Metadata Load Balance:

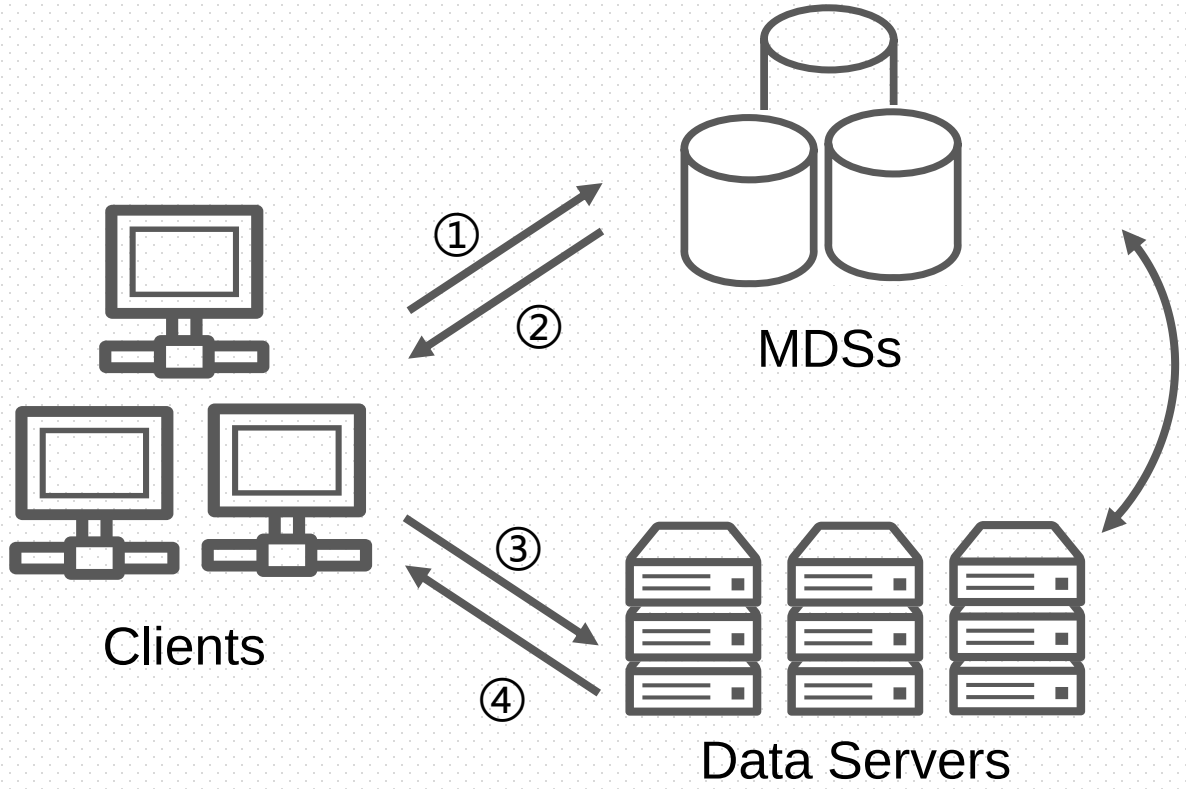
Validation, Modeling, Solution

Shared by Yiduo Wang, Daniel Shao

2020/07/16

Metadata in CephFS

- Necessity
  - Perform metadata ops first
- Account
  - More than **50%**
- Overhead
  - Exceeds Data ops on **lots of small files**



Access process of CephFS

Multi-MDS Development

- Single MDS is insufficient
- Hash partition will destroy the locality
- Dynamic adjusting namespace relevant performance

Ceph Metadata Balance Flow

1. Load statistics
   - Frag's popularity & System info
2. Role determination
   - Select part MDS as exporter
   - Amount: My_load – Avg_load
3. Directory selection
   - Select part hot fragments
   - Hot: Accessed amount(decayed) is high
4. Directory export
   - Migrating selected fragments
   - Irrevocability

Measuring load imbalance

- Using coefficient of variation as imbalance factor
- Definition:
  - n: MDS number
  - l_i: load of MDS_i
  - $\bar{l}$: Avg load
  - I: imbalance factor

$$I = \frac{\sqrt{\sum_{i=1}^{n}(l_i - \bar{l})^2/(n-1)}}{\sum_{i=1}^{n} l_i/\mathrm{n}}$$

Workload Classification

- AI pre-training
  - Scanning, flat directory structure
- Tar Linux kernel
  - Scanning, complex directory structure
- Zipfian
  - Skewed access, flat directory structure
- Web Access
  - Skewed access, complex directory structure
- Compilation
  - Skewed access, complex directory structure, with data ops & computing

