File Systems Unfit as Distributed Storage Backends: Lessons from 10 Years of Ceph Evolution

Speaker: 李嘉豪

Authors: Abutalib Aghayev, Sage Weil (Red Hat),
Michael Kuchnik, Mark Nelson (Red Hat),
Greg Ganger, George Amvrosiadis

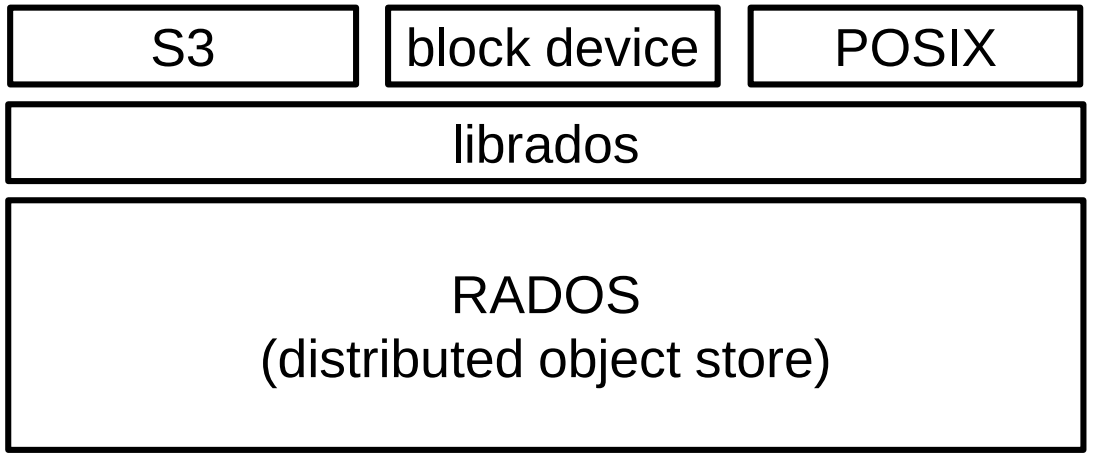
Outline

USTC, CHINA ADSLAB

1. What is Ceph?
2. Challenges
3. Design of Bluestore
4. Evaluation and results
5. Conclusion and Future Work

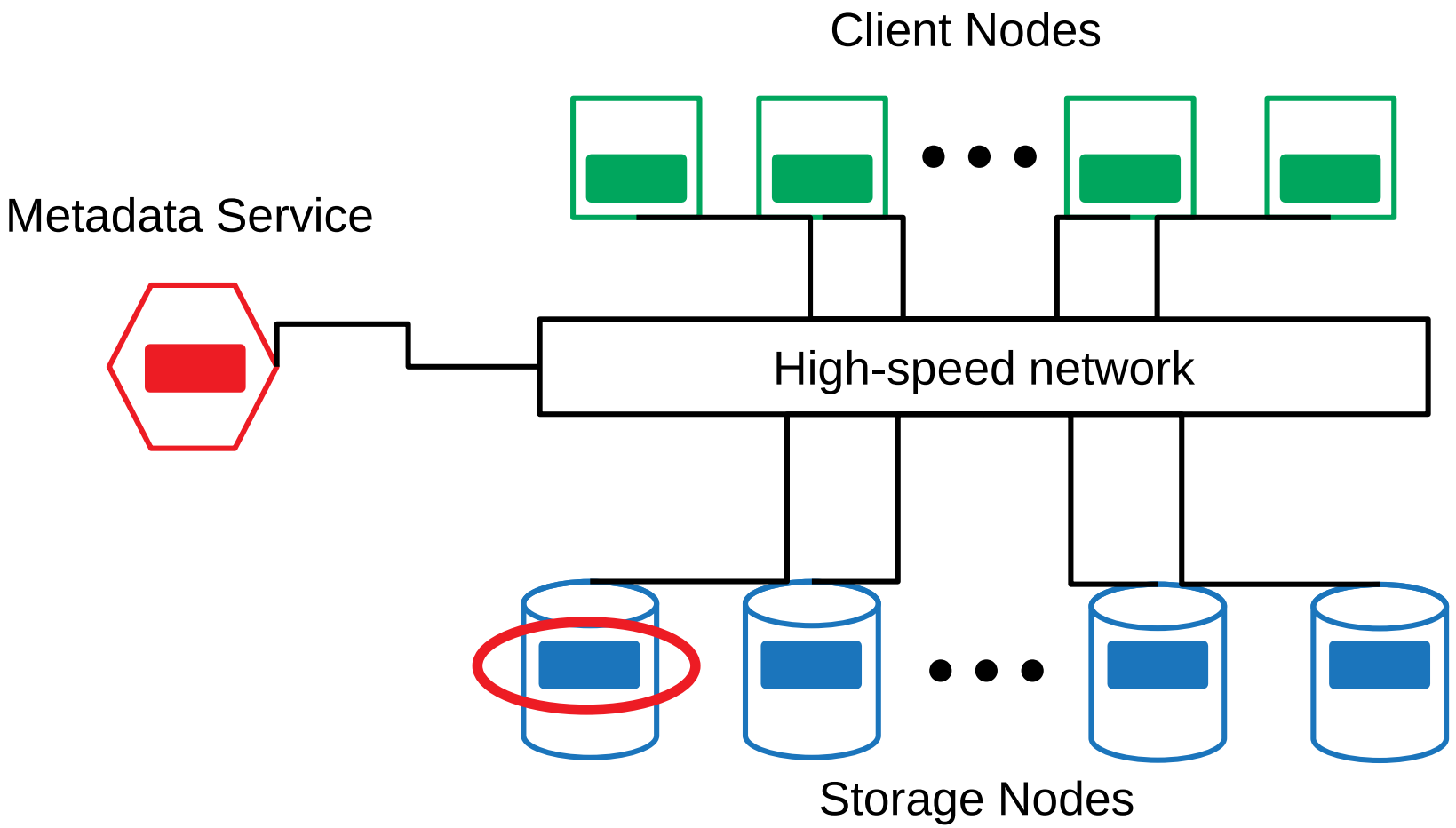
What is Ceph?

- A distributed storage system started in 2004 at UCSC
- Today, a widely used object, block, and file system

