IEEE MSST 2013

FSMAC: A File System Metadata Accelerator with Non-Volatile Memory

Jianxi Chen , Qingsong Wei, Cheng Chen , Lingkun Wu

Data Storage Institute, A*STAR, Singapore

May 10, 2013

DSI
A*STAR

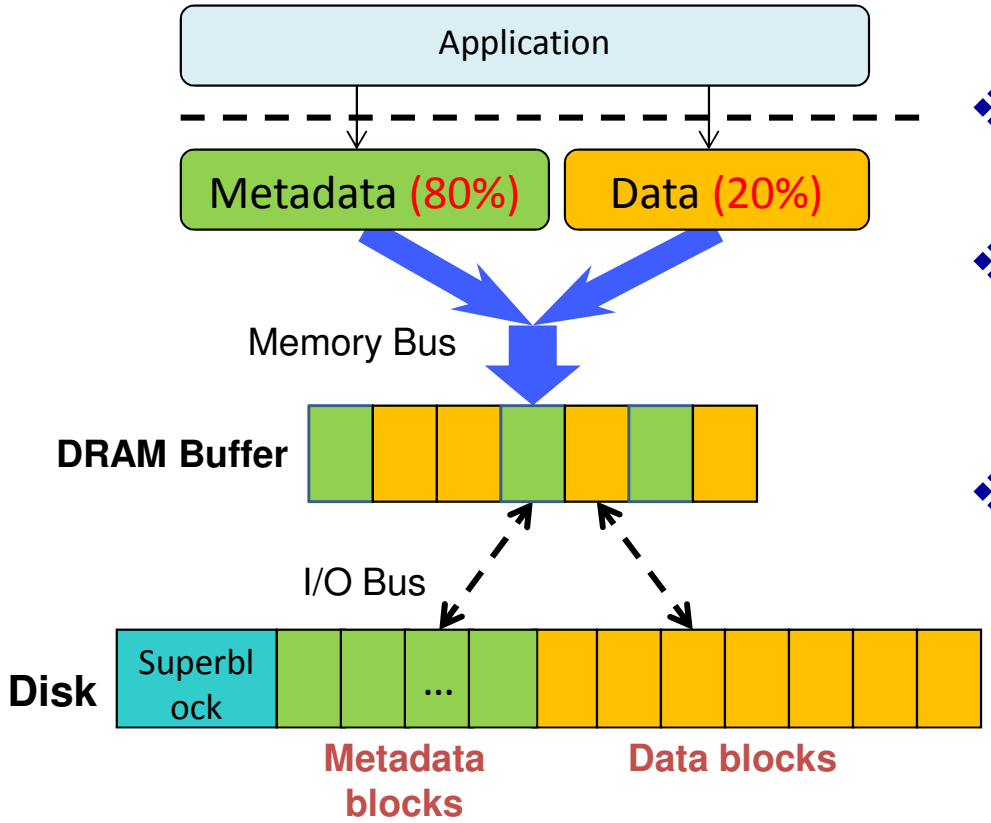
Agenda

- Introduction
- FSMAC: Design of A File System Metadata Accelerator using NVM
- System Implementation
- Evaluation
- Conclusion

1. Introduction

- File system metadata
  - Popular: which is about **60%-80%** of I/O accesses.
  - Small and random

Workloads	Data I/O(%)	Metadata I/O(%)
Varmail	15	**85**
Webserver	44	**56**

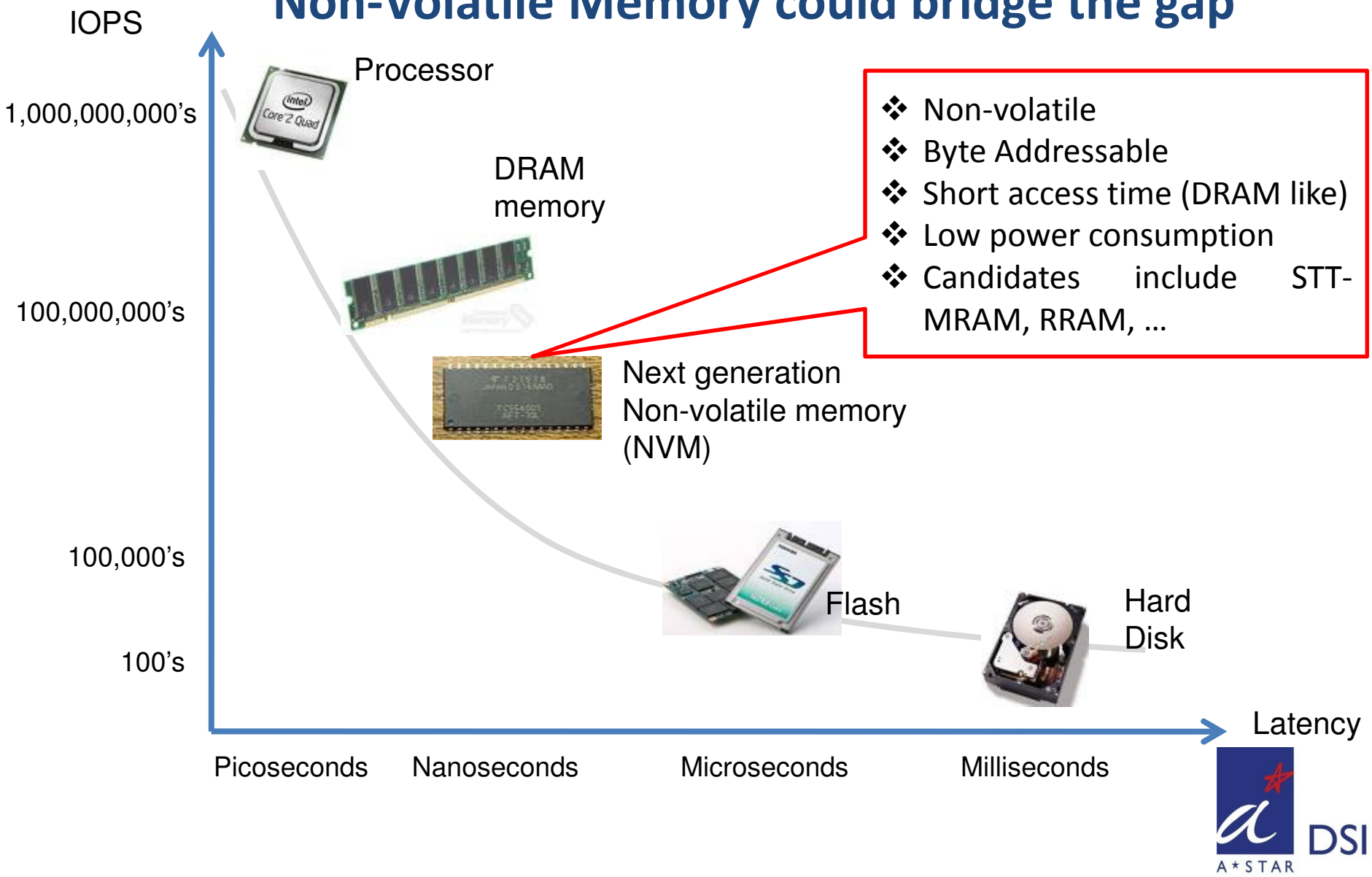


- **Metadata and data are stored /accessed in block.**
- **Metadata and data share and compete for I/O resource.**
- **Lots of fetch and flush between DRAM buffer and disk.**

