

Summer of CODES Workshop, 2015

## **Simulation of PPMDS:**

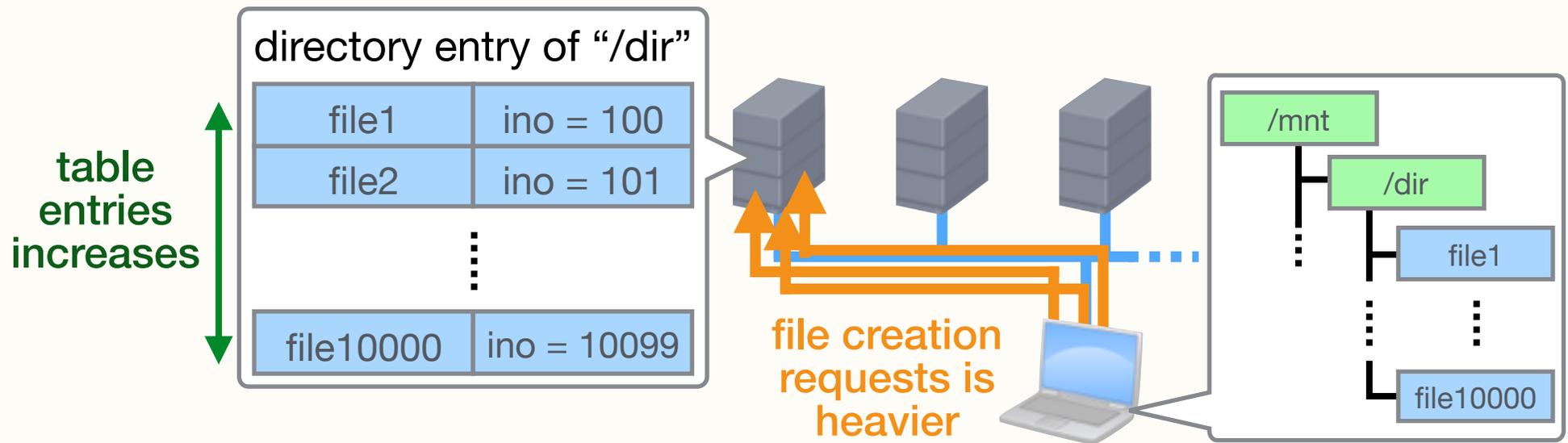
# **A Distributed Metadata Management System**

**Yuki Kirii, Hiroki Ohtsuji, Kohei Hiraga, Osamu Tatebe**  
**High Performance Computing System Laboratory,**  
**University of Tsukuba**

# Background

- Distributed file system manages metadata intensively
- No simple way to distribute metadata servers