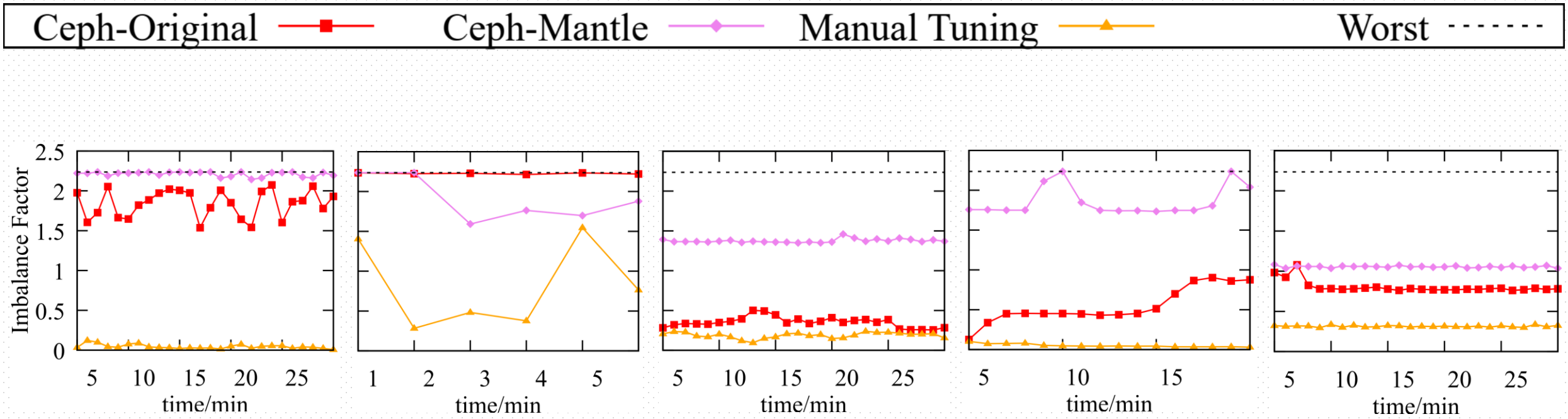
Policy Classification

- Ceph-Original
  - Native Ceph migration strategy + Default parameters
- Ceph-Mantle
  - Mantle enabled + Greedyspill.lua policy
- Dynamic hashing
  - Hashing hot fragments across MDSs
- Manual Tuning
  - Manual pin or hash namespace across cluster
  - Without migration

Experiment Setup

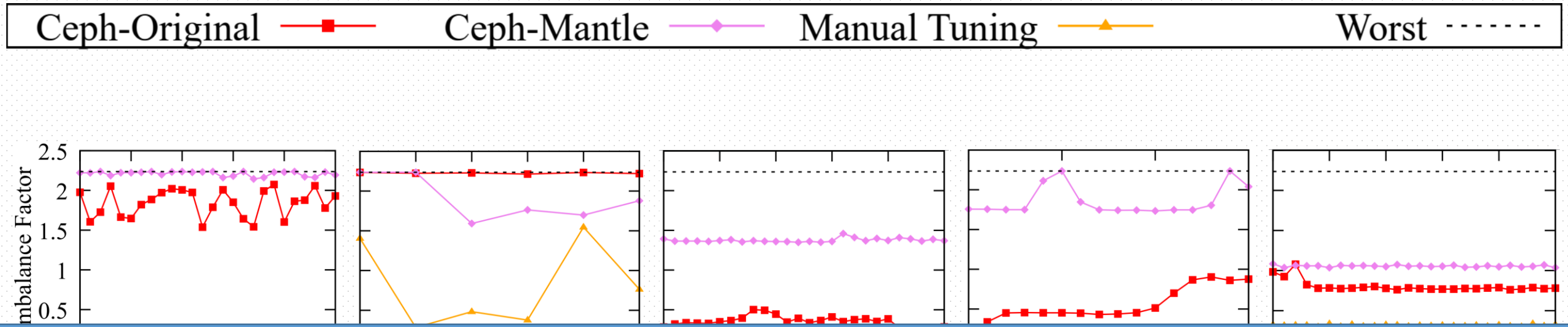
Metadata nodes	5
Client nodes	5
Metadata servers	5
Client threads	100
OS Kernel	CentOS 7.6.1810
CPU	Intel E5-2650 v4 @ 2.20GHz
Memory	64GB
Network	56Gbps InfinitiBand