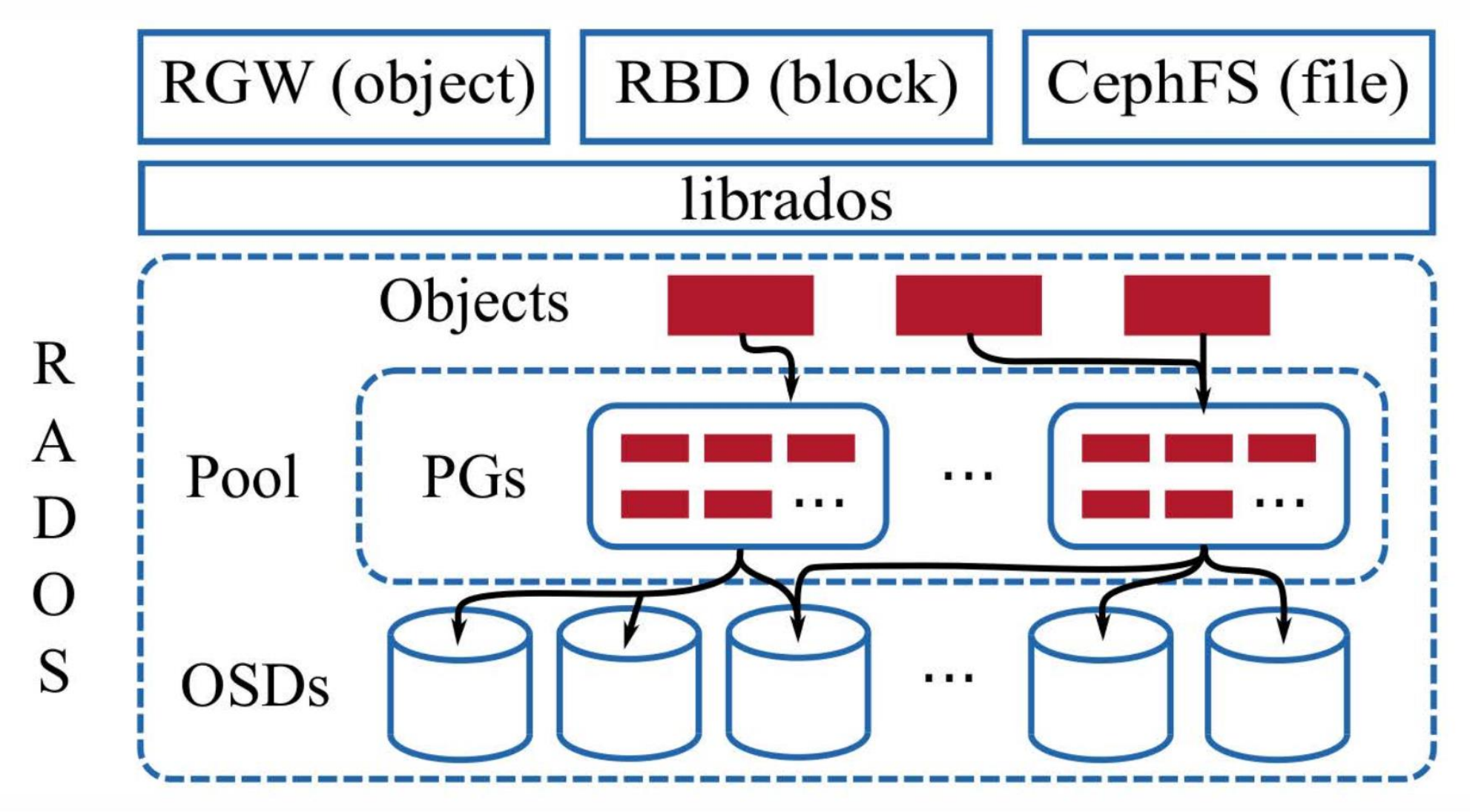


ceph

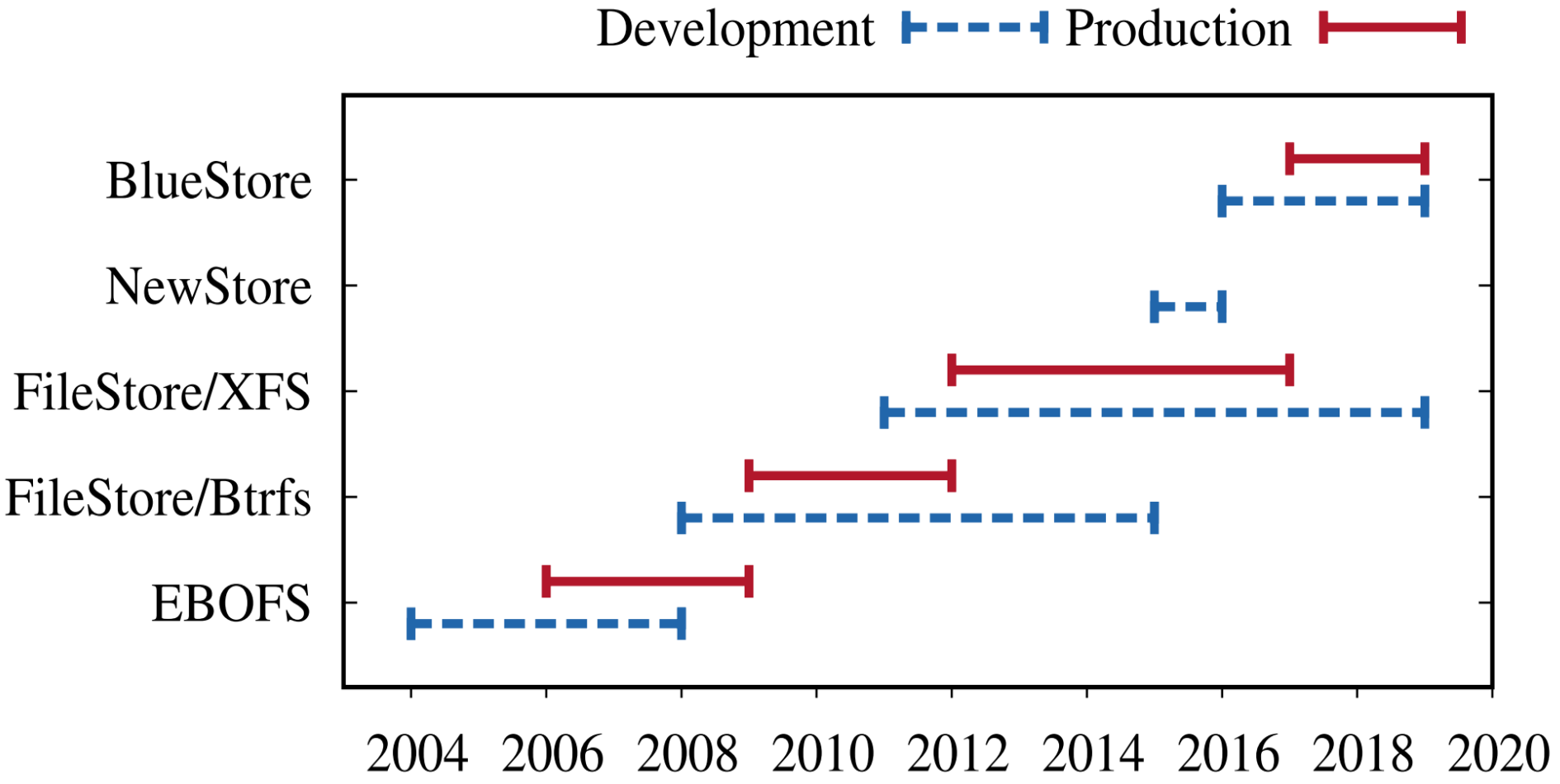
Distributed Storage Overview



Ceph Overview



Ceph - Storage Engine Evolution



Outline

USTC, CHINA ADSLAB

1. What is Ceph
2. Challenges
3. Design of Bluestore
4. Evaluation and results
5. Conclusion and Future Work

Challenge I: Efficient Transactions

Transactions simplify application development by encapsulating a sequence of operation into a single atomic unit of work.

e.g.: Modification to an object:

1. Update metadata
2. Write contents

Leveraging FS Internal Transactions

Drawbacks:

1. Most FS's transaction functionality is unavailable to users (mainly to keep internally consistent)

2. Btrfs expose internal transaction mechanisms to user, but it lacks rollback functionality, which can cause inconsistency.

Implementing the WAL in User Space

WAL:

1. Record modification in sequential logs

2. Call fsync

3. Apply modification to the file (successive transactions must wait until step 3 is done to witness the effect of this transaction)

Implementing the WAL in User Space

WAL:

1. Record modification in sequential logs

   Extra writes

2. Call fsync

   Can respond after this step is complete

3. Apply modification to the file (successive transactions must wait until step 3 is done to witness the effect of this transaction)

   inefficient for read-modify-write operations

Non-Idempotent Operations

Idempotent:$A * A = A$

Operations need to be idempotent to use WAL

BUT:

1. clone $a \rightarrow b$
2. update a
3. update c

Is non-idempotent on Btrfs or XFS (think about if error occurred between step 2 and 3)

Double writes

Latency can be hidden, but bandwidth cannot.

New objects may avoid double writes by only logging metadata changes.

However, Filestore's usage of the filesystem make this method hard to use.

Using a KV-Store as the WAL

NewStore Features:

- Store metadata in RocksDB
- Data overwrites are logged into RocksDB
- Namespace is decoupled from FS hierarchy

Using a KV-Store as the WAL

Newstore Advantages:

1. KV interface allows reading the new state of an object without waiting for transaction to commit

2. Operations can be replayed by Copy-on-Write.

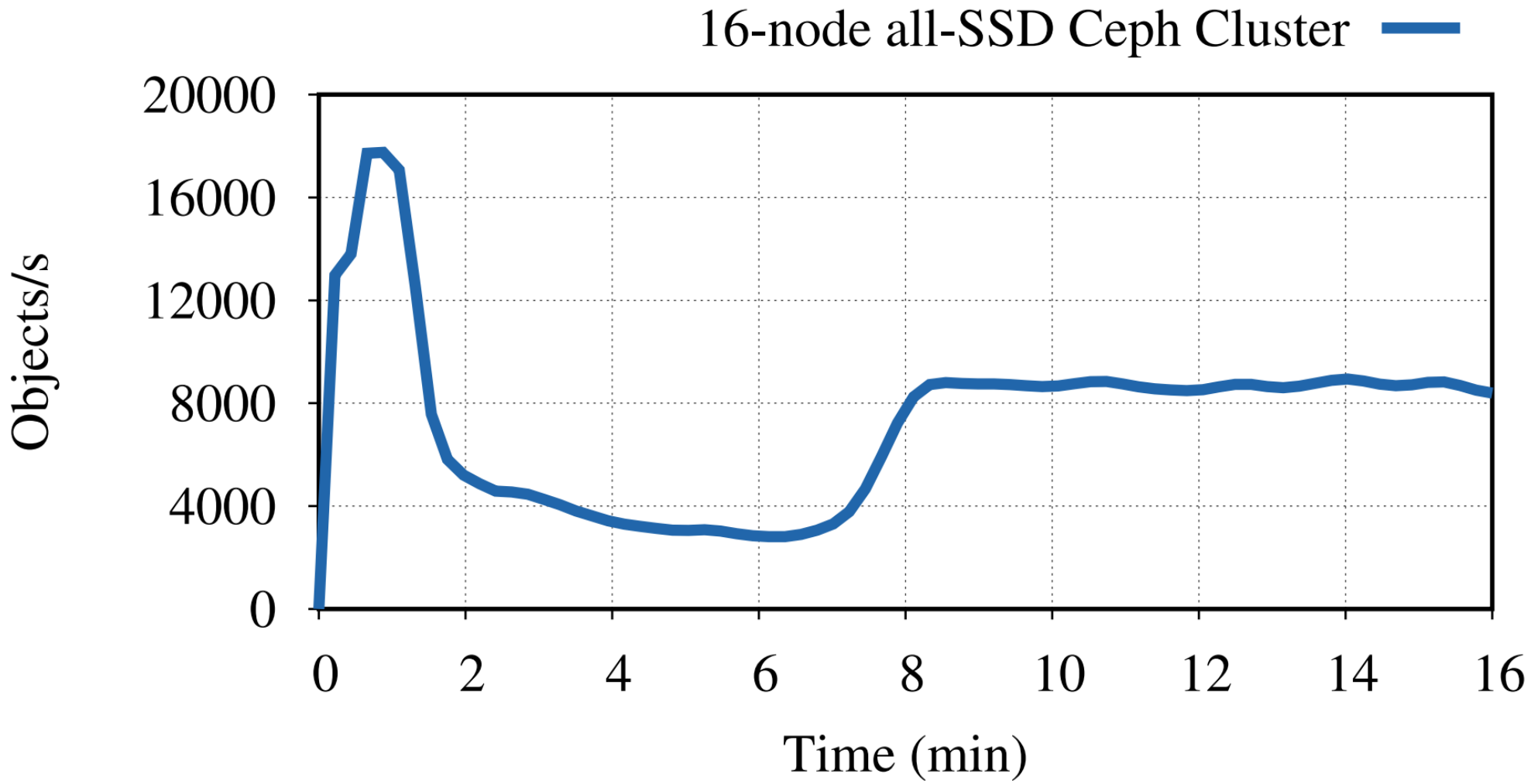
3. Double writes can be avoided for new objects

Challenge II: Fast Metadata Operations

Enumeration is necessary for operations like scrubbing, recovery, or for serving list calls.