## 1. it manages tree based namespaces



## 2. its performance is limited by synchronization for consistency and serialization

# Background

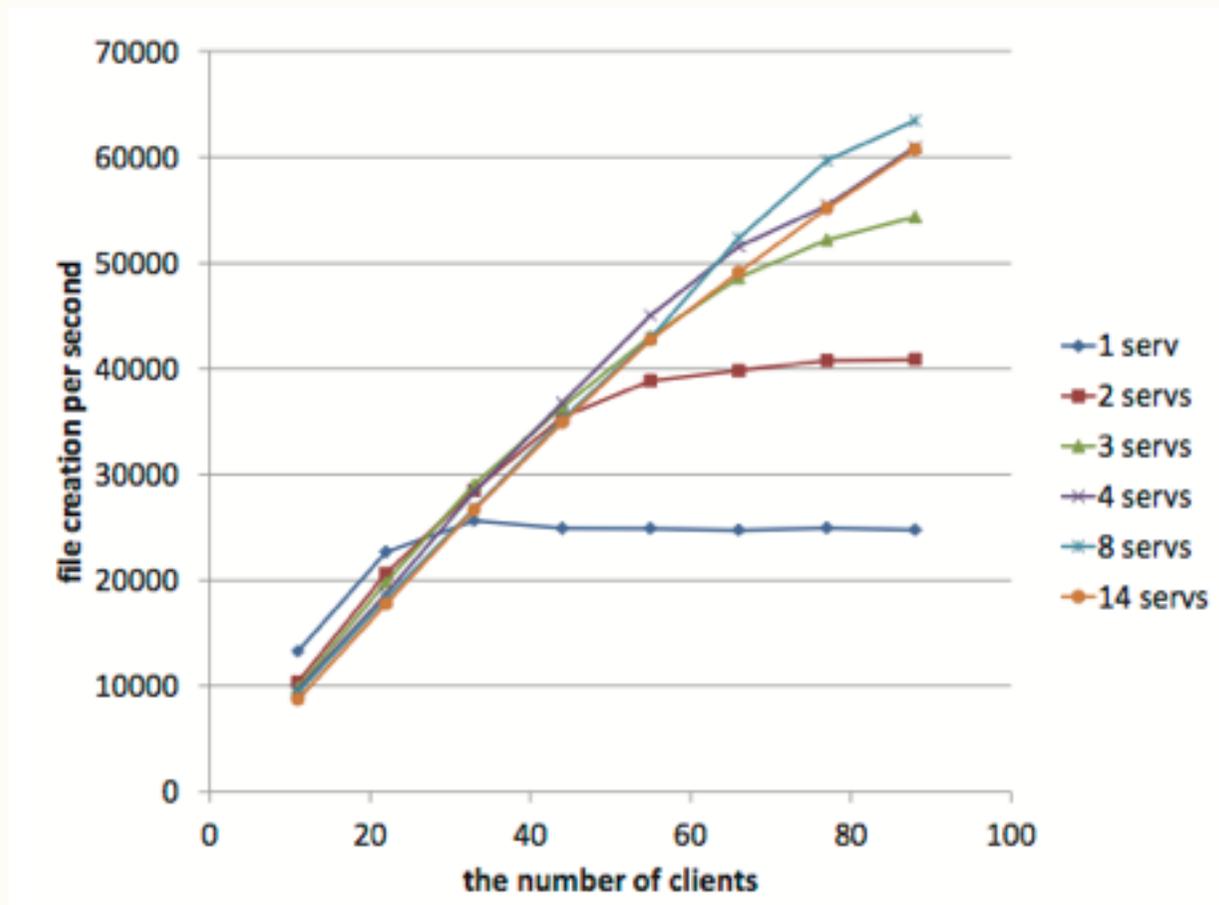
- ➔ Metadata server is not easy to be scalable
  - handling a lot of small files scalably is a difficult one
- In HPC field ...
  - quantities of files and nodes continues to grow
    - ➔ scale-out of distributed metadata management server is essential to support this

# PPMDS: A distributed metadata management system

- Features
  - ▶ Scalable
    - shared nothing Key-Value stores
  - ▶ Consistent
    - distributed transaction based on non-blocking STM
- These features have enabled
  - ➔ highly parallel read/write/delete access to a single directory

