重慶大學
CHONGQING UNIVERSITY

Wear-aware Memory Management Scheme for Balancing Lifetime and Performance of Multiple NVM slots

Presenter's: Yujuan Tan

Authors: Chunhua Xiao, Linfeng Cheng, Lei Zhang, Duo Liu, Weichen Liu

目 录

- Background
- Motivation
- Design
- Evaluation

Background

PERSISTENT

BYTE-ADDRESSABILITY

LOW LATENCY

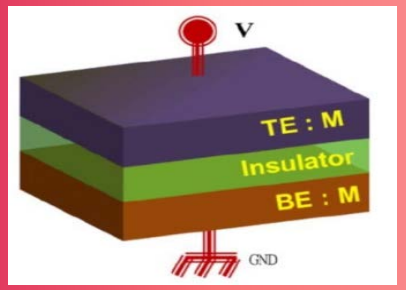
HIGH DENSITY

NVM

3DXPOINT

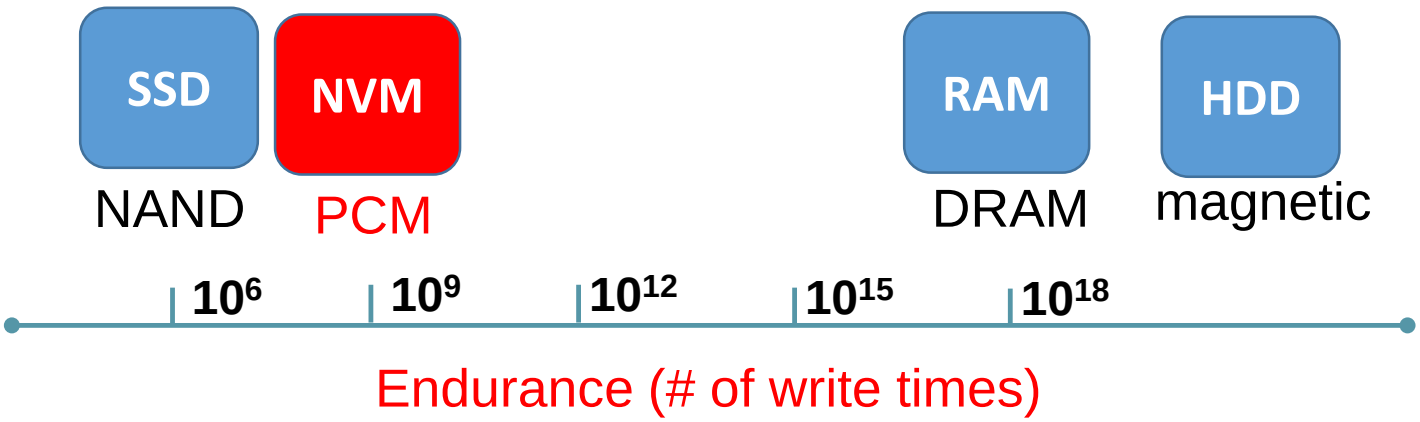
PCM

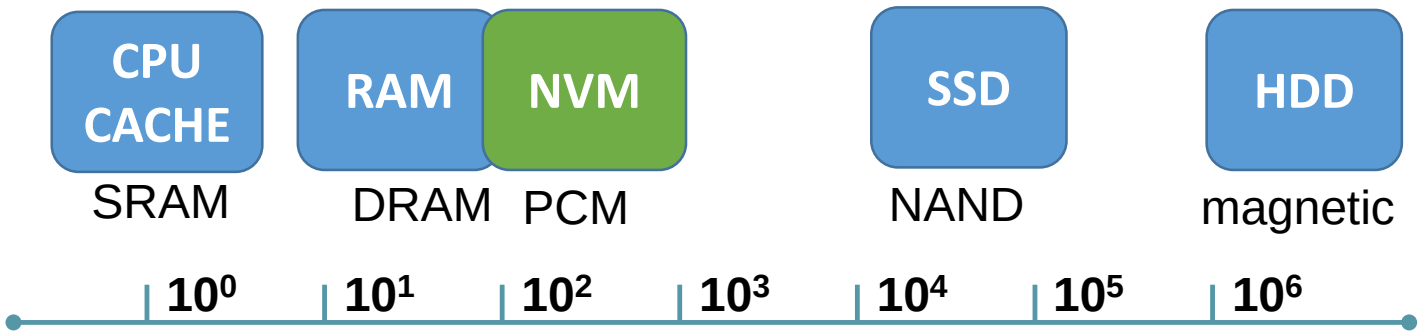
PRAM



Background

Lifetime limit of NVM



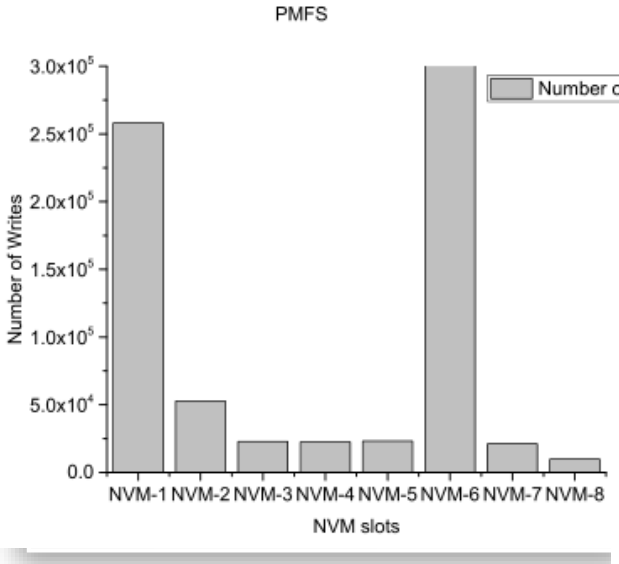


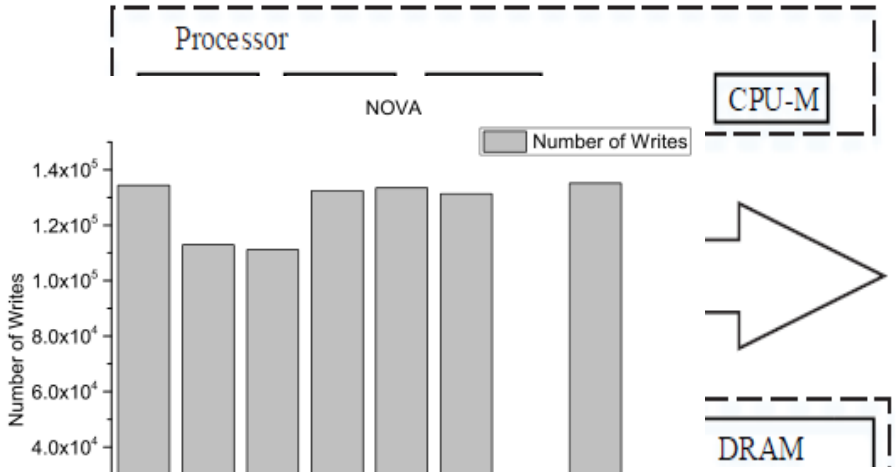
Motivation

Hardware Wear-Leveling technology for NVM