- Processing millions of inodes at once reduces the effectiveness of dentry cache
- Directory contents spread out due to XFS features, leading to slow split operations.

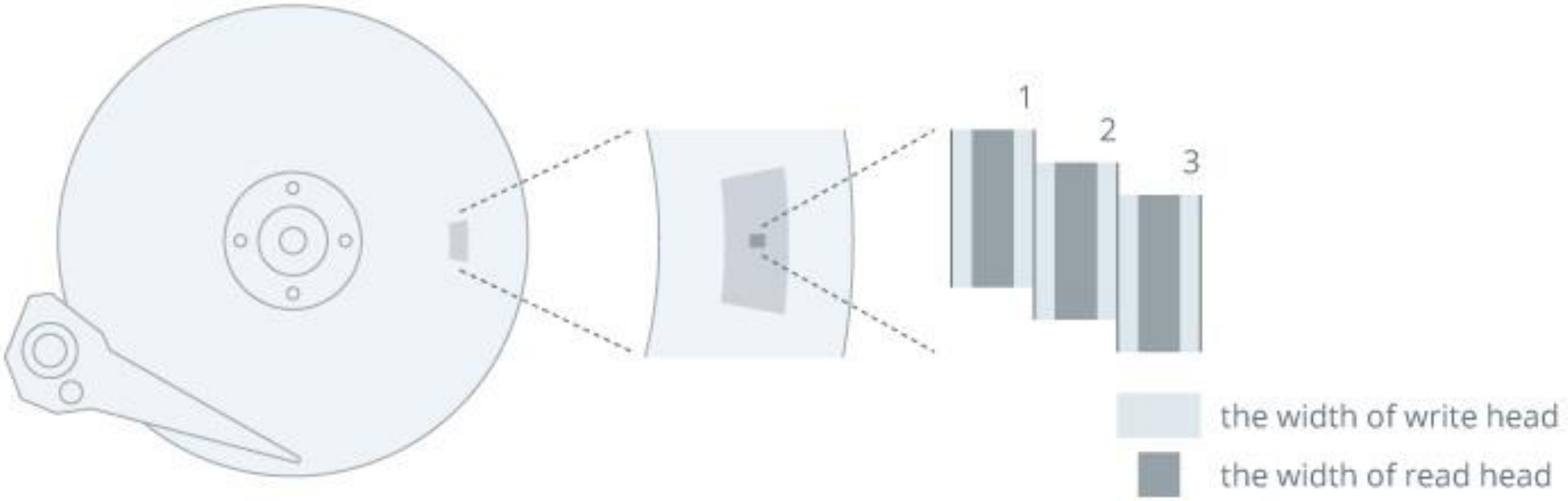
Challenge II: Fast Metadata Operations



Challenge III: New Storage Hardware

USTC, CHINA ADSLAB

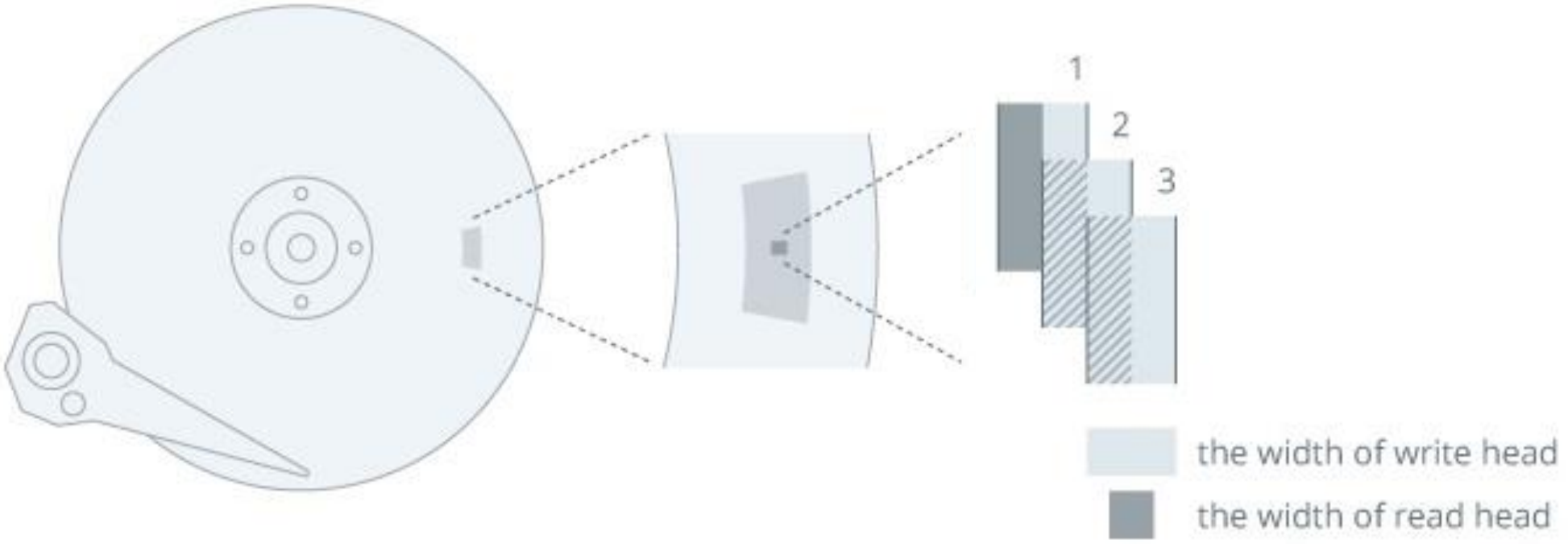
PMR HDDs



Challenge III: New Storage Hardware

USTC, CHINA ADSLAB

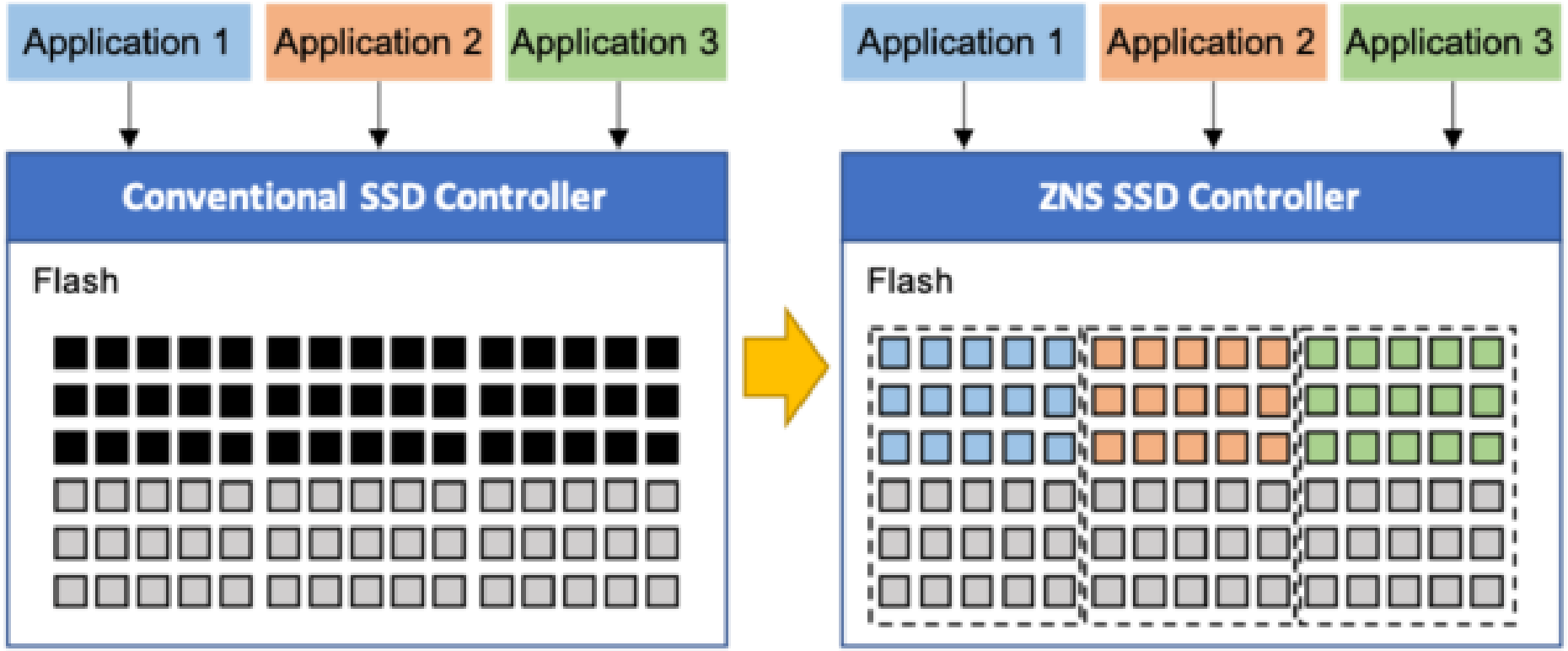
SMR HDDs



Challenge III: New Storage Hardware

USTC, CHINA ADSLAB

ZNS SSDs



Regular SSD: Device controls data placement

ZNS SSD: Applications control data placement in zones

Challenge III: New Storage Hardware

Attempts to modify production filesystems (XFS,ext4,etc.) to work with the zone interface have so far been unsuccessful.

