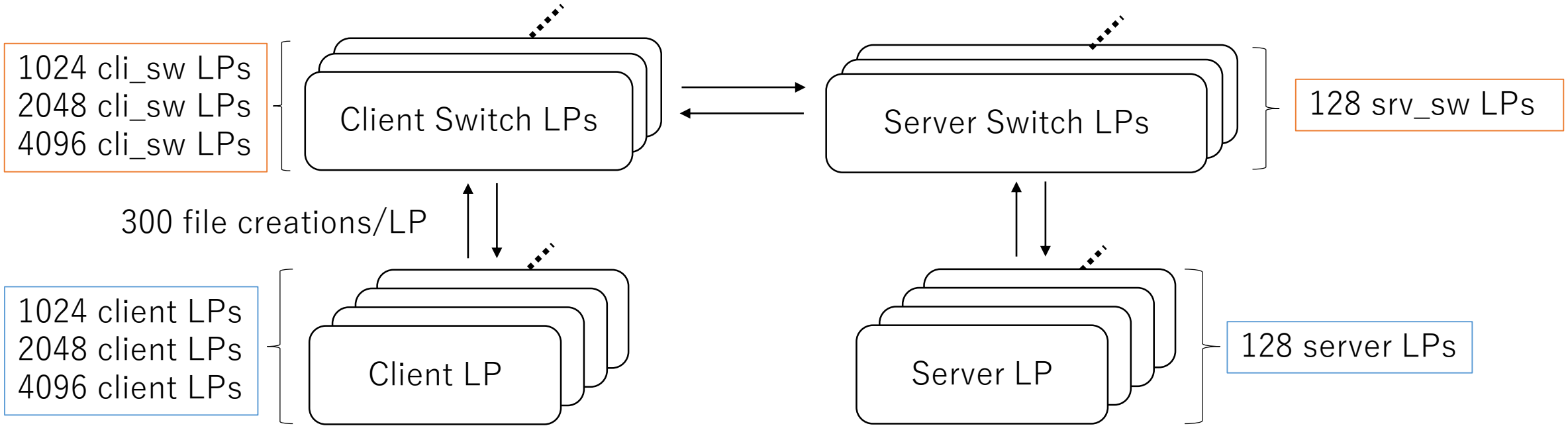
- 1 ~ 14 server LP nodes and 11 ~ 88 client LP nodes
- The result of the simulation shows a similar performance behavior to a real system





Performance of parallel simulation

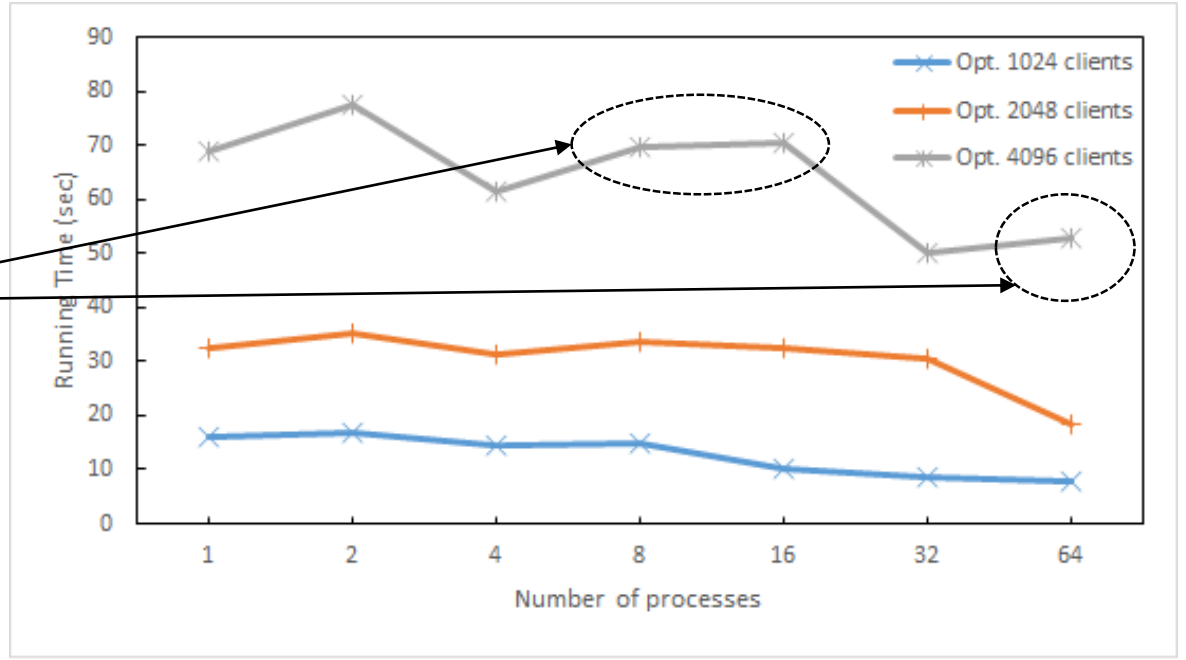
- Measure the running time of parallel simulation with 128 server LP nodes and client LP per file creation = 300 (Synchronization Protocol : optimistic)