A*STAR DSI

1. Introduction

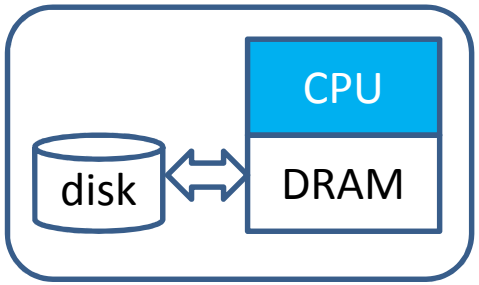
Non-Volatile Memory could bridge the gap



DSI
A*STAR

1. Introduction

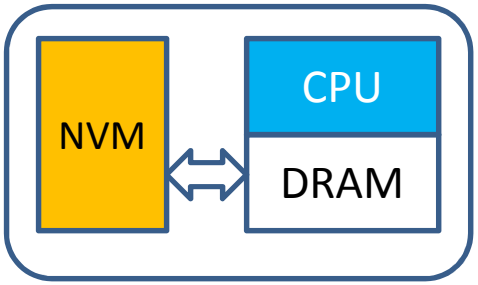
NVM System Integration



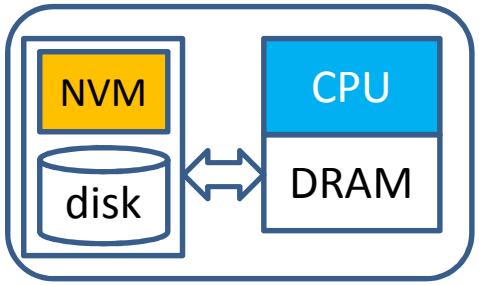
a, Current system

Slow IO bus
Long data path

NVM as Block Device



b, Replace disk

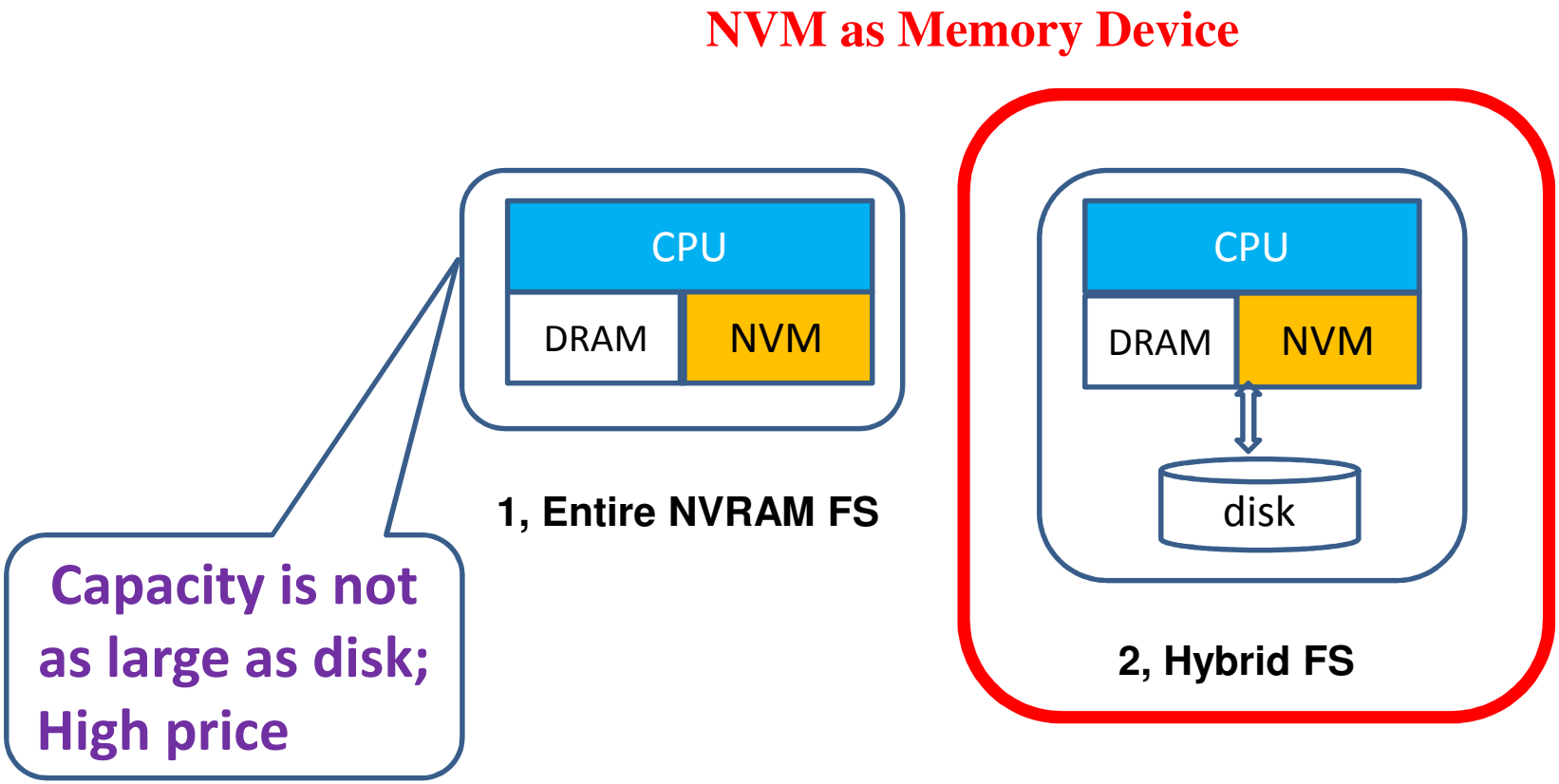


c, Hybrid disk

A*STAR DSI

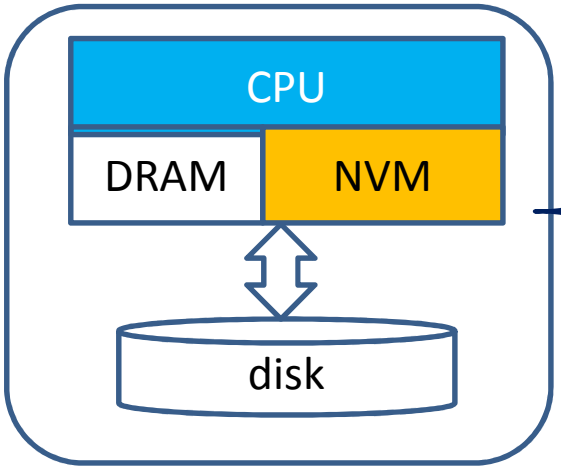
1. Introduction

NVM System Integration



1. Introduction

Current Solutions on Target System



- **NVM as persistent storage:**
  Data and metadata are separated,
  data on disk, metadata on NVM
  incompatible, difficult to porting FS data