Production filesystems are overwrite filesystems, whereas the zone interface requires copy-on-write approach.

Other Challenges

- Without the complete control of the IO stack, it is hard for distributed filesystems to enforce storage latency SLO

- Not good enough copy-on-write support

Outline

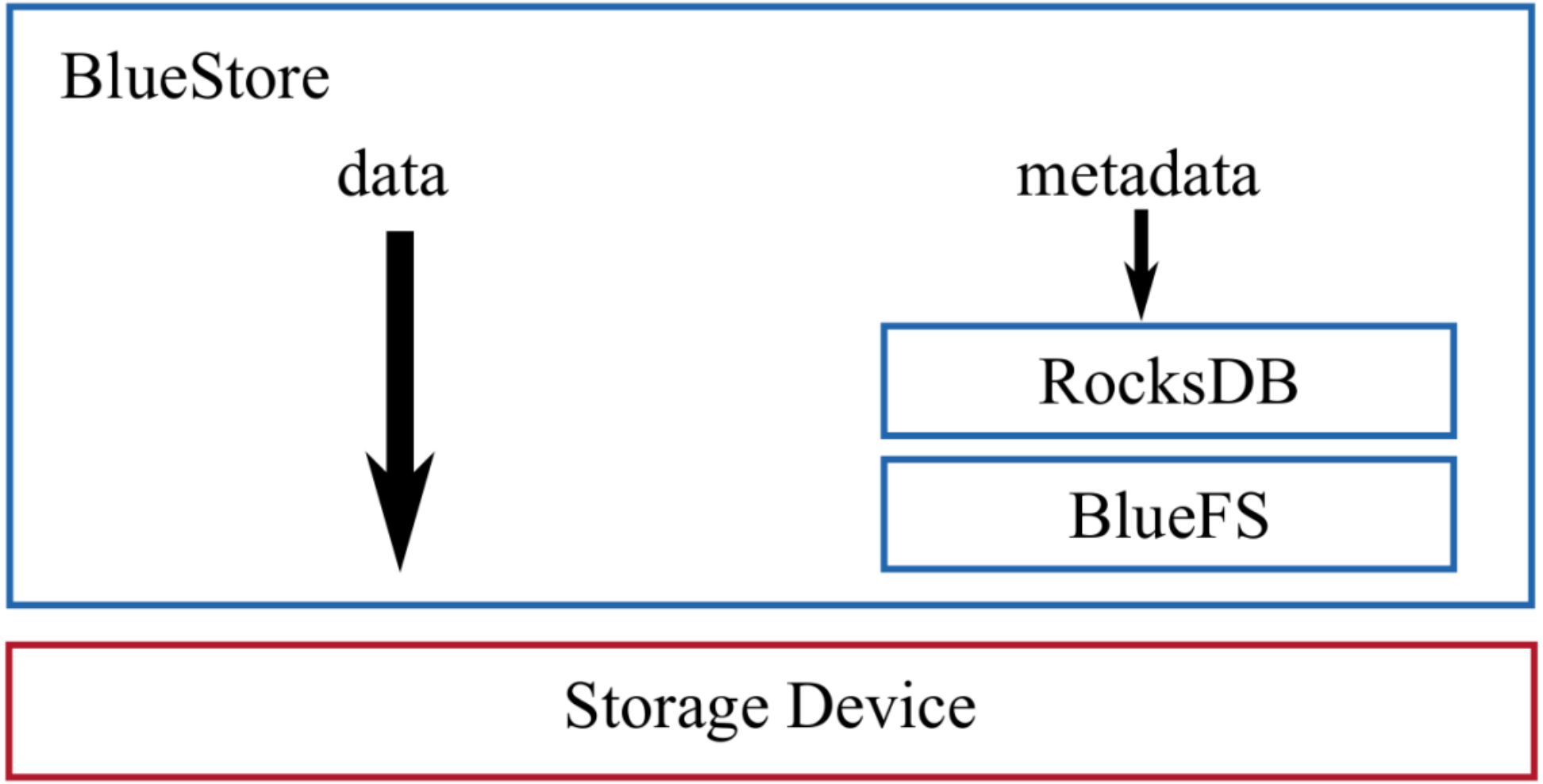
USTC, CHINA ADSLAB

1. What is Ceph
2. Challenges
3. Design of Bluestore
4. Evaluation and results
5. Conclusion and Future Work

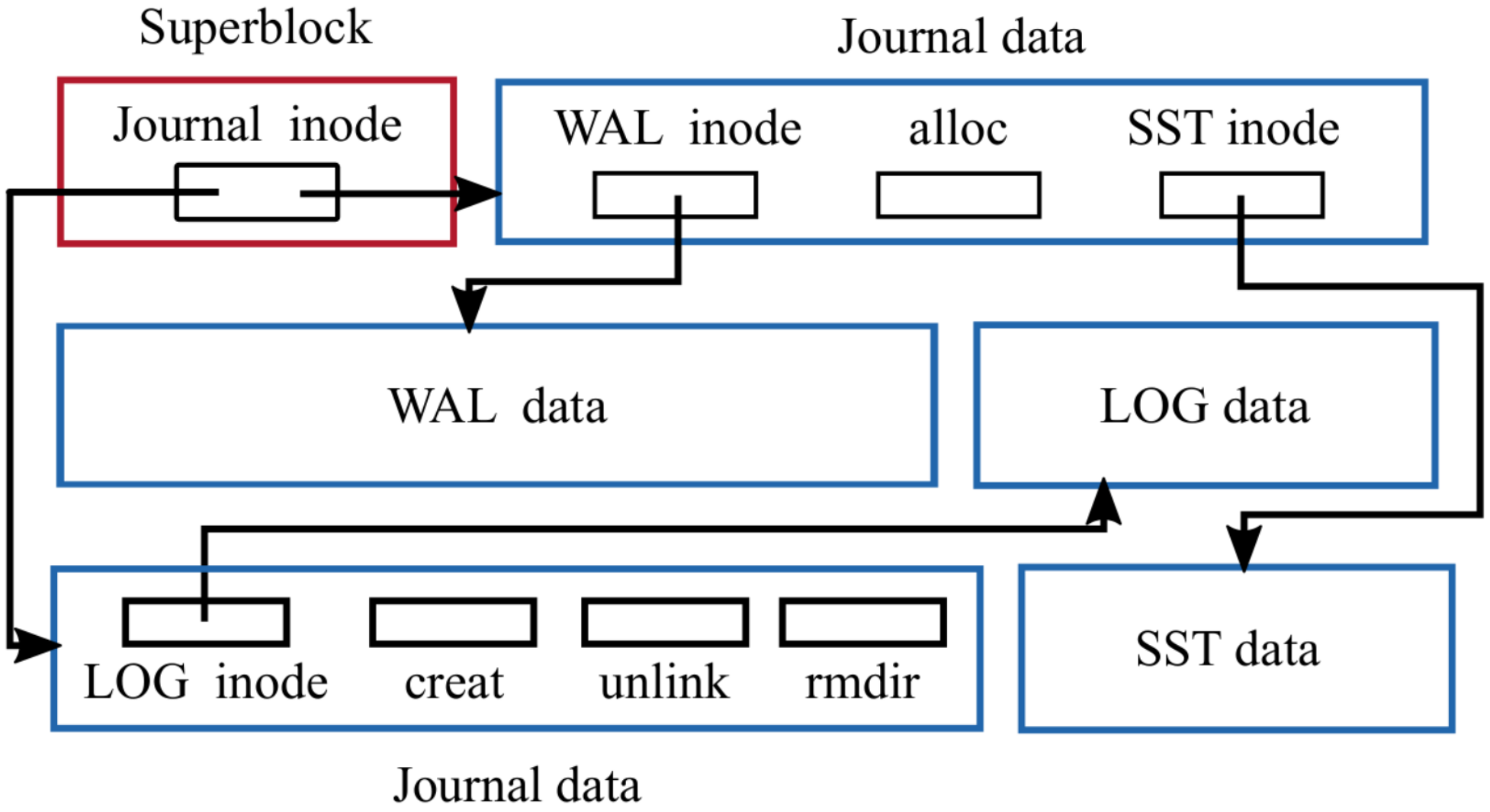
Goals of BlueStore

1. Fast metadata operations
2. No consistency overhead for object writes
3. Copy-on-write clone operation
4. No journaling double-writes
5. Optimized IO pattern for HDD and SSD