Evaluation: Imbalance Factor



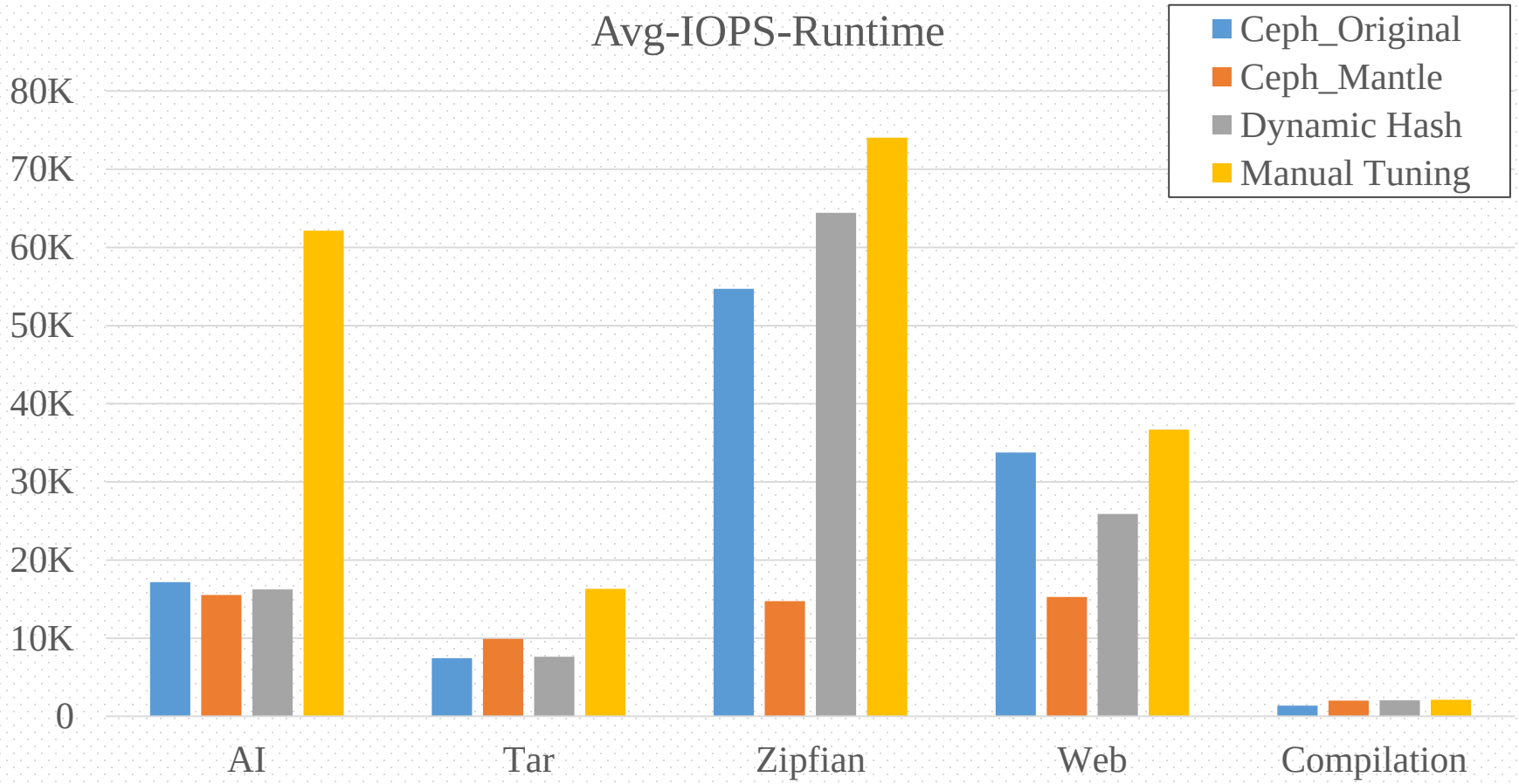
AI pre-training | Tar | Web Access | Zipfian | Compilation