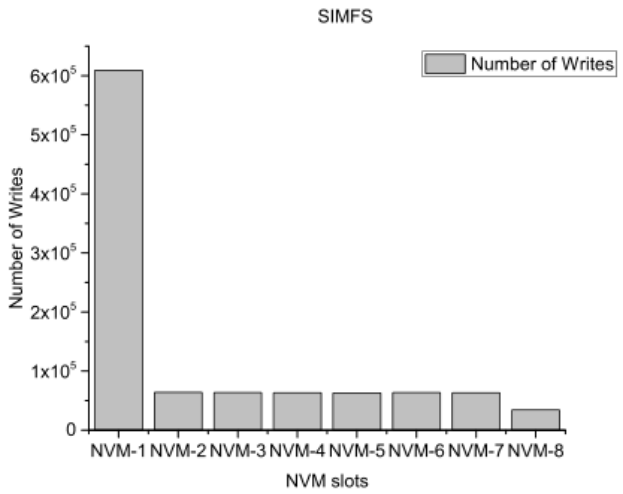
Software Wear-Leveling technology for NVM

Motivation



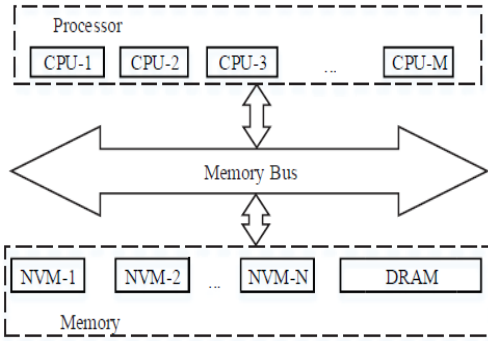


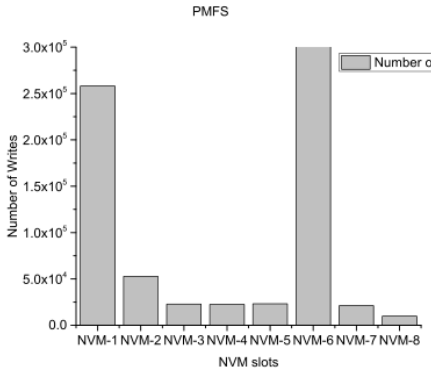
Single NVM

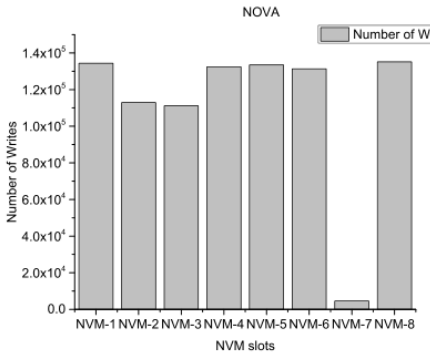


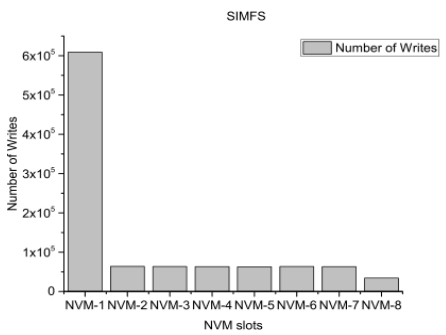
Software Wear-Leveling technology for NVM