Design of BlueStore



Layout of BlueFS



Design of BlueStore

1. Fast metadata operations

   Store metadata in RocksDB

2. No consistency overhead for object writes

   One cache flush for data write and metadata write

Design of BlueStore

3. Copy-on-write clone operation
4. No journaling double-writes
5. Optimized IO pattern for HDD and SSD

BlueStore is a copy-on-write storage engine

Small writes (16KiB for SSD, 64KiB for HDD) first

Features Enable by BlueStore

1. Choose the checksum block size based on the IO hints
2. Overwrite of EC data
3. Transparent compression
4. Exploring new interfaces

Outline

USTC, CHINA ADSLAB

1 What is Ceph

2 Challenges

3 Design of Bluestore

4 Evaluation and results

5 Conclusion and Future Work

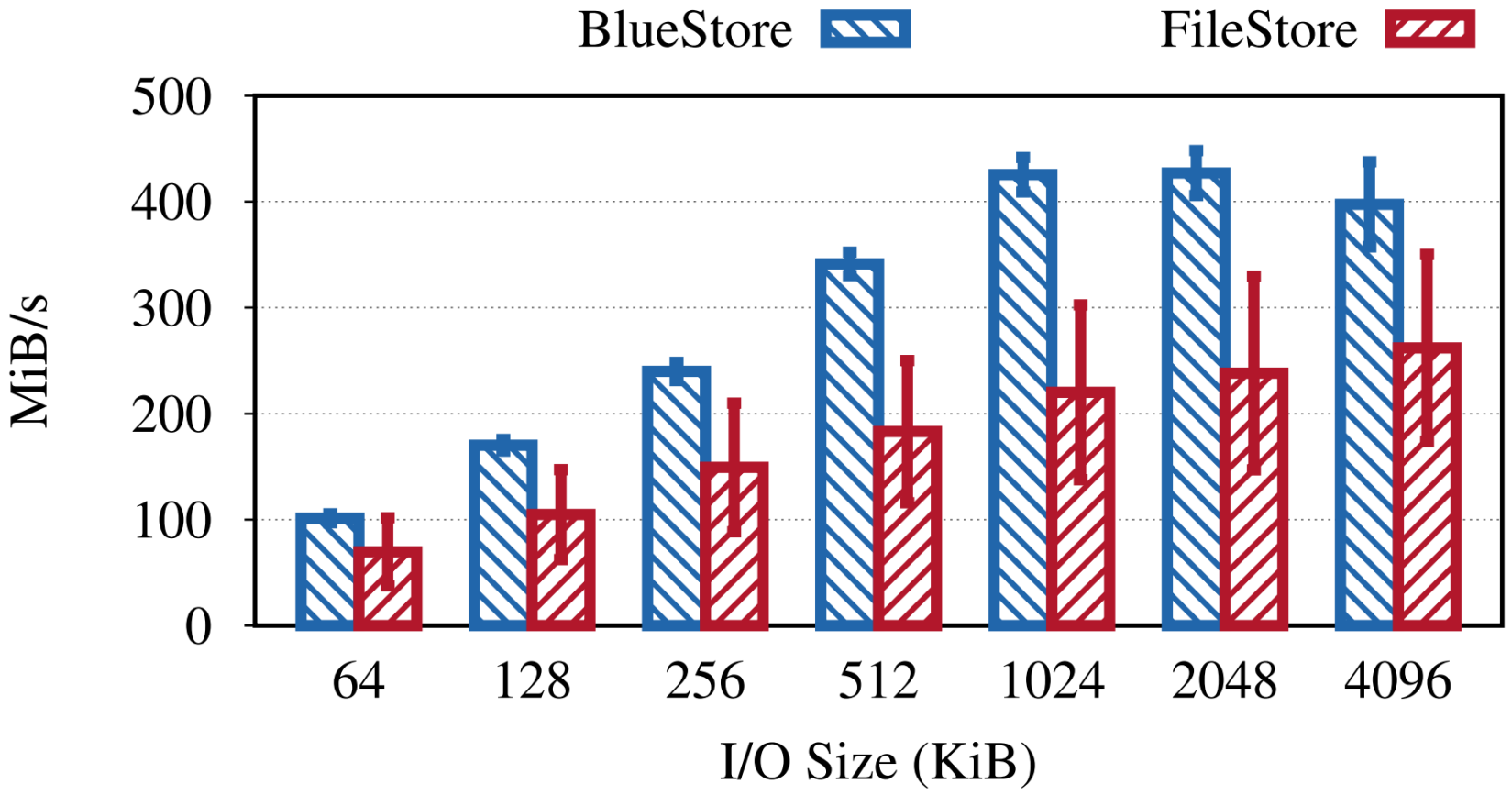
Evaluation Setup

16 nodes Ceph cluster

1. Switch: Cisco Nexus 3264-Q 40GbE
2. CPU: 16-core Intel Xeon E5-2698Bv3 2GHz
3. RAM: 64GiB
4. SSD: Intel P3600 NVMe SSD
5. HDD: Seagate 4TB 7200RPM
6. NIC: Mellanox MCX314A-BCCT 40GbE

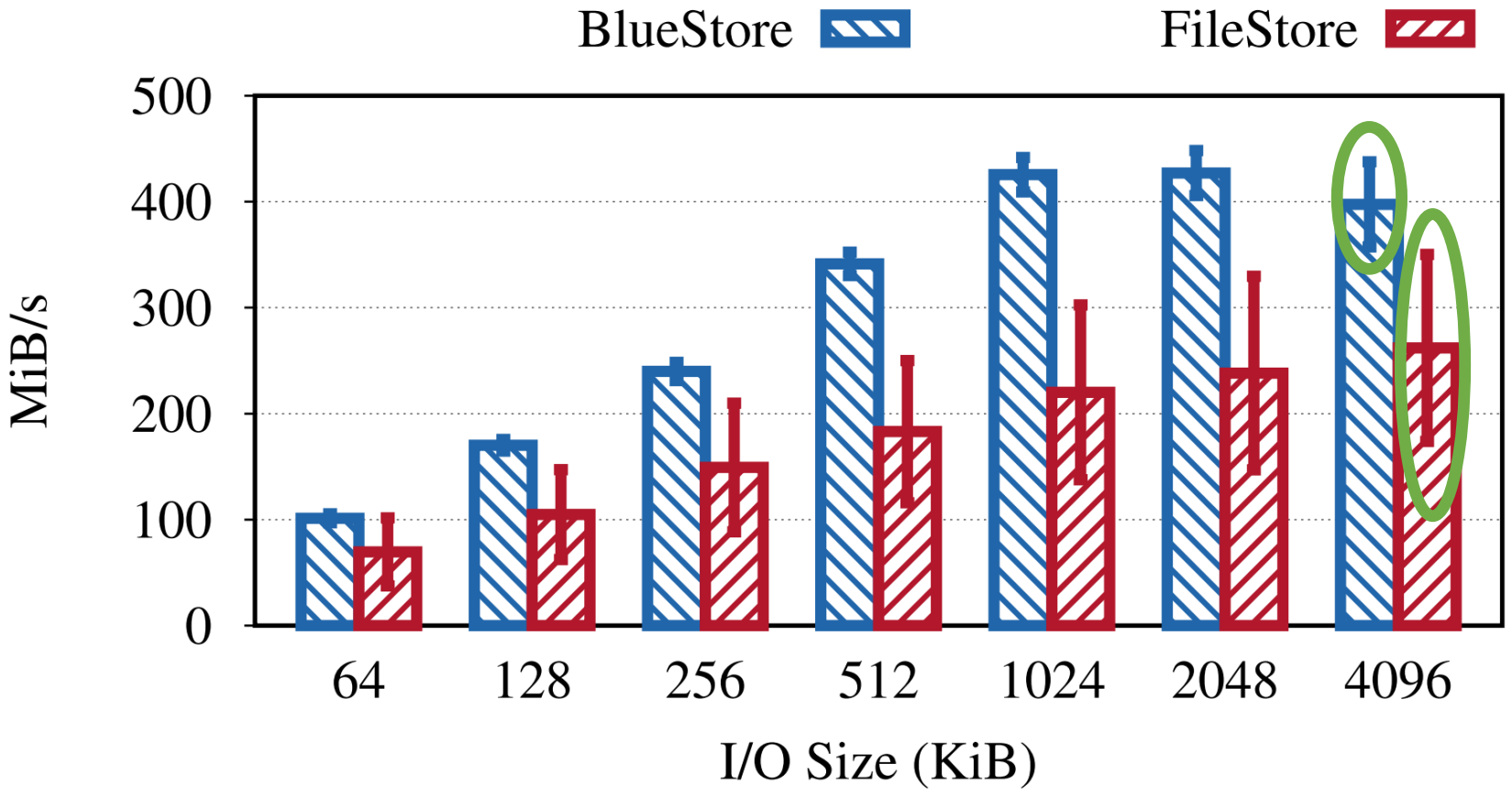
Bare RADOS Benchmarks

Different object sizes written with a queue depth of 128 (HDD)