Evaluation: Imbalance Factor

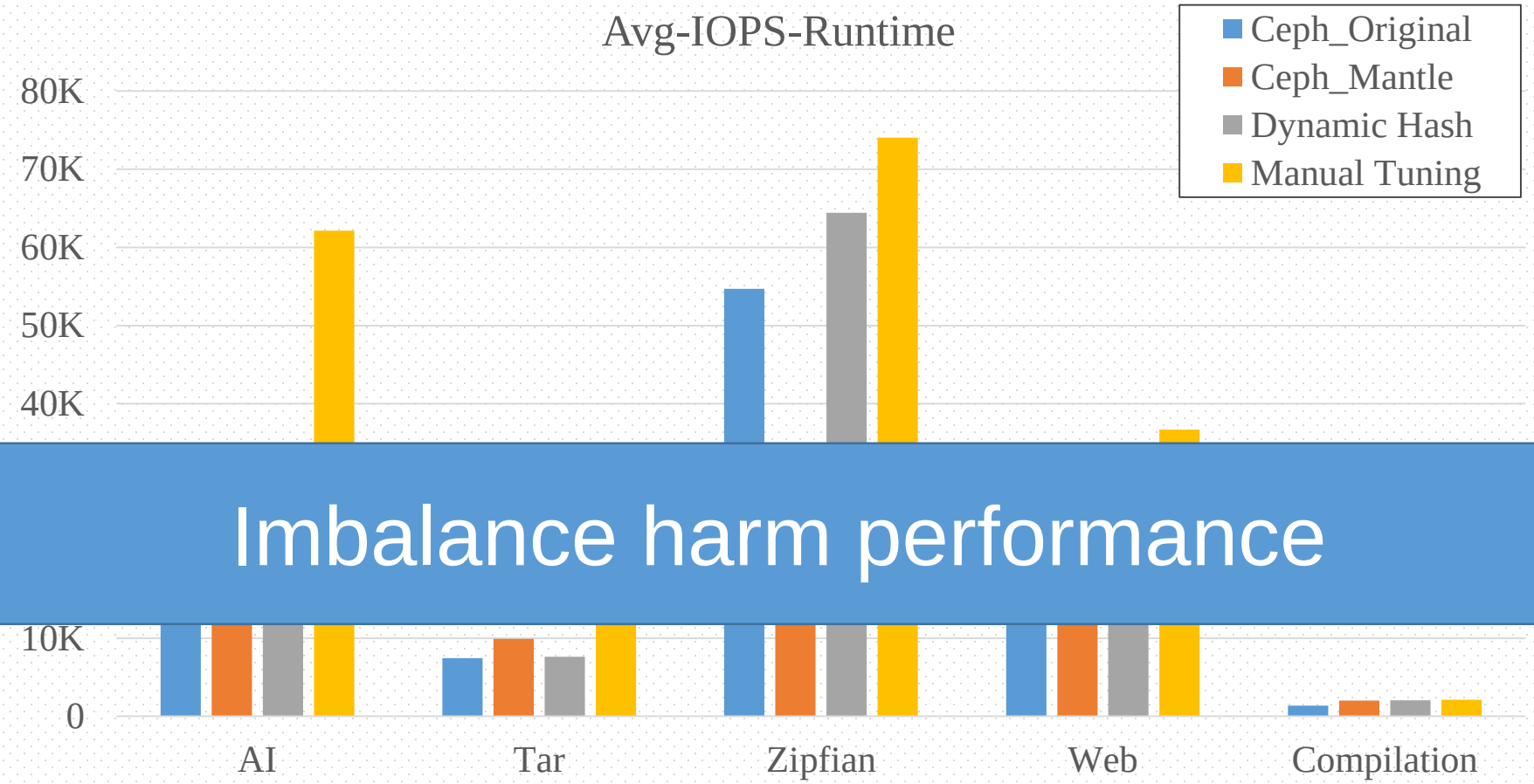


Metadata imbalance is existed & common

Evaluation: IOPS



Evaluation: IOPS



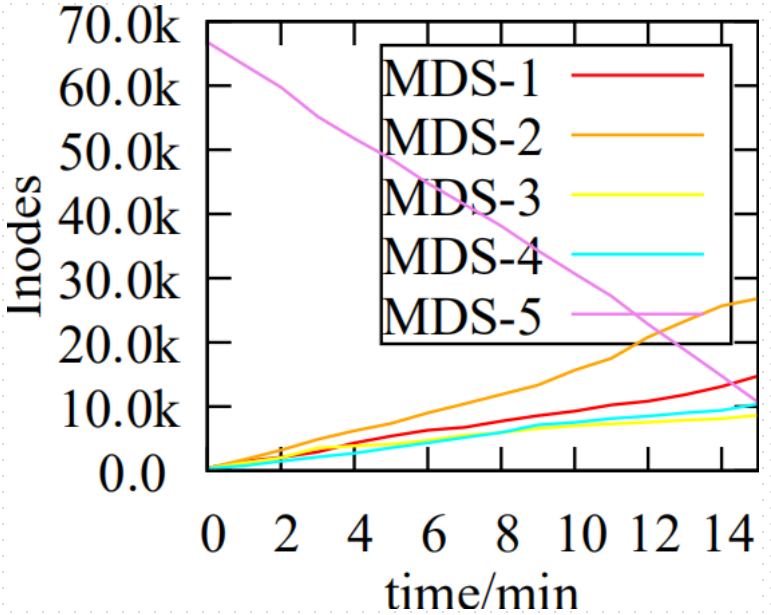
Imbalance harm performance

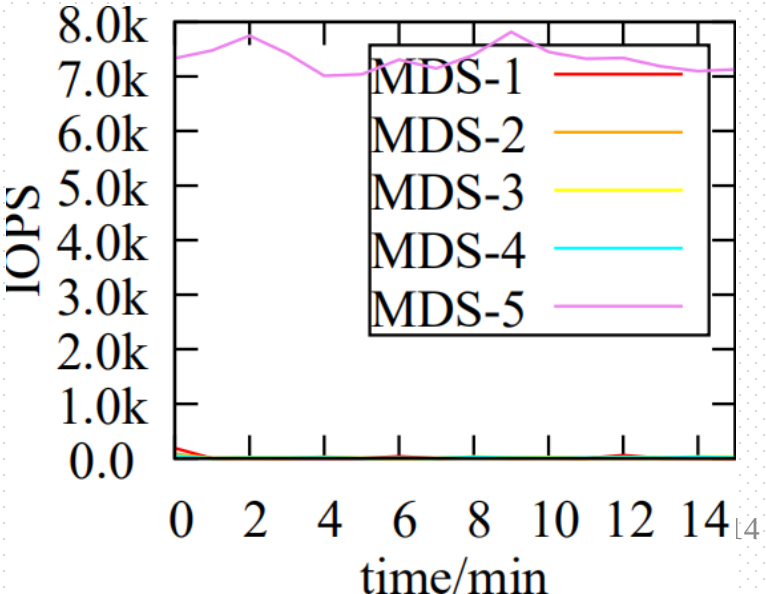
Analysis

- Why does the one-for-all policy does not fit well with four workloads?
  - AI pre-processing
  - Tar
  - Zipfian
  - Web access