Motivation









Hardware Wear-Leveling technology for NVM

Single NVM

Software Wear-Leveling technology for NVM

Innovation Point

Extend the lifetime for multiple NVMs

Memory Management for In-Memory File System

Collaborate with Hardware Wear-Leveling for single NVM

Design

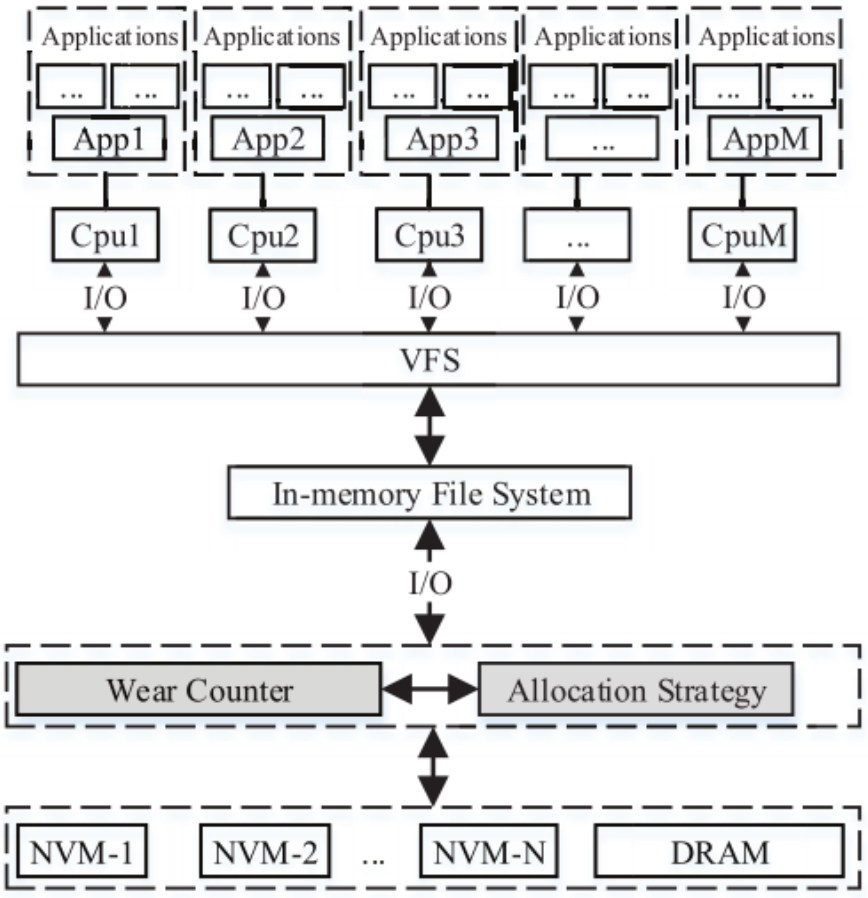
Wear-Counter

- Recording the number of writes of each NVM
- Dynamic aware the wear of each NVM and be used to adjust the use of each NVM

Allocation Strategy

- Considering concurrent workload pattern
- Enhancing lifetime and minimizes the cost of performance