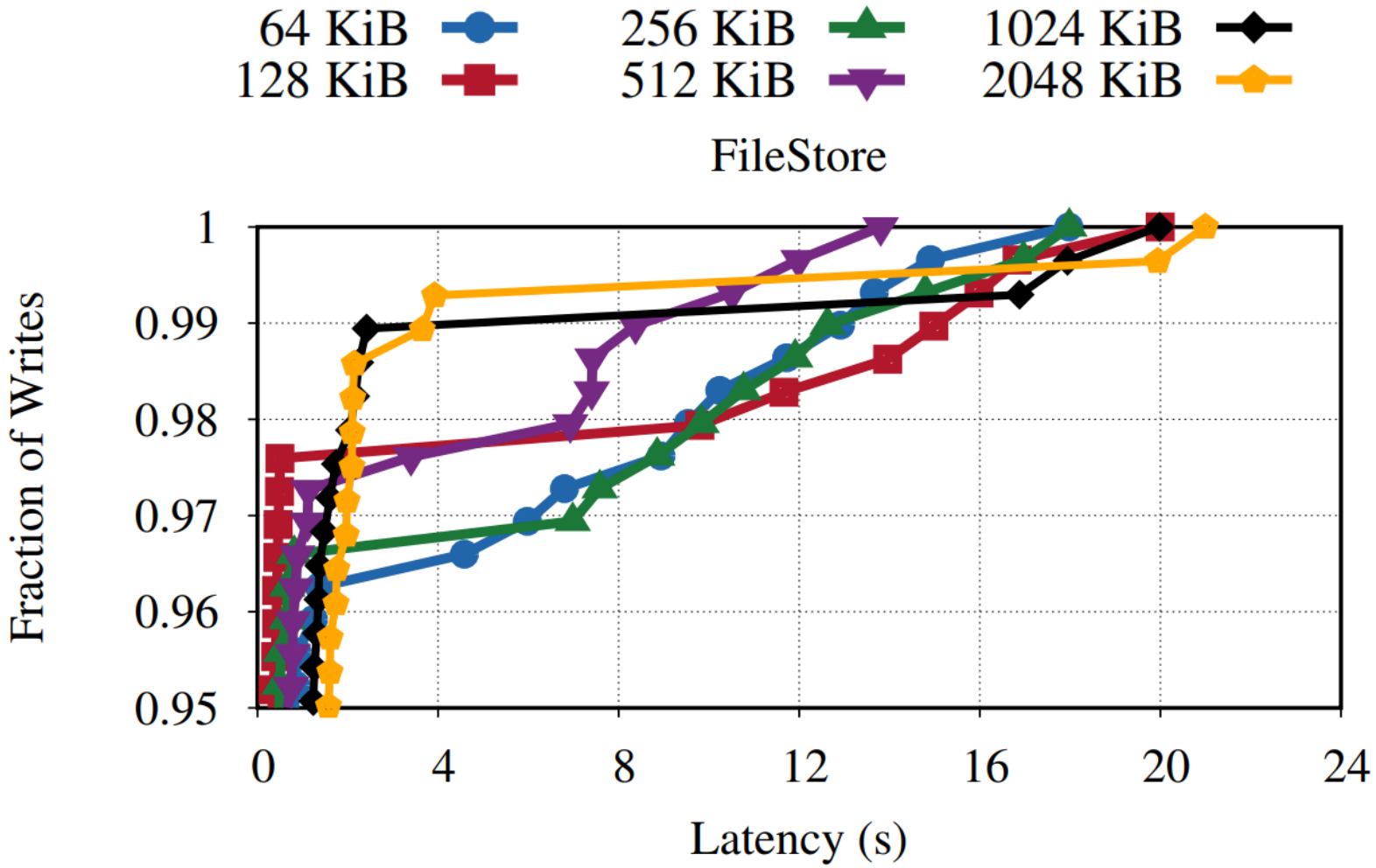
Bare RADOS Benchmarks

Different object sizes written with a queue depth of 128 (HDD)



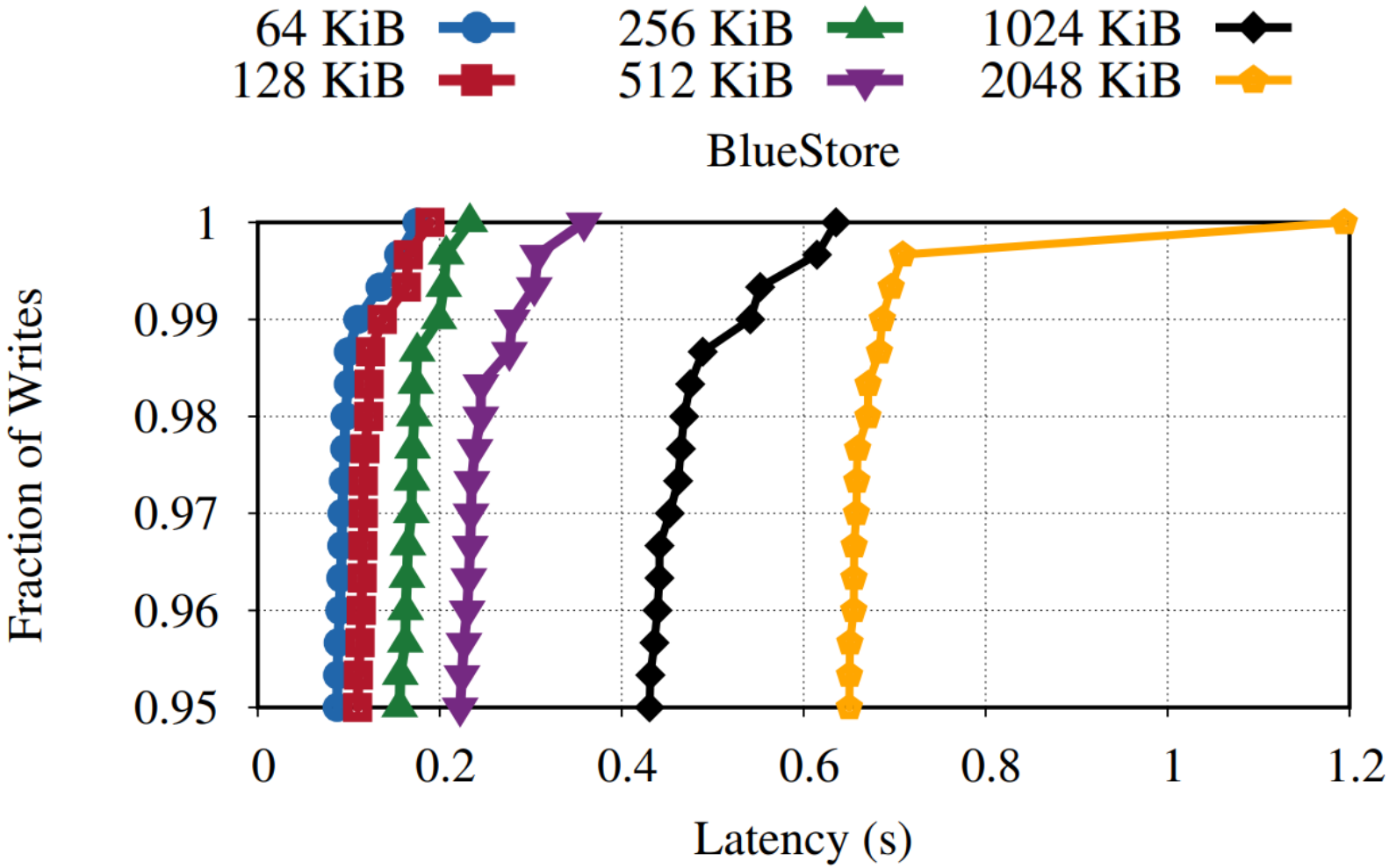
Bare RADOS Benchmarks

Different object sizes written with a queue depth of 128 (HDD)