Case study: Tar

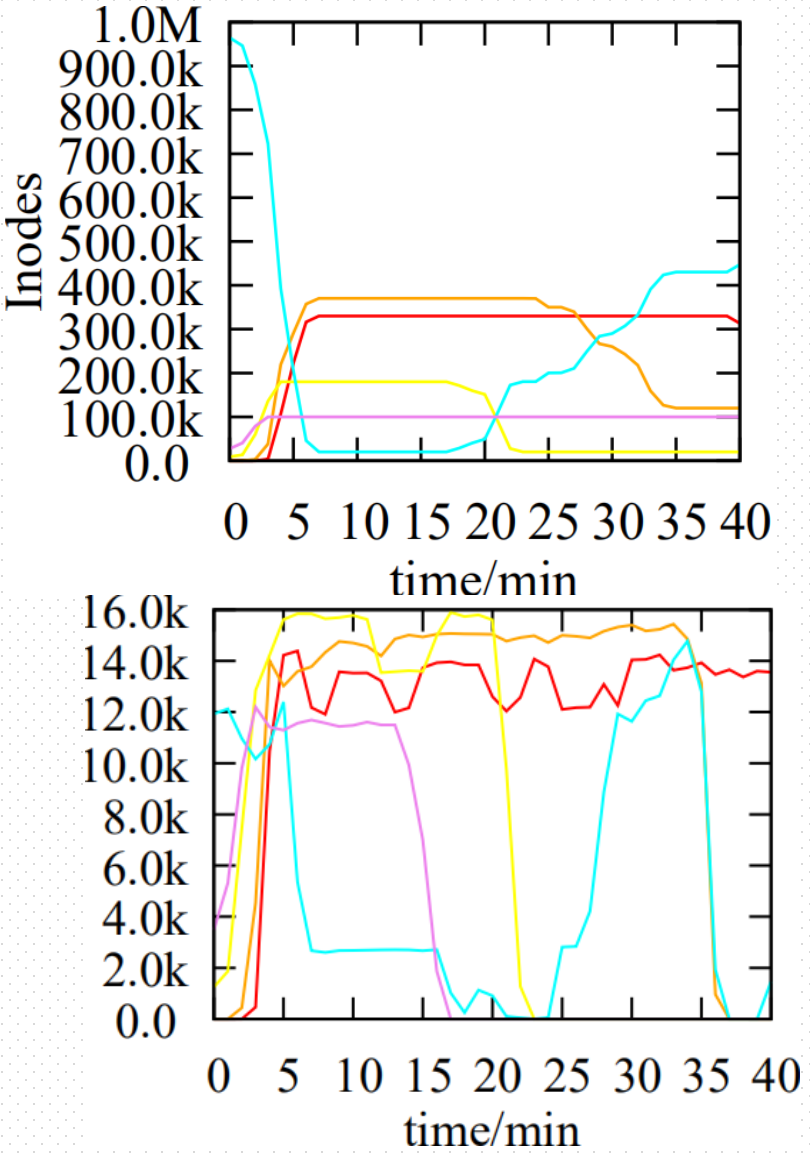
- Problem:
  - Always choosing hot directories, which are not visited later.
- Lessons:
  - Popularity is not everything.
  - Hot directories do not mean the future.





Case study: Zipfian

- Problem:
  - Migrated hot directories are found hot again in the importer's view and migrated again.
- Lesson:
  - Popularity-based policy should be global.



Optimization Goals

- Imbalance Factor Minimization
- Negative Impact Minimization
- Aggregated Throughput Maximization