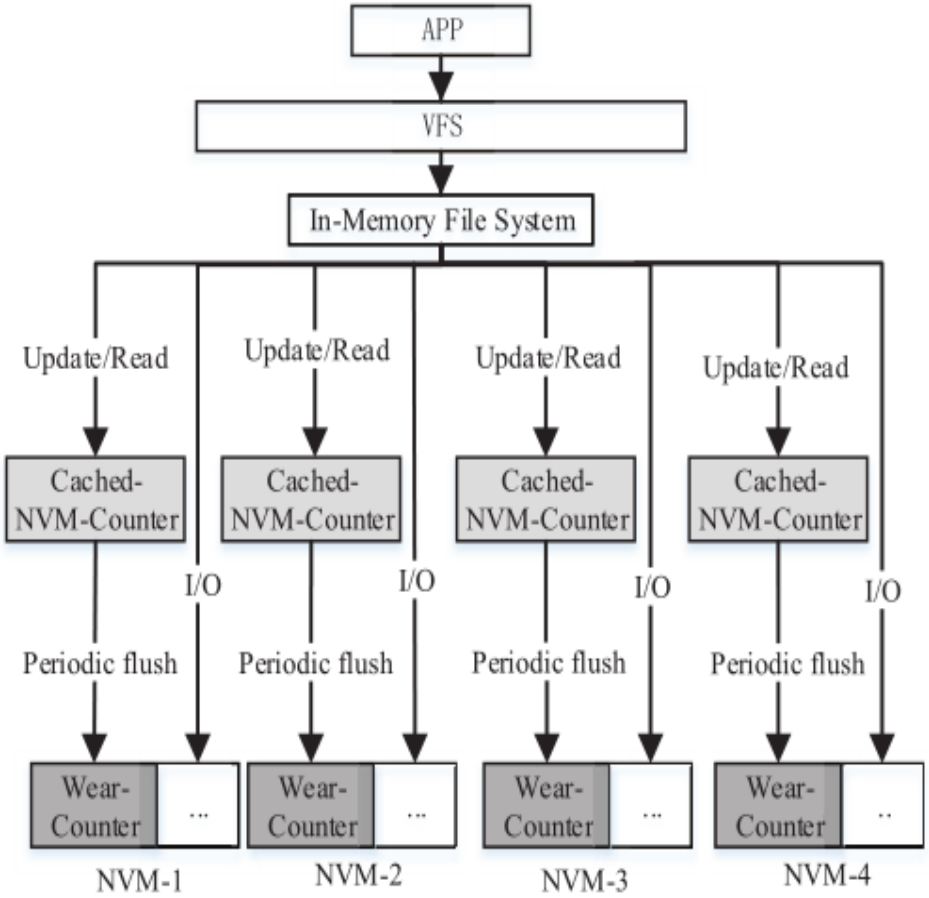
OVERVIEW



Design

Wear Counter

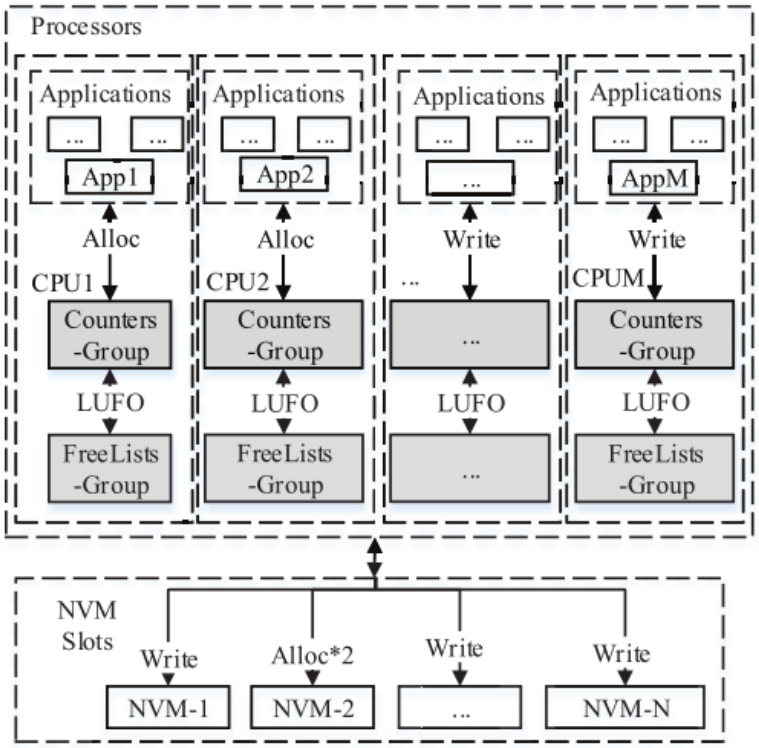
- **Record the total number of writes of each NVM.**
- **Insert auditing code in the file write path**
- **Write-Through (WT) strategy.**
- **Write-Back (WB) Strategy.**