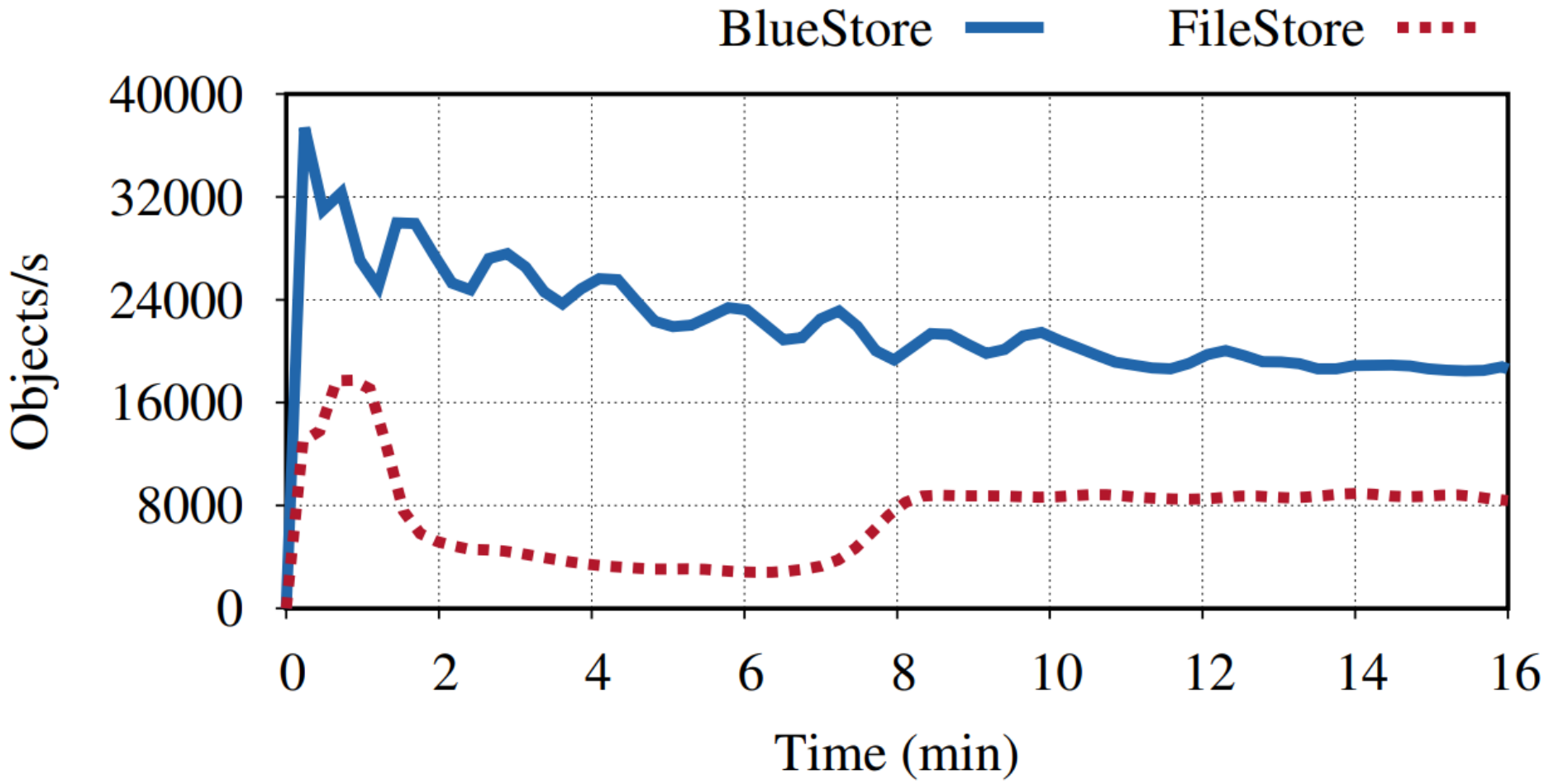
Bare RADOS Benchmarks

Different object sizes written with a queue depth of 128 (HDD)



Bare RADOS Benchmarks

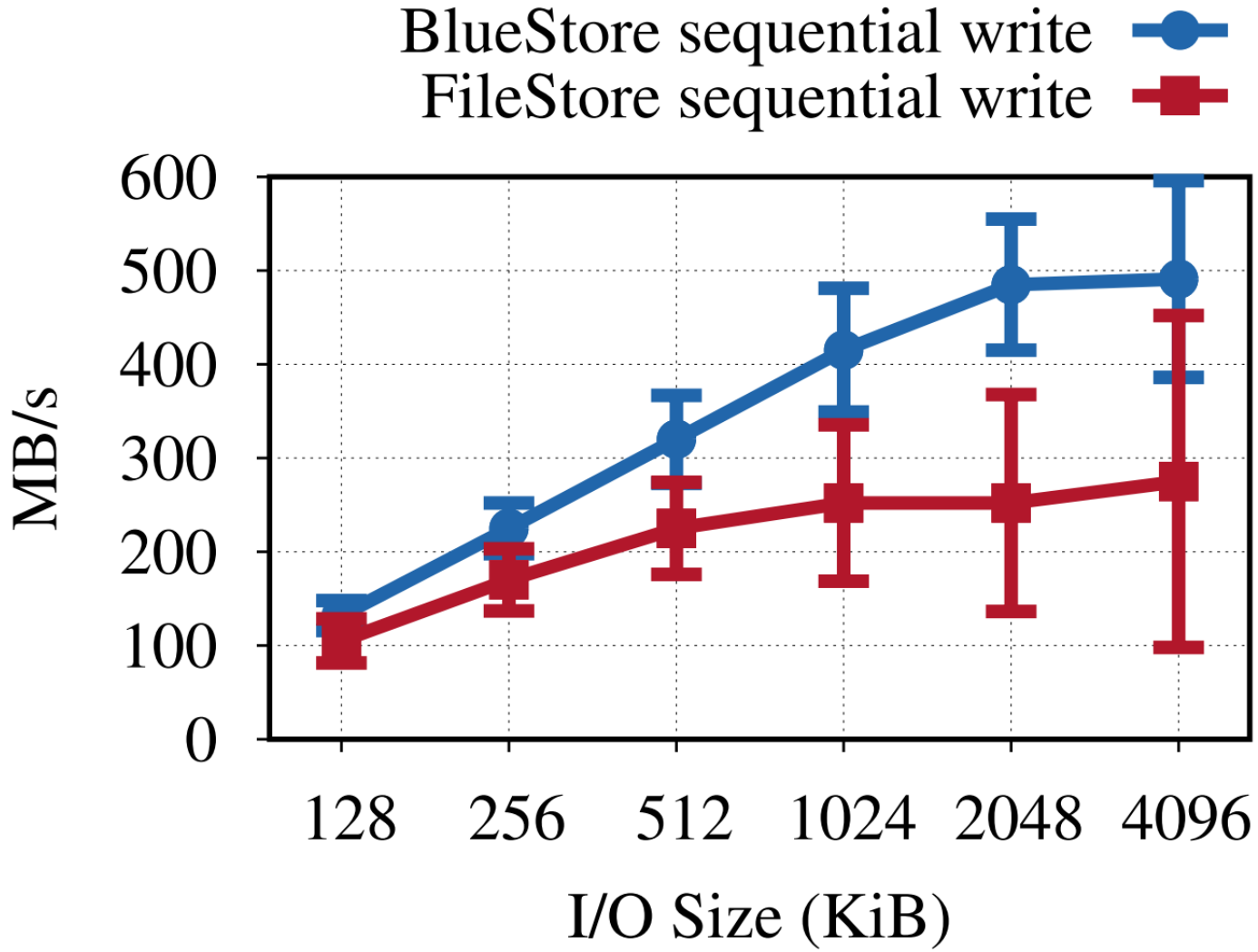
Different object sizes written with a queue depth of 128 (SSD)



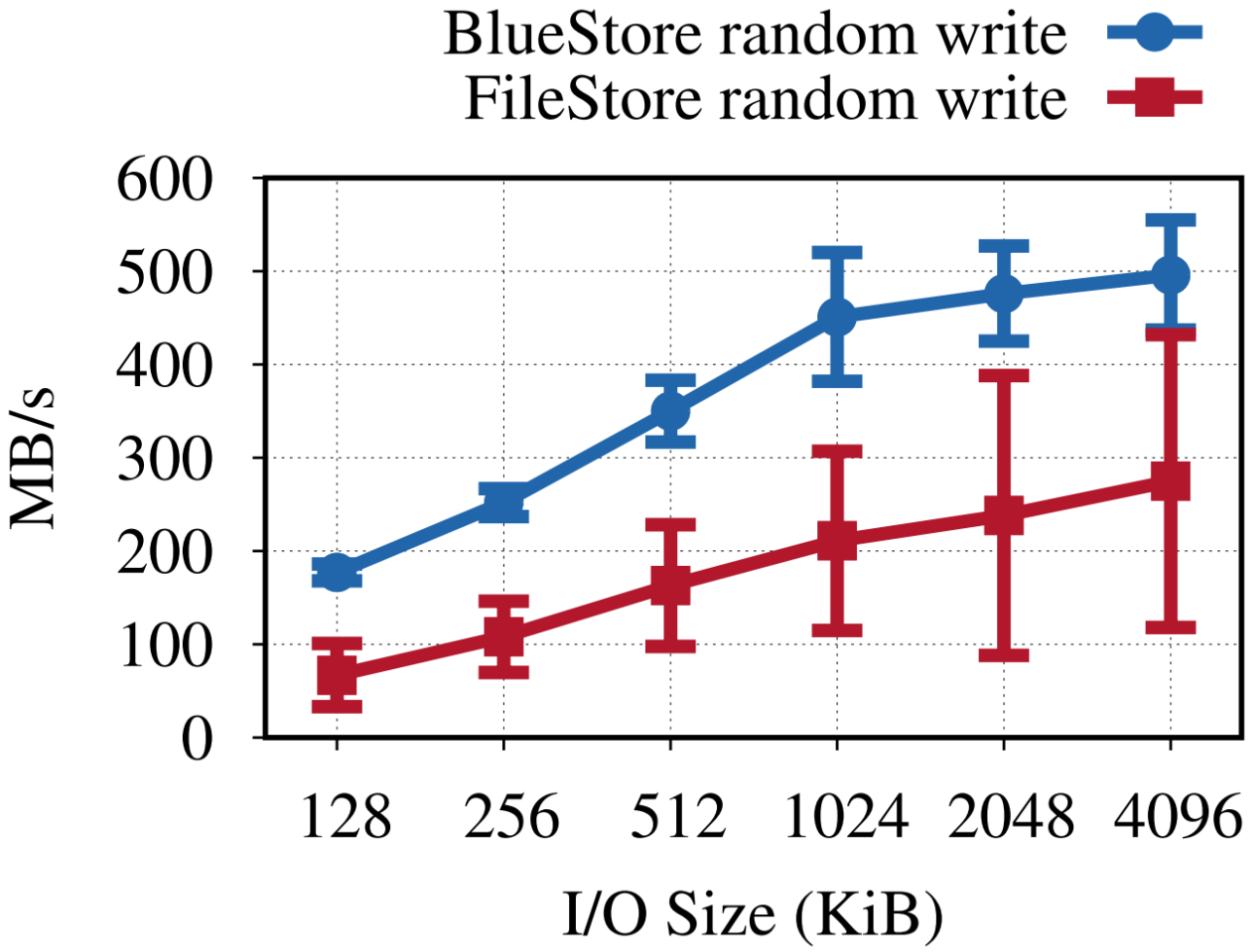
RBD Benchmarks

- Data written to the device is striped into 4MiB RADOS objects
- Create a 1TB virtual block device, and format it with XFS
- Use fio to run tests with queue depth of 256, IO size ranges from 4KiB to 4MiB
- Each test writes about 30GiB of data
- Cache is reset before every experiment