- **NVM as buffer or non-volatile buffer:**
  replaced frequently → **lots disk I/Os**
  buffer pollution → **low performance**

DSI
A*STAR

1. Introduction

Our solution——NVM as Metadata Accelerator

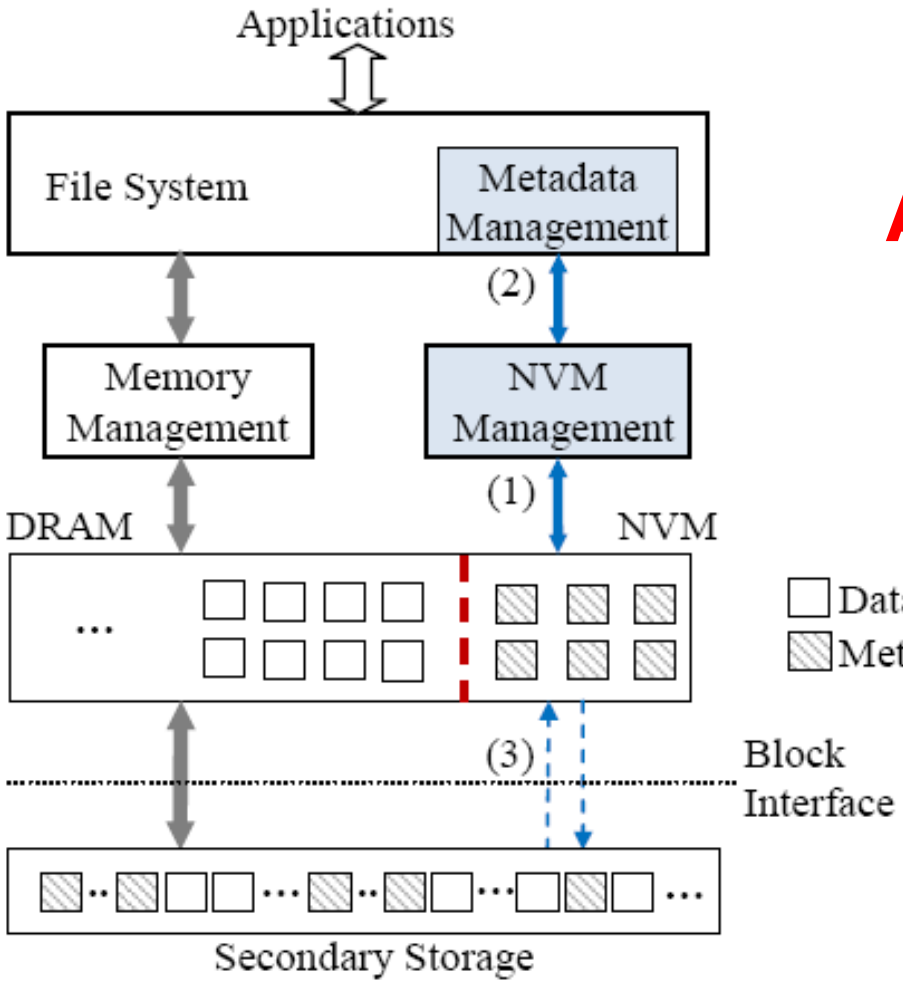
- NVM is used as both persistent storage and buffer for file system metadata
  - As storage for metadata, no necessary to write dirty block to disk **at runtime**, but keep original version on disk, compatible and portable
  - As buffer: in-memory computing, updating in-place, byte-addressable, but no replacement except deleting.

Compatible and portable,

Avoid frequent small random metadata IO

A*STAR DSI

2. Design of FSMAC



Architecture:

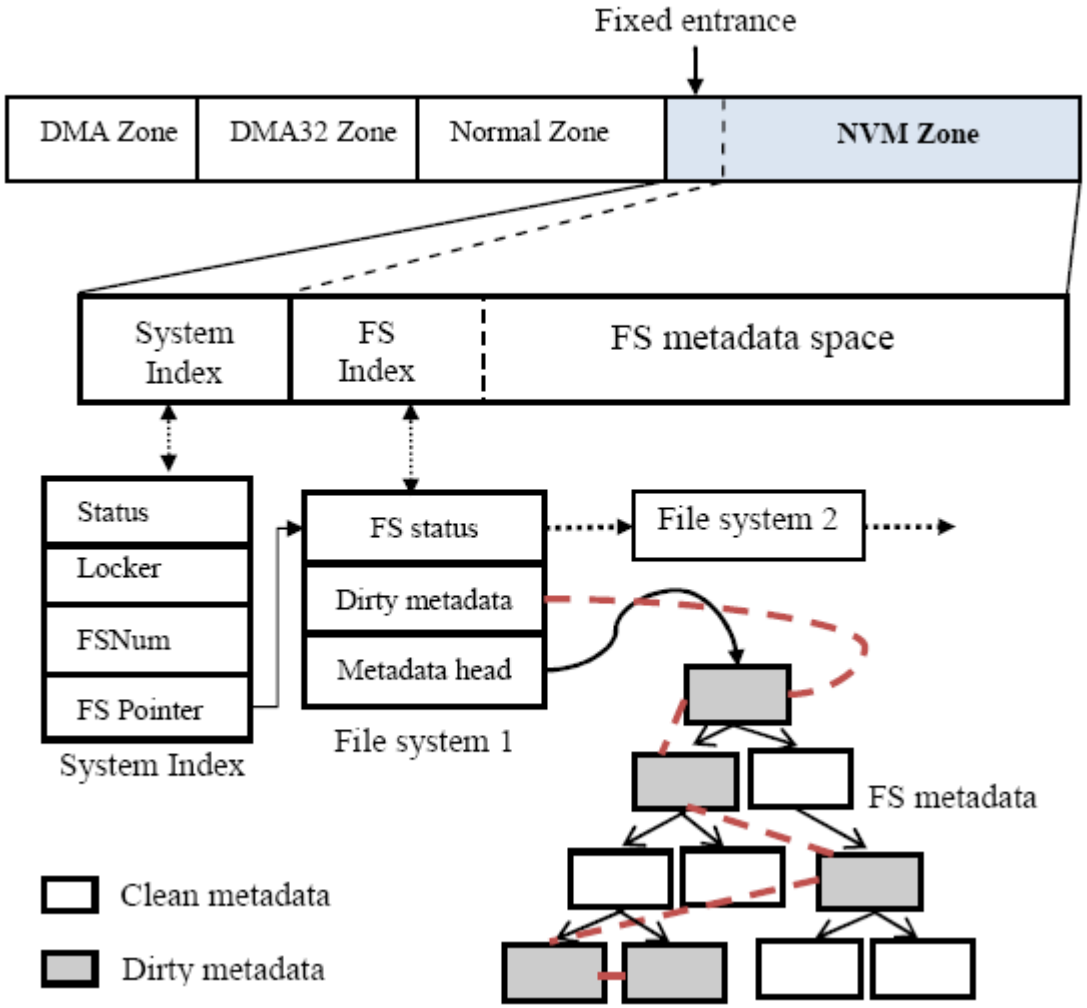
(1)NVM management

(2)Metadata management

(3)Consistency mechanism & metadata Sync.