Design

Allocation strategy

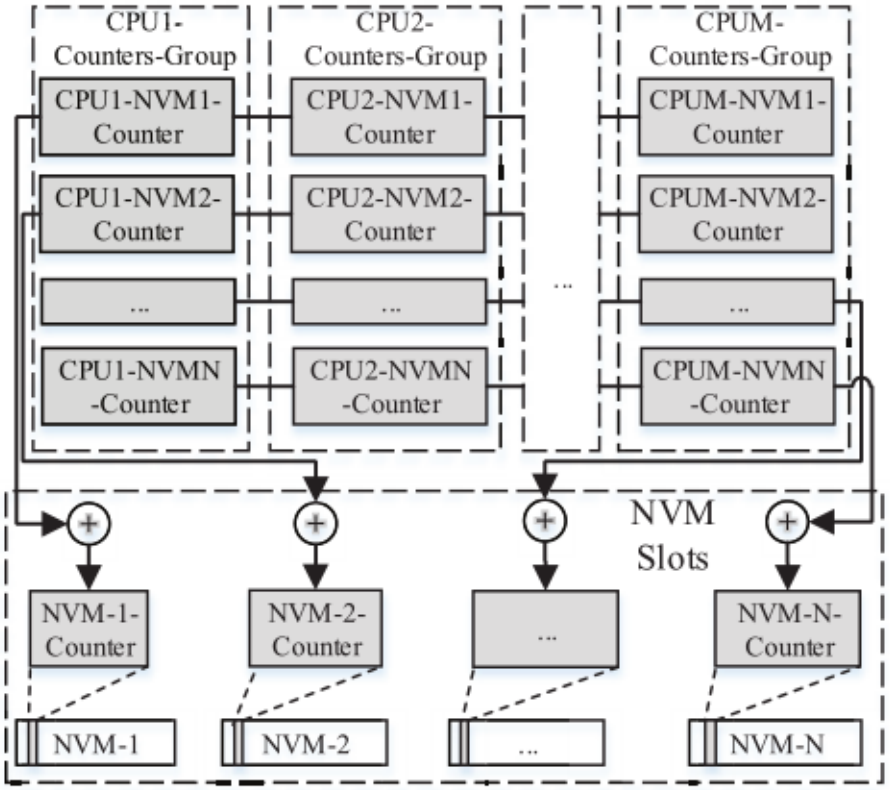
- **Per-CPU Free List Lease Use First**
- **Counters-Group**
- **FreeLists-Group**
- **Lease Use First Out strategy.**



Design

Counters-Group

- **m Counters-Group**
- **n global persistent NVM-Counter**
- **n local CPU-NVM-Counter**
- **Least use first strategy on each Counters-Group**
- **From local wear-leveling to global wear-leveling**