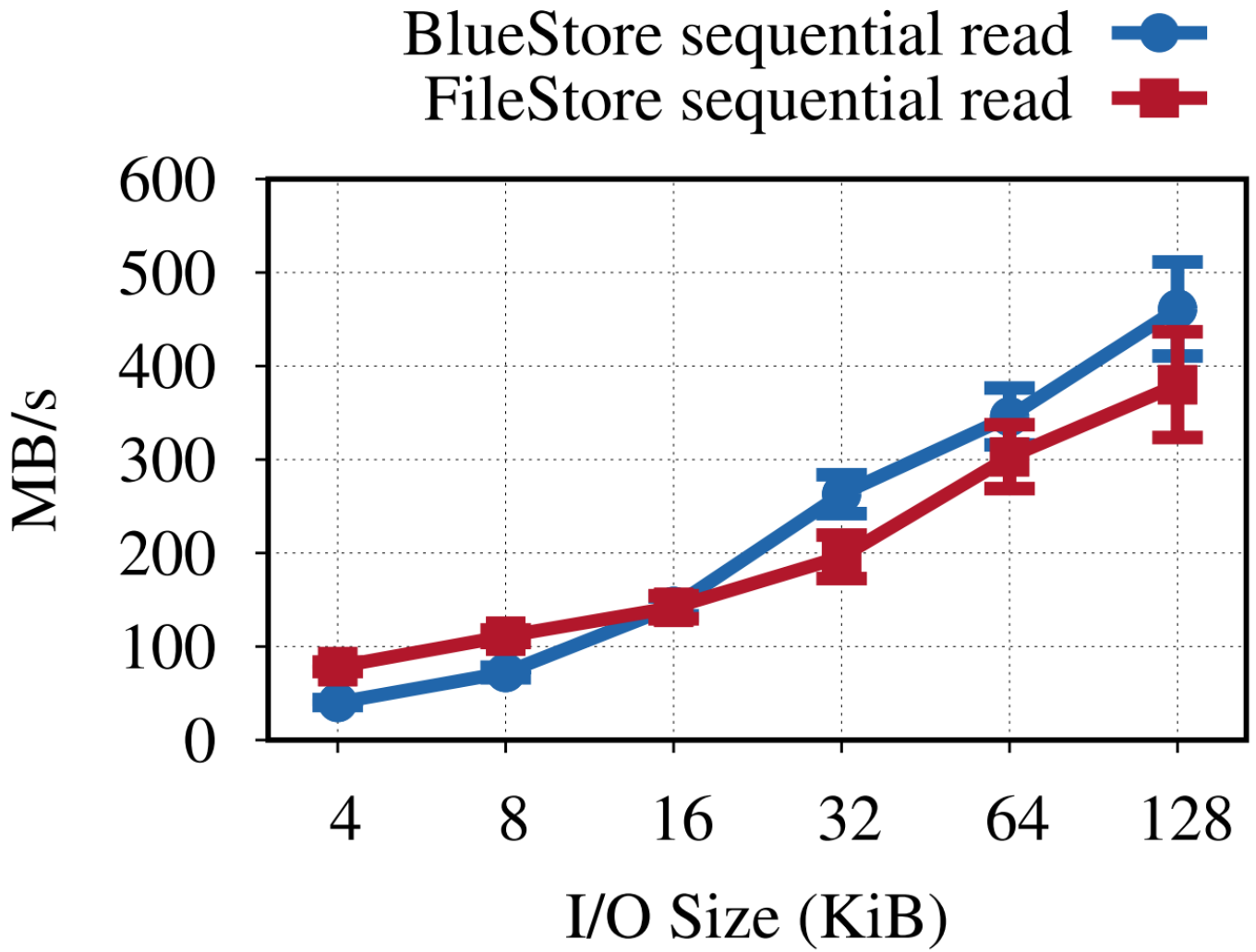
RBD Benchmarks (HDD)



RBD Benchmarks (HDD)

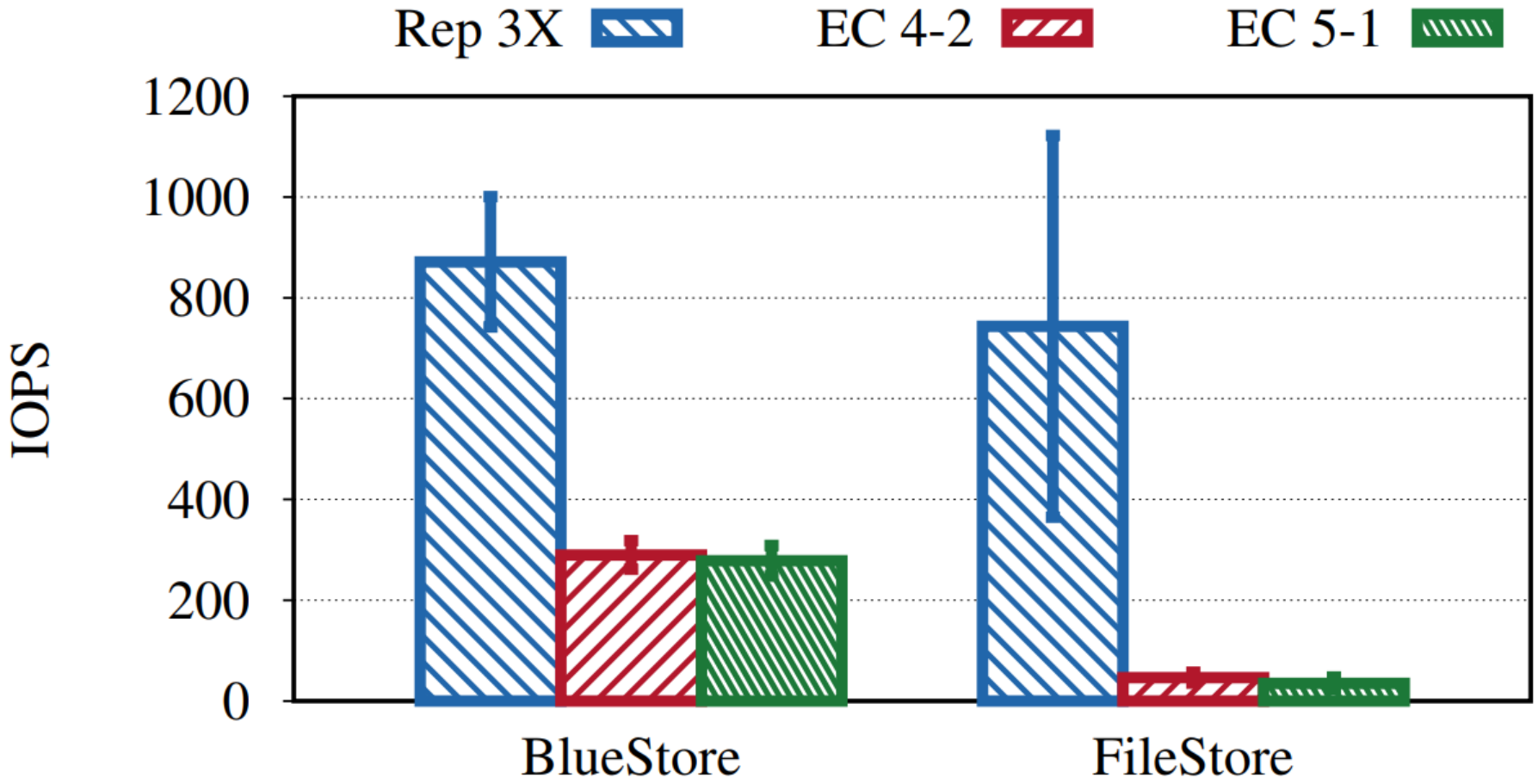


RBD Benchmarks (HDD)



Overwriting EC Data (HDD)

5GiB of random 4KiB writes with 256 queue depth



Outline

USTC, CHINA ADSLAB

1 What is Ceph

2 Challenges

3 Design of Bluestore

4 Evaluation and results

5 Conclusion and Future Work

Conclusion

- BlueStore achieves its design goals and outperformes FileStore which is established on FS

Conclusion

USTC, CHINA ADSLAB

- BlueStore achieves its design goals and outperformes FileStore which is established on FS

- BUT there are new challenges when building a storage backend on raw storage

New Challenges

- Cache sizing and writeback
  - How to dynamically resize page cache in user space?
- Key-value store efficiency
  - RocksDB’ s compaction and high write amplification becomes primary performance limit
  - Data serialization and copy consumes CPU time
  - RocksDB has its own threading model
- CPU and memory efficiency
  - How to reduce waste of memory due to data structure padding
  - On high-end NVMe SSDs, workloads become CPU-bound

Thanks for your attention!