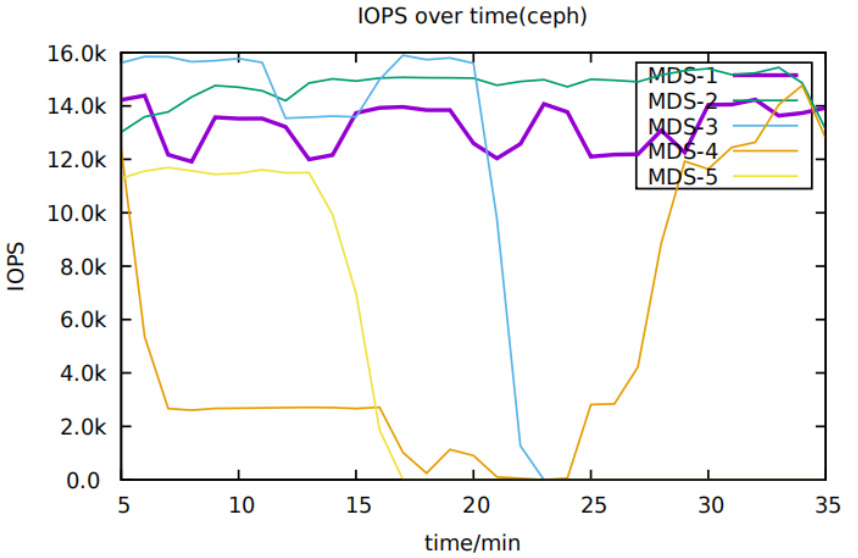
Preliminary Solutions

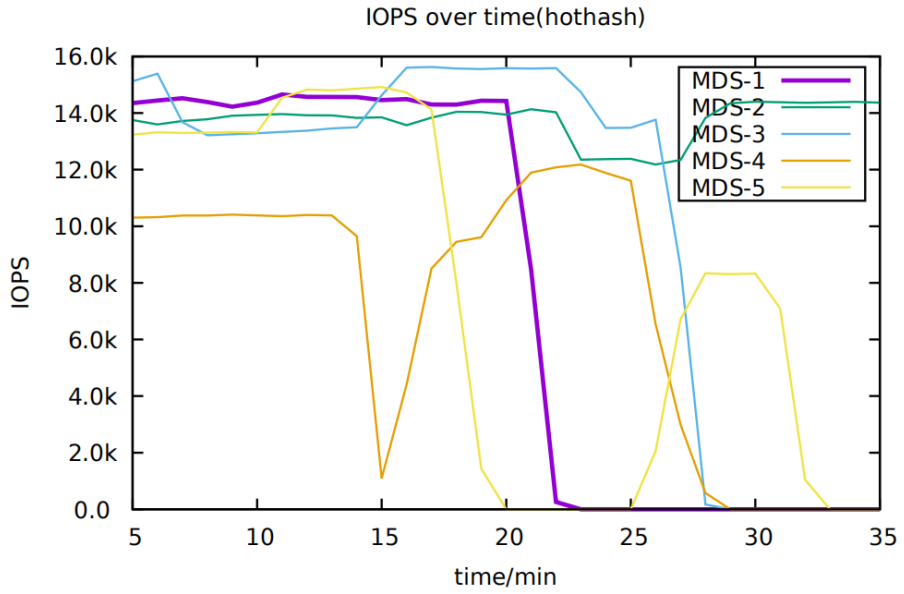
We use several preliminary solutions to solve this problem.

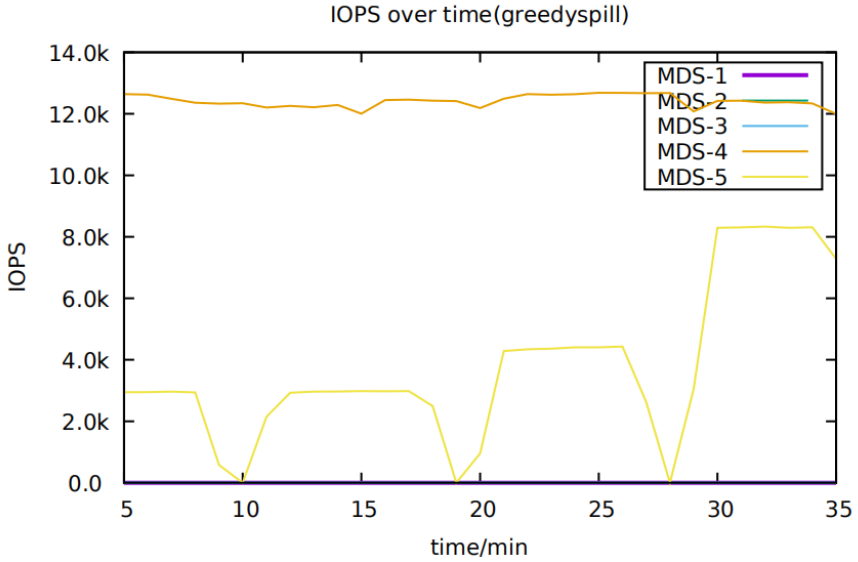
1. Dynamic Hashing
2. Pinning directories according to client IDs
3. Foreseeing dividing the directory tree according to its trace

Dynamic Hashing

- Policy:
  - Based on popularity
  - Select role based on hashing
  - Select only hot directories
  - No changes to migration
  - Evaluated on Zipfian workload