A*STAR DSI

2. Design of FSMAC



NVM & metadata management

(1)A dedicated NVM Zone only for FSMAC

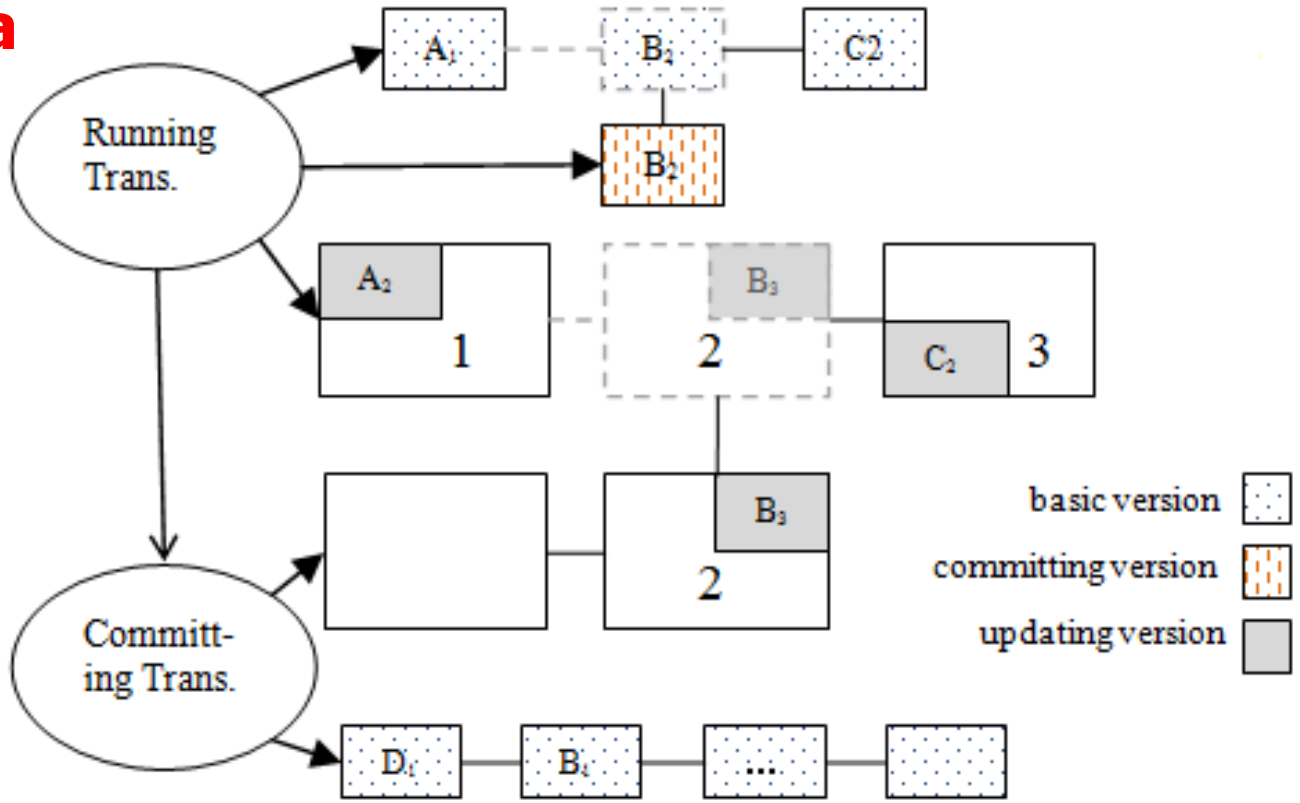
(2)Fixed entrance to reach all metadata

(3)Status for recovery

A*STAR DSI

2. Design of FSMAC

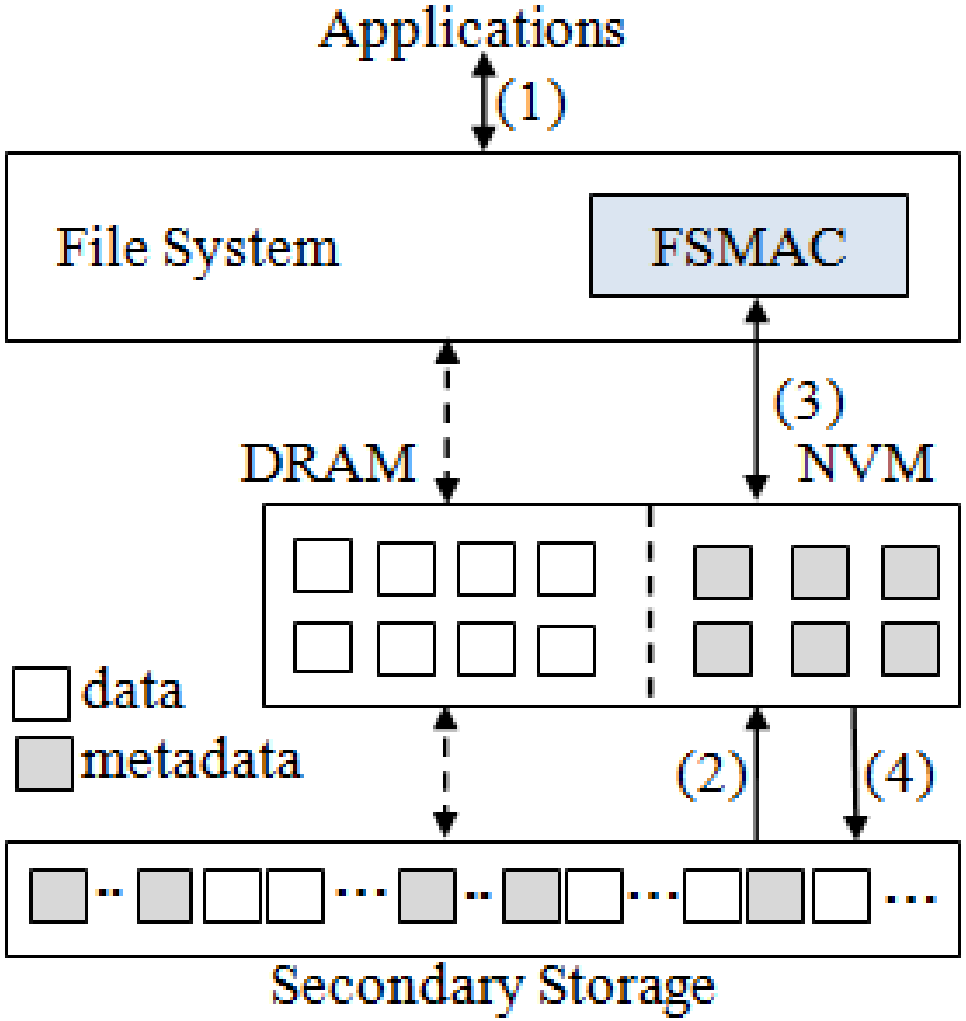
Maintain data consistency



- Block is divided into fine-granularity sub-block
- Sub-block is duplicated on the first update
- Updating on committing sub-block, additional copy is reserved