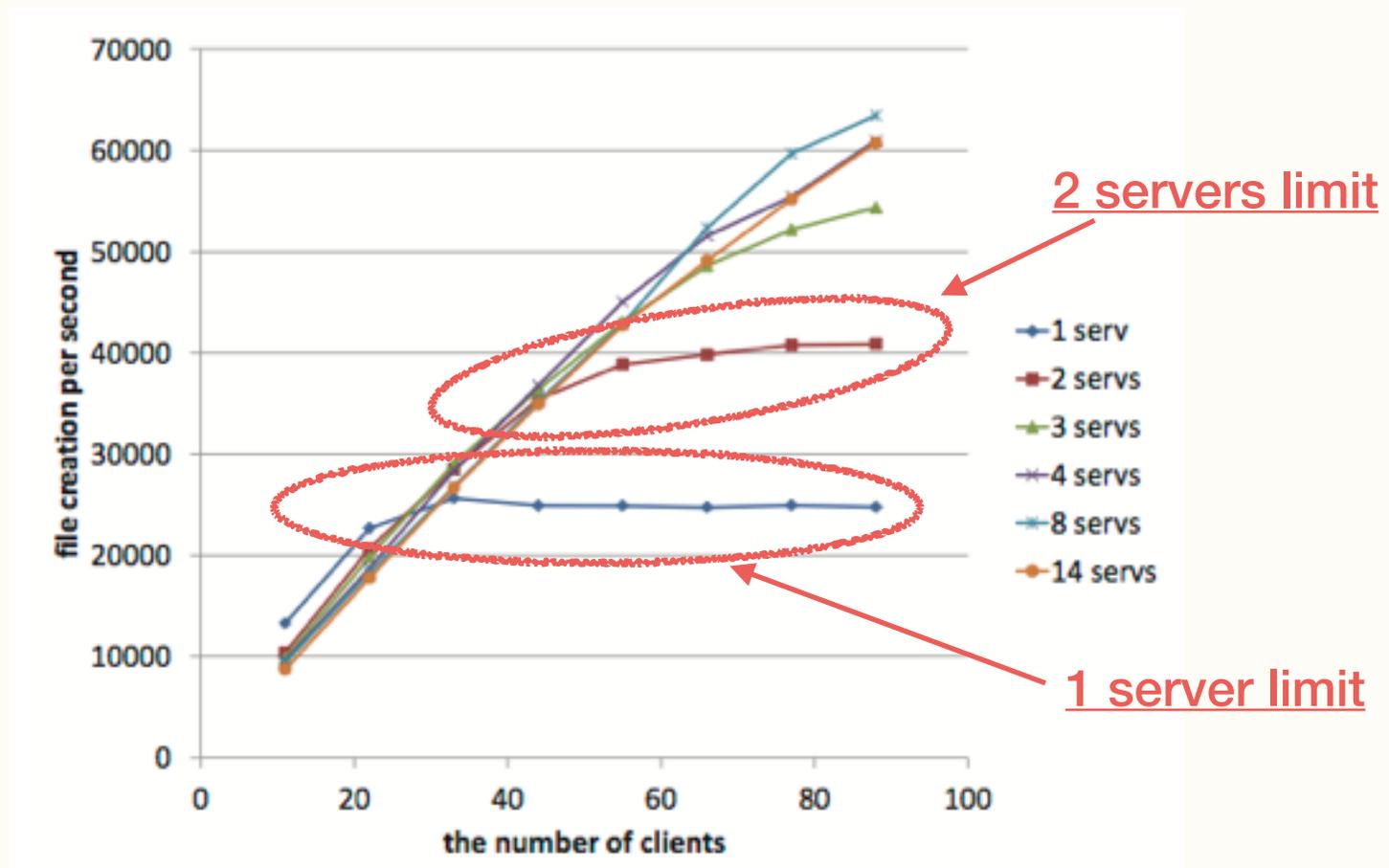
# PPMDS: A distributed metadata management system