Performance of parallel simulation

- There is a case such that the running time increases even when the number of processes increases

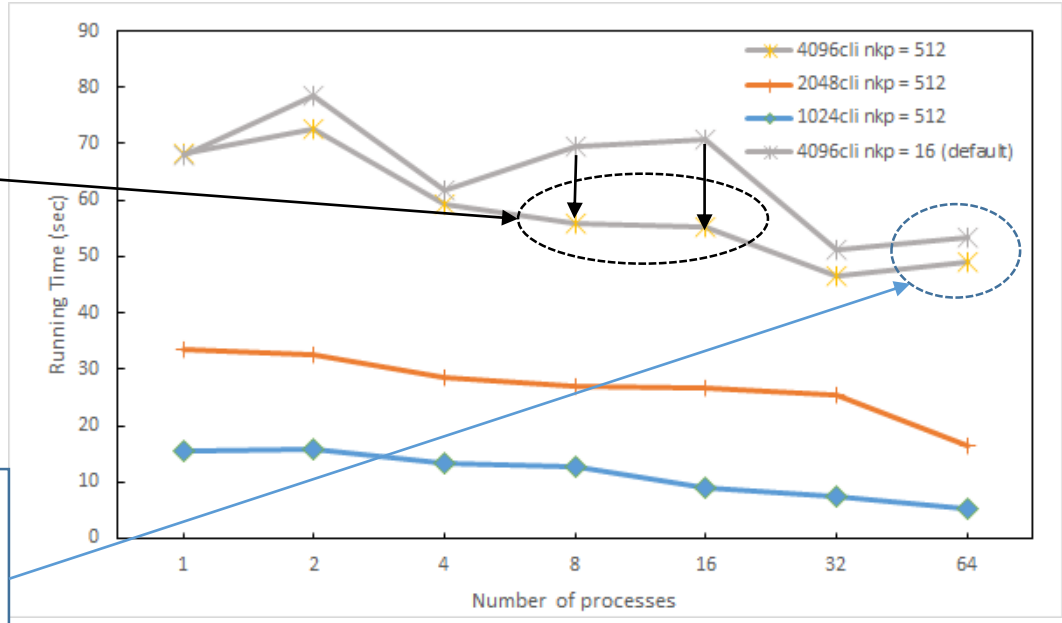
Even when the number of processes increases, running time does not always reduce



Performance of parallel simulation

- To improve the performance of parallel simulation
  - ROSS supports a number of parameters that effect optimistic mode performance such as batch, gvt_interval, KP count …
- When the global variable g_tw_nkp is set to 512 from 16, the performance slightly improves

The performance is improved in case of 8 and 16 processes



- We would like to know any suggestion to improve the performance further

The performance was not improved in case of 64 processors due to insufficient number of server LP nodes.

Next Steps

- Improvement of the simulation accuracy
  - The result of file-creation simulation shows similar performance behavior, but the performance number is not same
- Support of directory operations
  - Directory operation requires complex operations including a transaction across several servers
  - Simulation study of several access patterns is required