A*STAR DSI

2. Design of FSMAC



Decouple metadata and data path:

(1) Request reaches and is processed normally

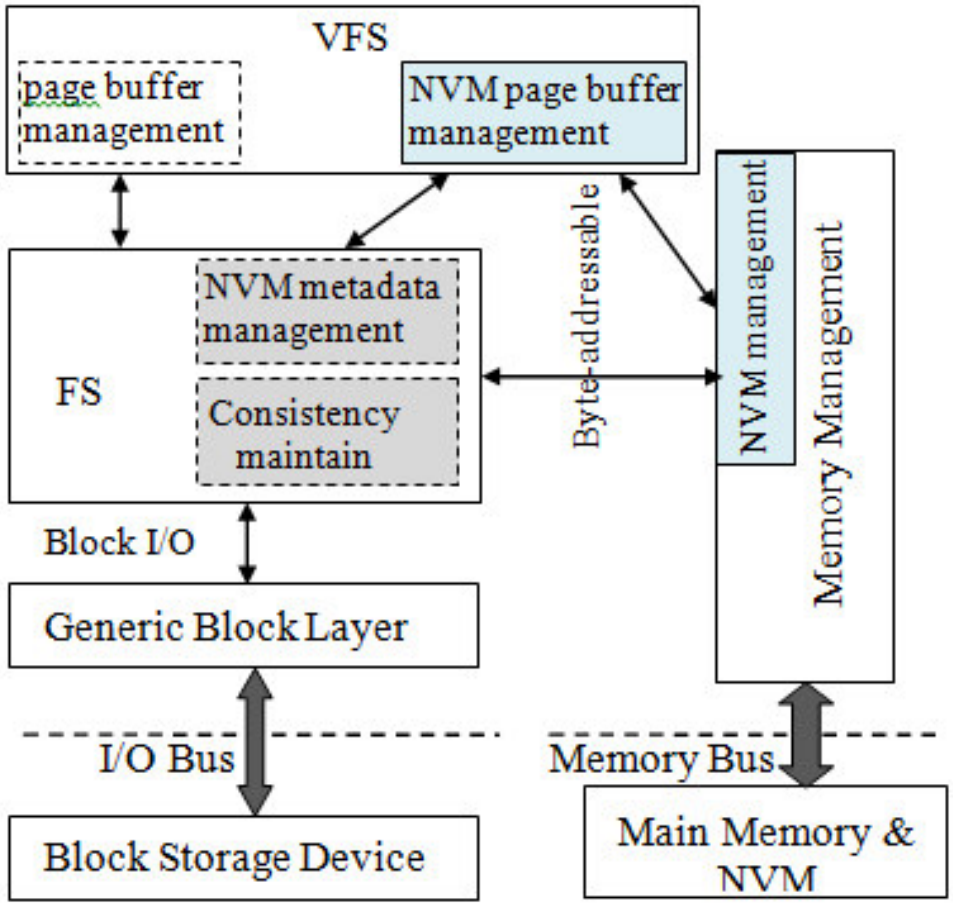
(2) Load metadata to NVM, never write back till unmount

(3) Flush out at unmount time with options

Data operation path is not changed

A*STAR DSI

3. System Implementation



- **Two modules are added:**
  - NVM management
  - NVM page buffer management
- **Two modules are modified:**
  - metadata management
  - consistency maintain

A*STAR DSI

4. Evaluation

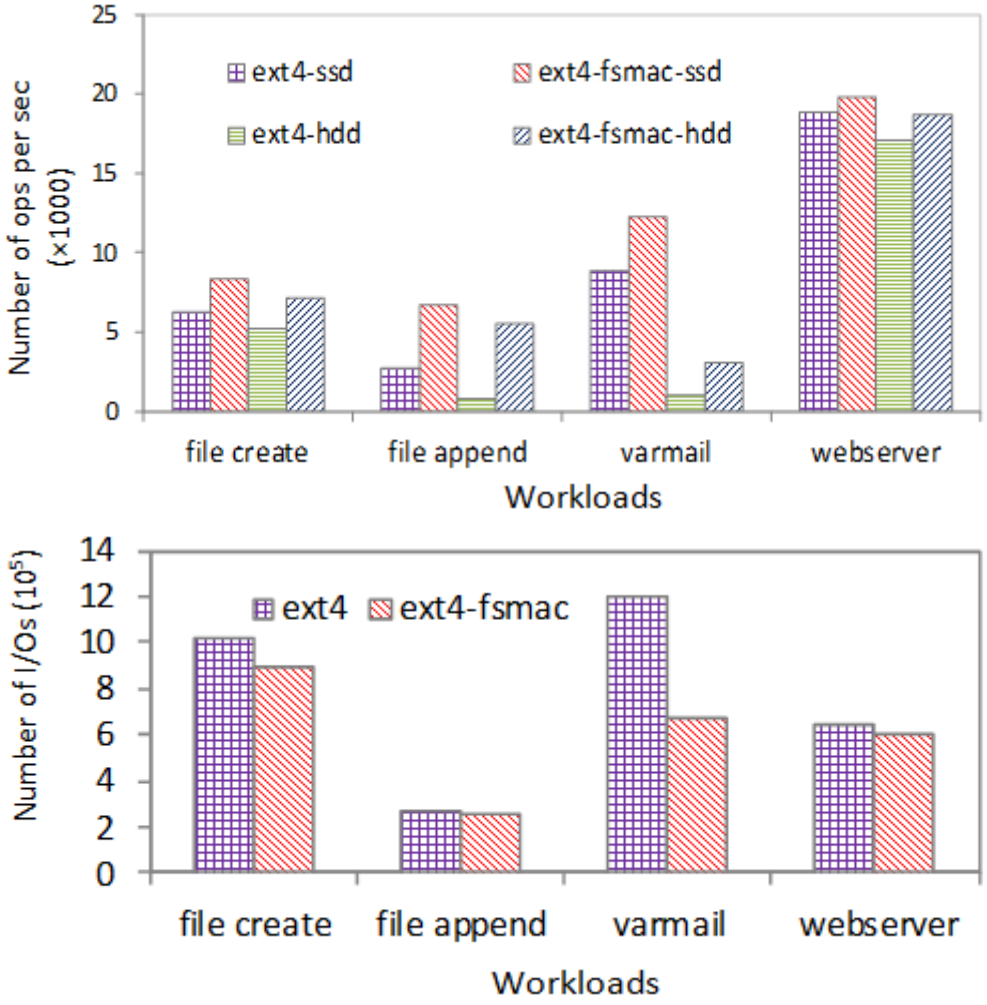
Evaluation Setup:

- Intel duo-core CPU
- DRAM simulated as NVDIMM (NVM)
- Intel 64GB SSD with SATA Interface
- 160GB HDD with PATA Interface
- Linux kernel version 2.6.34
- ext4 V.S. ext4 with accelerator (ext4-fsmac)