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



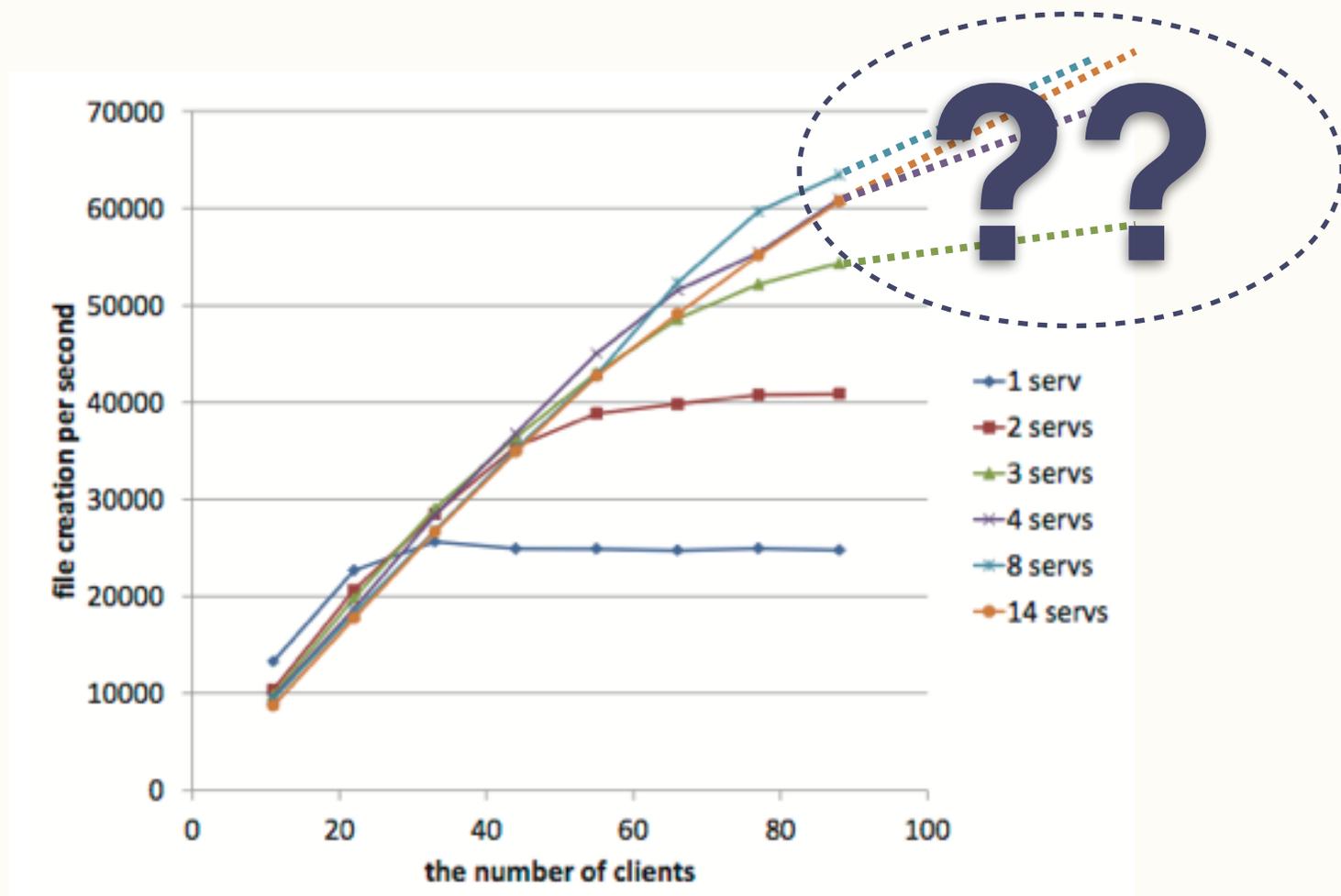
# PPMDS: A distributed metadata management system

- When the number of servers is 1 or 2
- 88 clients saturate servers performance



# PPMDS: A distributed metadata management system

- 88 clients did not saturate servers performance
- we couldn't get enough data to evaluate its scalability



# Simulation of PPMDS

- Simulate clients & servers using CODES/ROSS
- Our focus :

## 1. How the performance scales out

- If the performance does not scales out, we need to find out why

## 2. Compare with related distributed file systems

- ▶ GIGA+
- ▶ TABLEFS
- ▶ Batch FS
- ▶ Coda
- Adopt similar workloads of PPMDS using quantitative approach

# To study CODES/ROSS

- I implement simulations of Chord algorithm based DHT
  - construct DHT network and put Key-Value pairs
- I'm looking for a best way to exchange data between LPs
  - I have used the message of LP's events as arguments or return values...

```
typedef struct node_msg
{
    int params_size;
    int params[MAX_PARAMS_SIZE];
}
```

- If anyone knows better ways, please advise me!