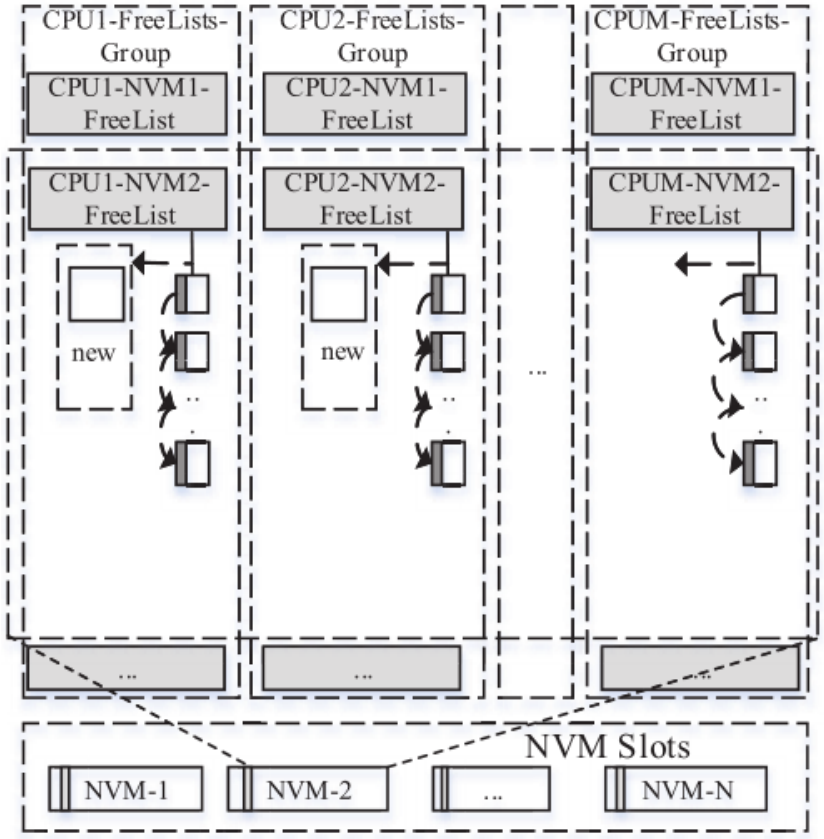


$$NVM_j - Counter = \sum_{i=0}^{m} CPU_i - NVM_j - Counter$$

Design

FreeLists-Group

- **m FreeLists-Group**
- **n local CPU-NVM-FreeList on FreeLists-Group of each CPU**
- **Each local CPU-NVM-FreeList is organized as a single linked list**



Evaluation

Configuration

参 数 项	具体配置
OS	Ubuntu 16.04
Kernel	Linux 4.4.4
CPU	Intel i5-6500, 4-Core, 3.20 GHz
DRAM	16GB DDR3 2133MHz DRAM
NVM	8GB DRAM to emulate 4 NVMs

Evaluation

Lifetime

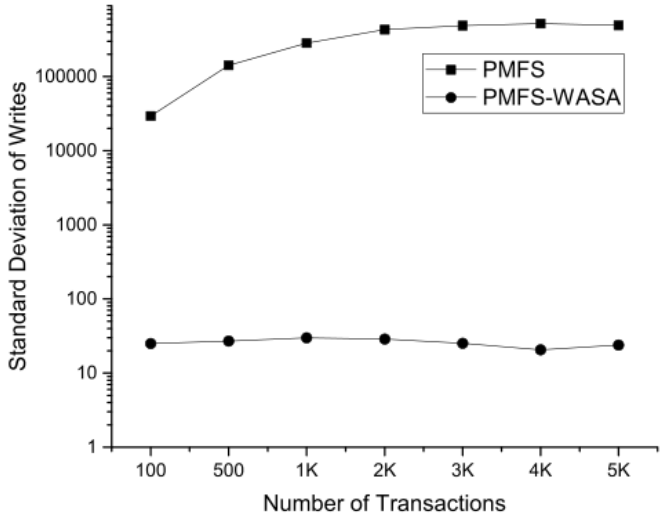
$$IN-WLE=\sqrt{\frac{\sum_{i=1}^{N}(C_i-\mu)^2}{N}}$$

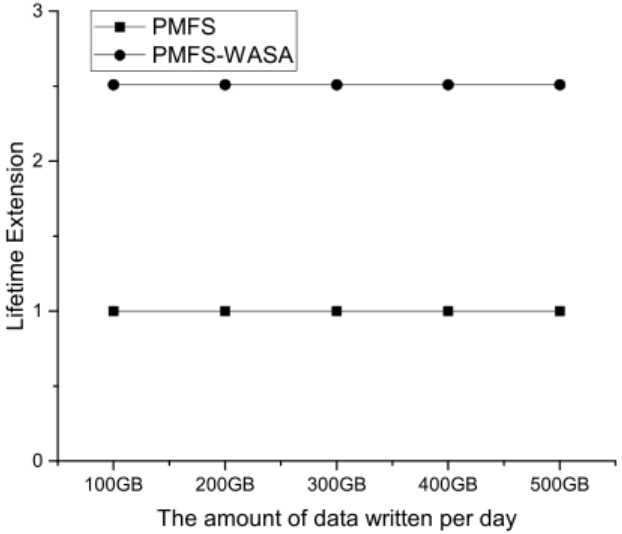
- **Outperform wear-leveling effect by more than 2600x**