A*STAR DSI

4. Evaluation

(1) Filebench

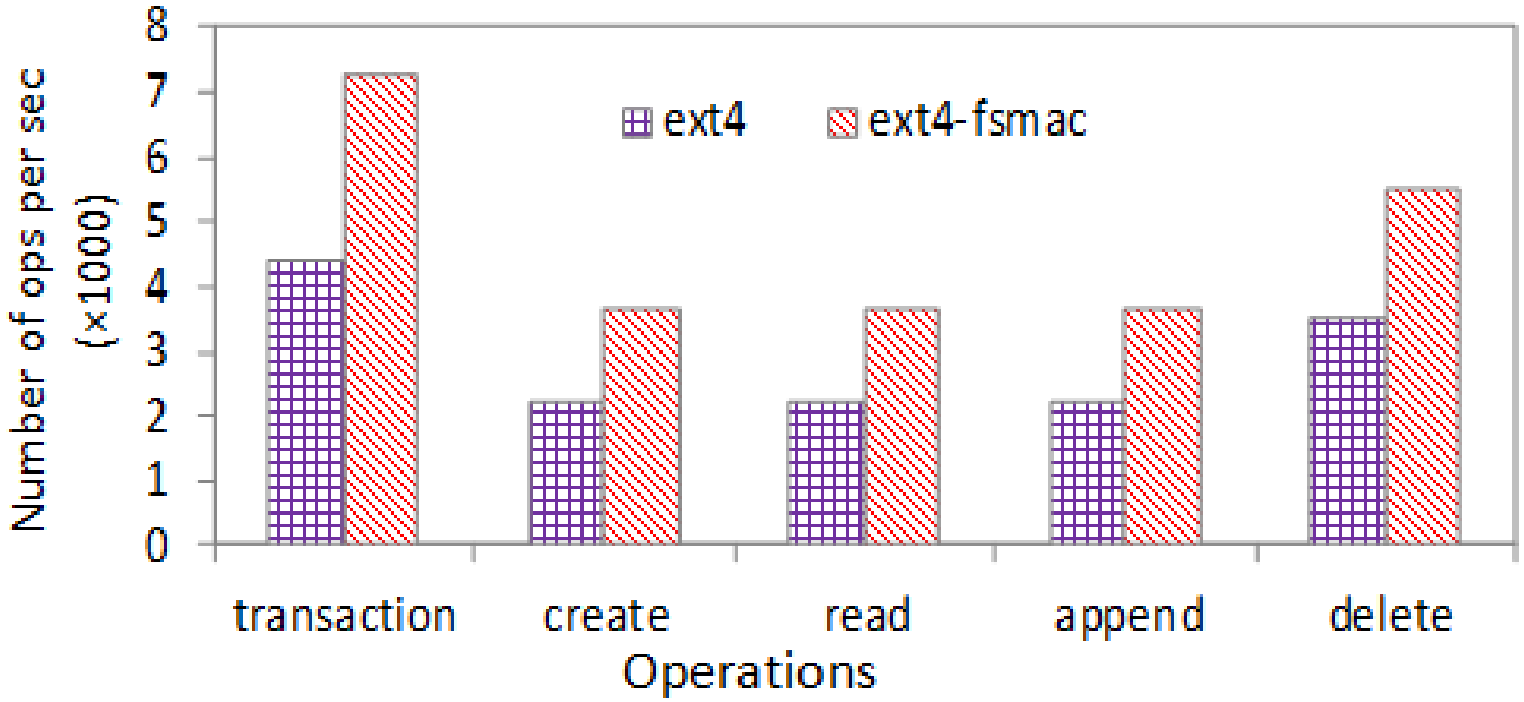


- FS on SSD and HDD are evaluated
- On SSD, performance improved from 5.4% to 147.4% in different workloads
- On SSD, performance improved from 10.9% to 621.7%
- Disk IO reduced (see second plot)