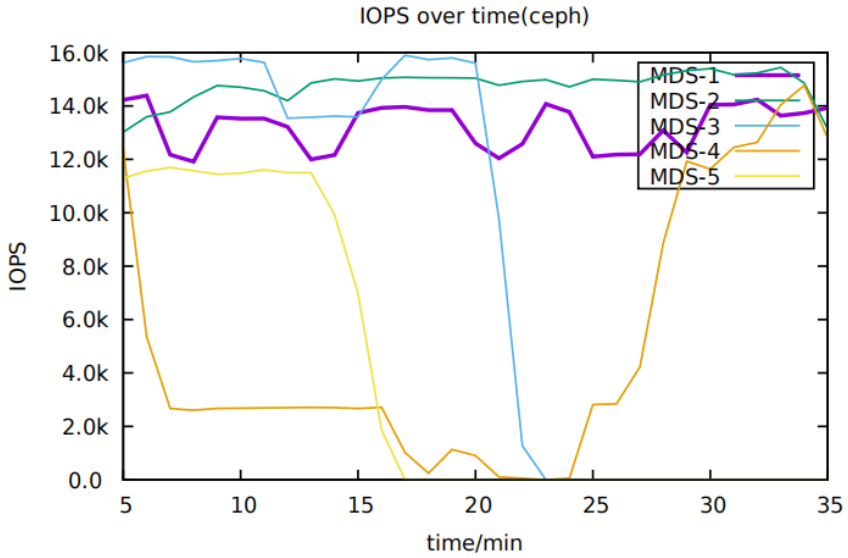


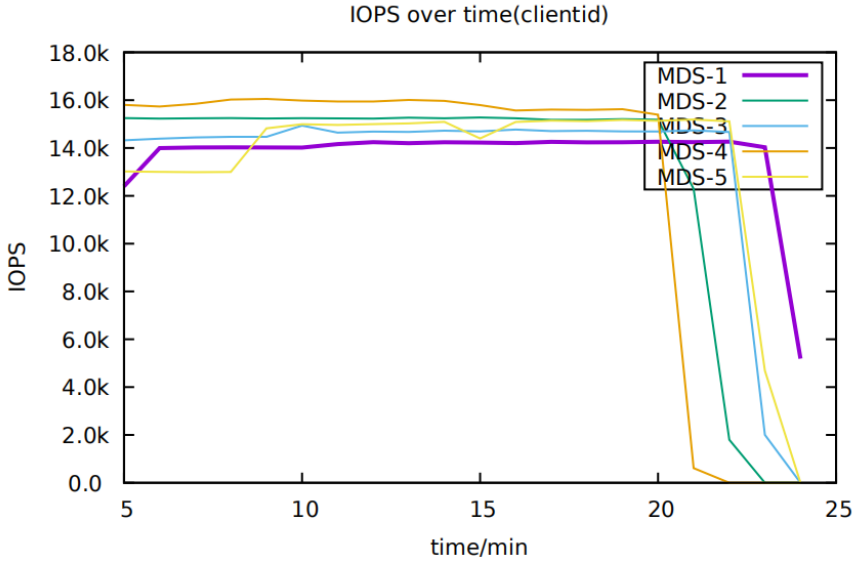


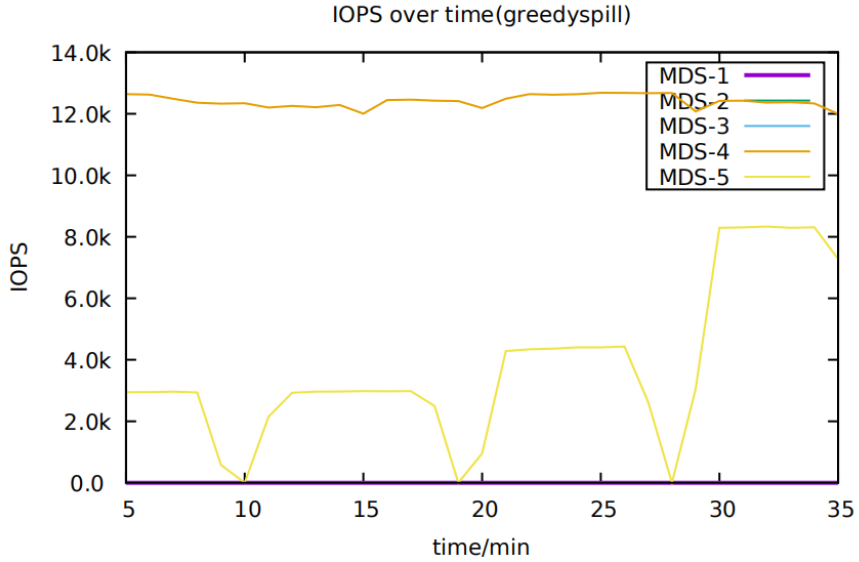


Pinning due to client IDs

- Policy:
  - Care nothing about popularity and MDS load
  - Select roles by first level directory name
  - Whole subtrees are migrated
  - No changes to migration
  - Evaluated on Zipfian workload