$$Lifetime = W_{MAX} * S/B_{MAX}$$

$$B_{MAX} = NVM - Counter_{MAX} * W$$

- **The lifetime can be prolonged by 2.5x**

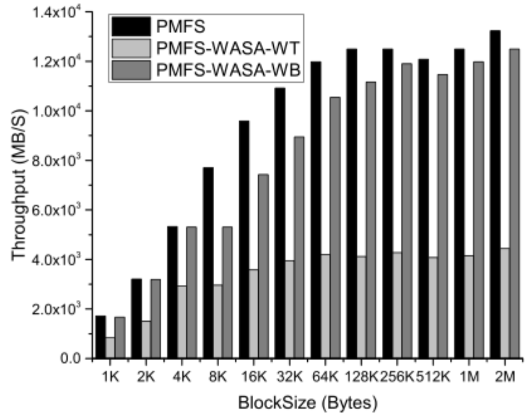


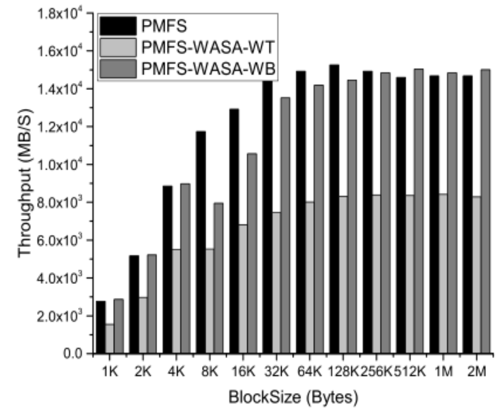


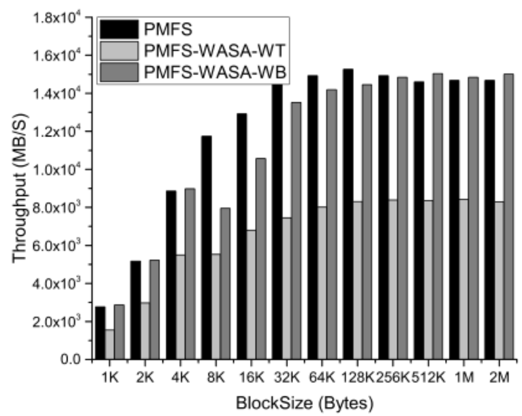
Evaluation

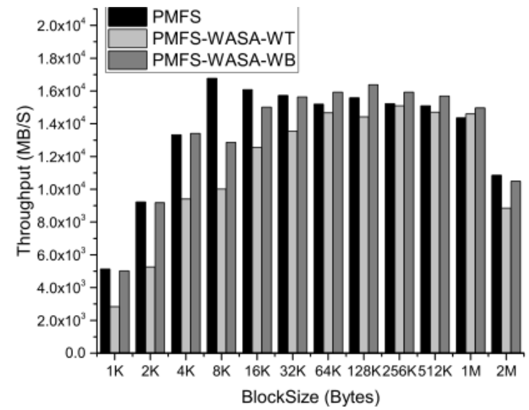
Performance

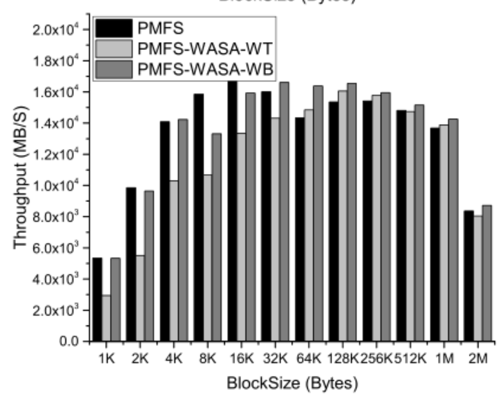
- **PMFS-WASA-WB basically shows the best performance in high concurrent workload.**
- **PMFS-WASA-WT has the worst performance.**
- **The performance is improved up to 15% when 16 threads and 64KB block size.**











重慶大學
CHONGQING UNIVERSITY

Thanks