A*STAR DSI

4. Evaluation

(2) PostMark

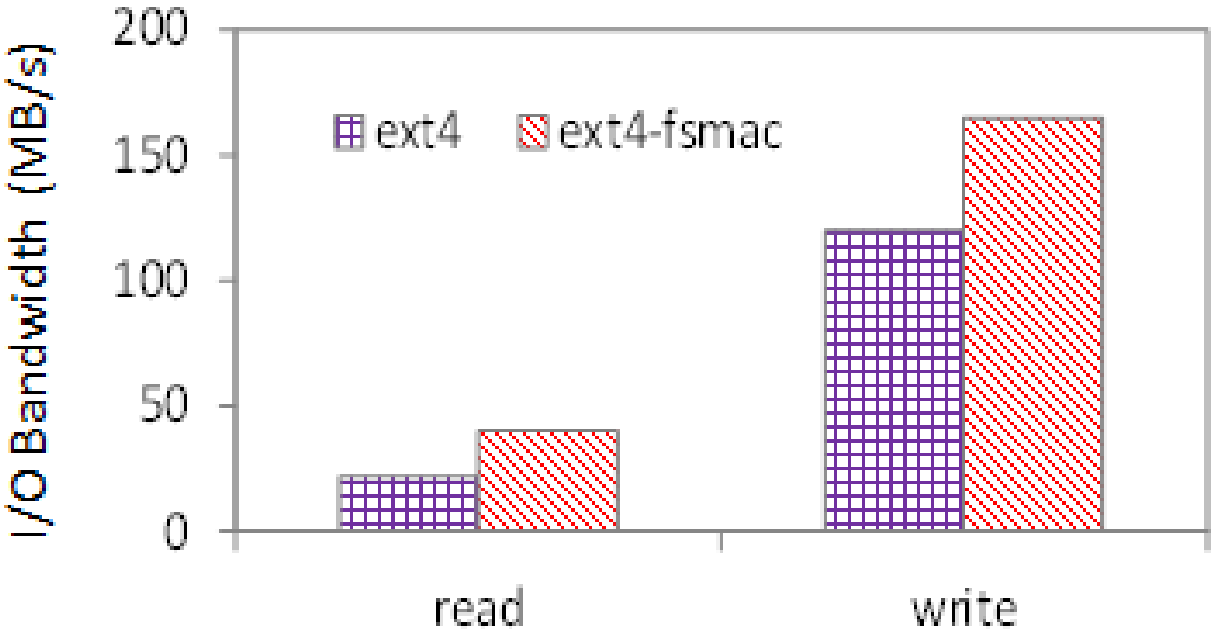


❖ **1.67× performance speedup achieved**

A*STAR DSI

4. Evaluation

(3) FFSB

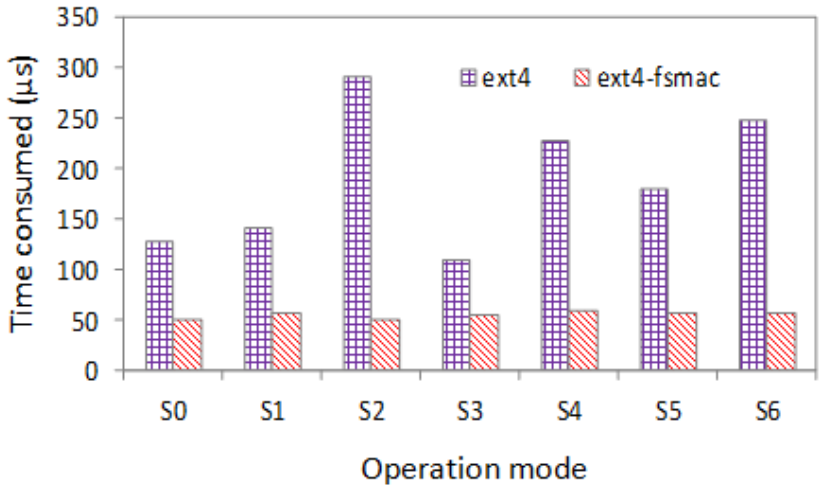


- **Ext4-FSMAC improves read performance by 82% and writes performance by 35%**

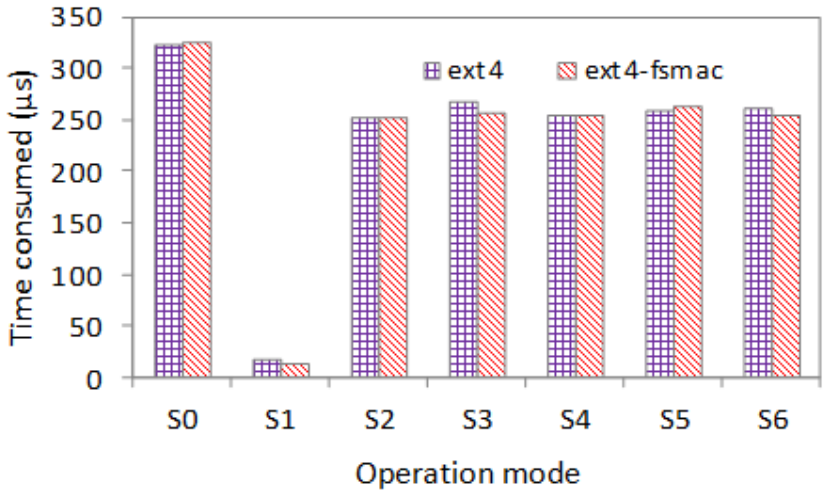
A*STAR DSI

4. Evaluation

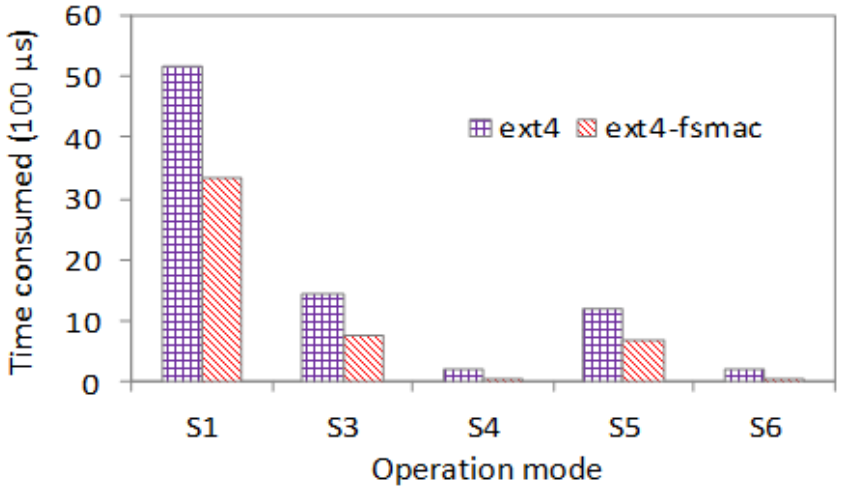
(4) Micro-benchmark (FS_Mark)



(a) Time for creation



(b) Time for write



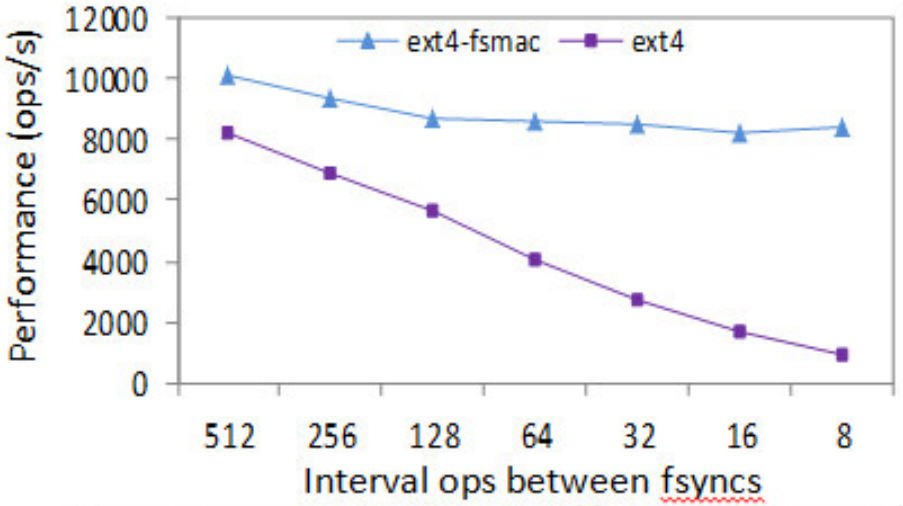
(c) Time for fsync

- ***Fcreate*** **and** ***fsync*** **are improved significantly**

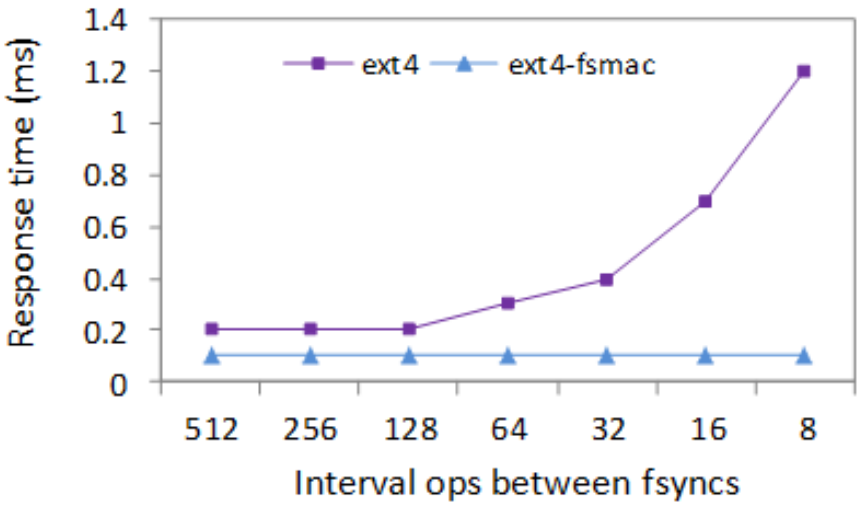
A*STAR DSI

4. Evaluation

(5) Effect of Synchronization Frequency



(a) Write operations per second by varying interval operations between 2 fsyncs.