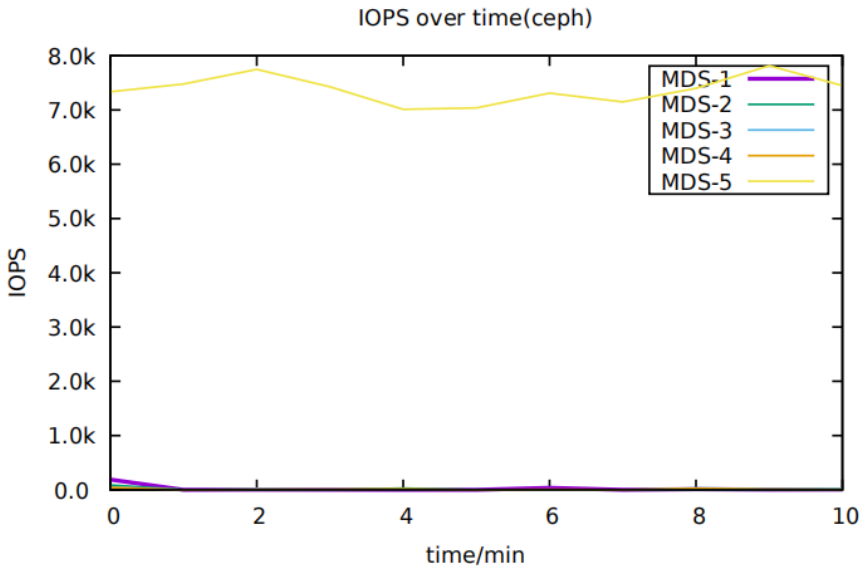


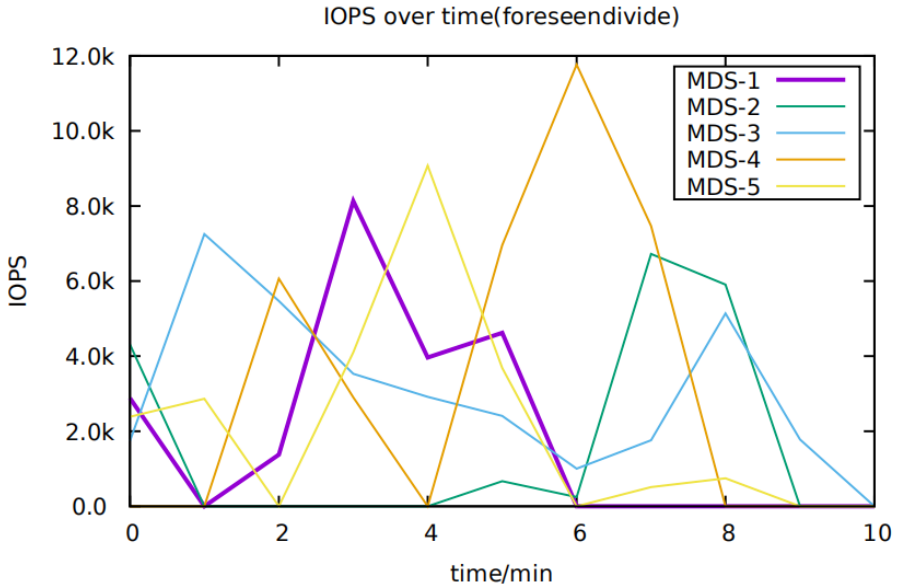


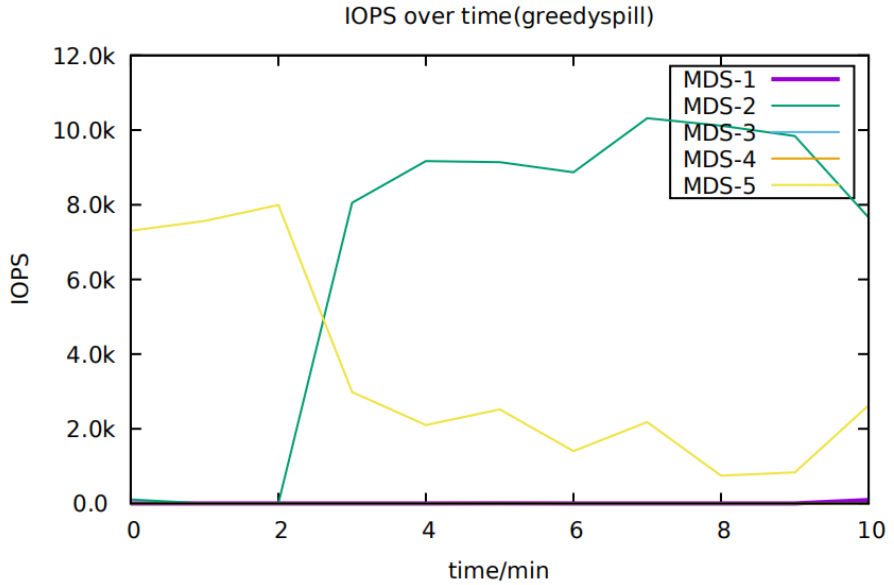


Foreseeing Dividing

- Policy:
  - Care nothing about popularity & MDS load
  - Cut the directory tree into several fragments and assign them to MDSs based on popularity in the trace
  - No changes to migration
  - Evaluated on Zipfian workload







Summaries & Plans

- Summaries:
  - Factors we need to consider besides popularity:
  - There’s no policy for all workloads.
    - Specially optimized policy for all workloads
    - Adaptive policy
- Plans:
  - Formalize the multi-objective optimization problem
  - Workload-aware policy decision