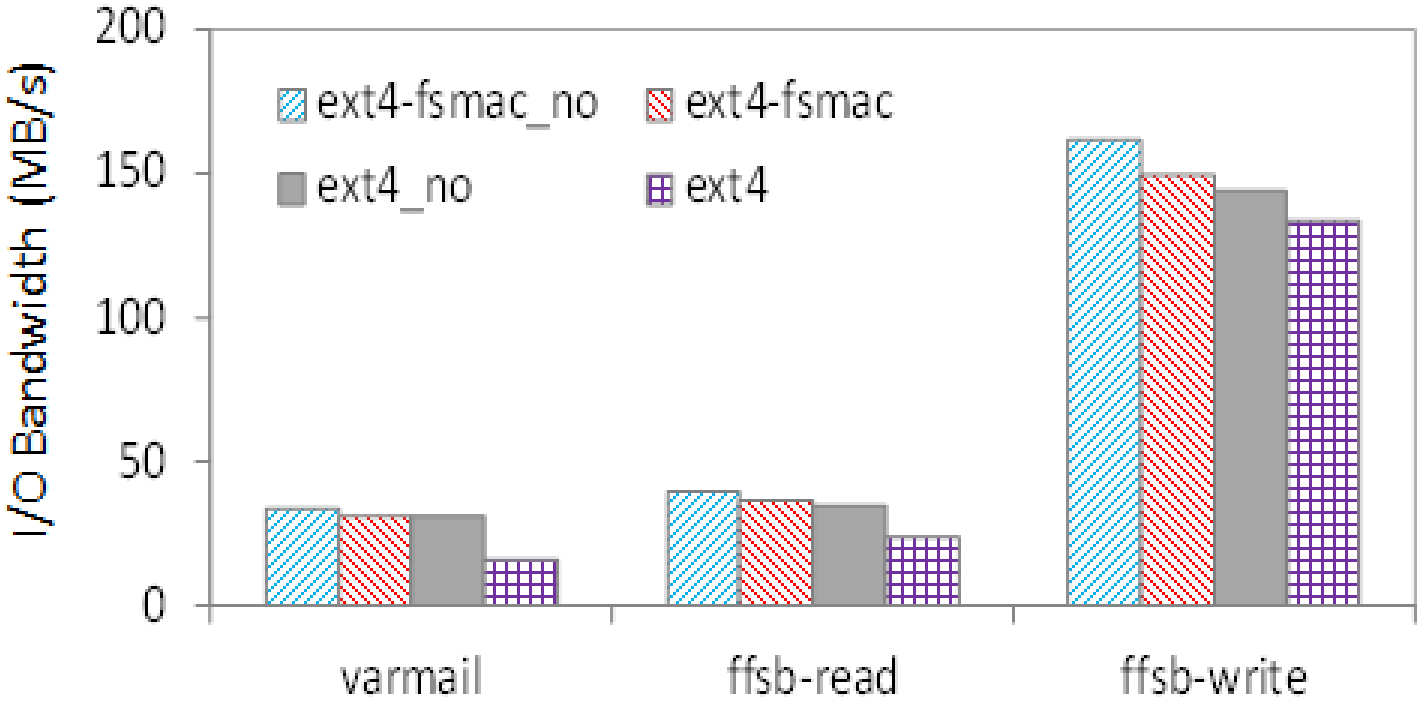
(b) Write response time by varying interval operations between 2 fsyncs.

- **frequent file synchronizations degrade Ext4 file system performance significantly but do not affect ext4-fsmac so much**

A*STAR DSI

4. Evaluation

(6) Overhead of Consistency Maintain

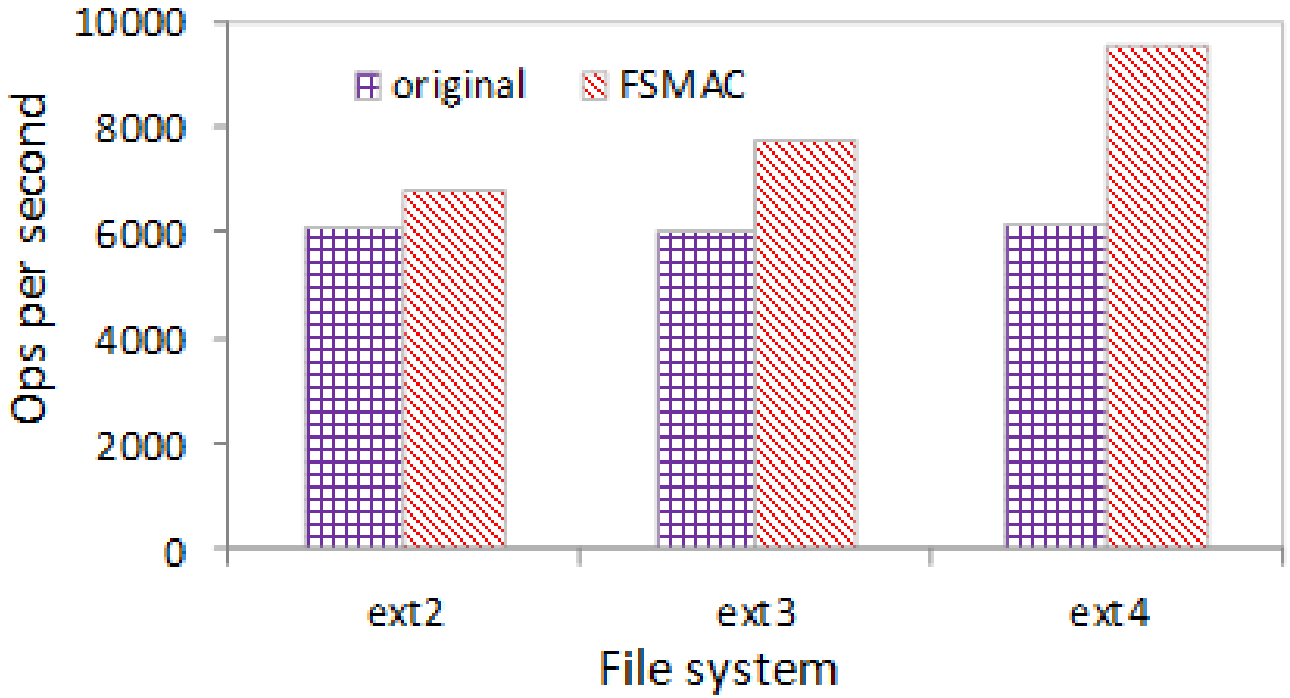


❖ **Overhead is much smaller than journal in ext4**

A*STAR DSI

4. Evaluation

(7) For Different File Systems (FFSB)



- The metadata accelerator works better on ext4 than on ext2 and ext3 file system

A*STAR DSI

5. Conclusion

- A file system metadata accelerator is designed by using NVM to accelerate metadata accessing
  - Metadata and data accesses is decoupled.
    - Metadata is stored in NVM and accessed in byte-addressable through memory bus
    - Data is stored in Disk and accessed in block manner through I/O path.
  - A method combined fine-grained metadata versioning and transaction mechanism is introduced to overcome the problem of consistency
- It accelerates ext4 file system
  - Up to 7.22 times for asynchronous I/O
  - Up to 49 for synchronous I/O

A*STAR DSI

THANKS!

For further query, please contact:
WEI_Qingsong@dsi.a-star.eddu.sg

A*STAR DSI