A Zero-Copy Approach with Metadata-Driven File Management by Persistent Memory

Guangxing Hu

NC State University

OAK RIDGE National Laboratory

North Carolina State University - Computer Science

Oak Ridge National Laboratory (ORNL)

Advisors: Dr. Frank Mueller, Awais Khan

Introduction

- ➢ PM is a next-generation storage device that combines the properties of both volatile (like DRAM) and non-volatile (like SSDs) memory.
- ➢ This study aims to leverage PM 's byte addressability to optimize deep learning training processes, addressing **repeated reading from PFS** in traditional methods that involve multiple data copies.

Background

➢ Persistent Memory Capabilities

- **Intel Optane PM** offers DRAM-like speed with disk-like persistence.
- PM can be accessed via memory channels with high throughput.
- Previous work has not fully utilized PM's byte addressability.

➢Preliminary Study Findings

- PM outperforms SSDs in both random and sequential read/write operations.
- Devdax mode offers performance close to traditional system memory (RAM).

Motivation

➢ **Challenges in Deep Learning Training**

- Multiple data copies during training reduce efficiency.
- High cost and power consumption of DRAM.
- Inefficiencies in I/O operations due to frequent reads/writes from storage devices.

Reference

[1] Awais Khan et al. Hvac: Removing i/o bottleneck for large-scale deep learning applications. CLUSTER, 2022.

[2] Cheng Chen et al. Openembedding: A distributed parameter server for deep learning recommendation models using persistent memory. ICDE, 2023.

Methodology

➢ Zero-Copy Data Handling

- Cache data into PM during the first read/write operation.
- Use PM's byte addressability to avoid redundant operations.

•**POSIX Application Workflow**: File operation request retrieves metadata:

1. Metadata is processed, and file information is returned.
2. I/O redirection with LD_PRELOAD bypasses the traditional file system.
3. Data is directly accessed in PM via DAX-style access.

WorkFlow

➢ Normal Workflow:

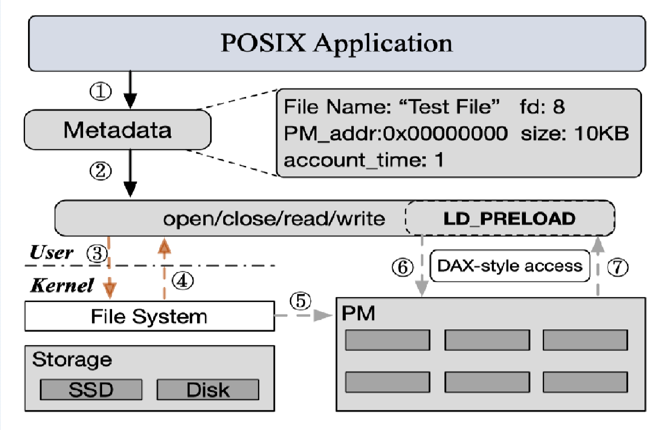
① ➔ ② ➔ ③ ➔ ④

➢ First-Time I/O Redirection Workflow:

① ➔ ② ➔ ③ ➔ ⑤ ➔ ⑦

➢ subsequent Access to the Same File Workflow:

① ➔ ② ➔ ⑥ ➔ ⑦



POSIX Application

①

Metadata

File Name: "Test File" fd: 8
PM_addr:0x00000000 size: 10KB
account_time: 1

②

open/close/read/write

LD_PRELOAD

User ③

Kernel ④

⑥ DAX-style access ⑦

⑤

File System

PM

Storage

SSD Disk

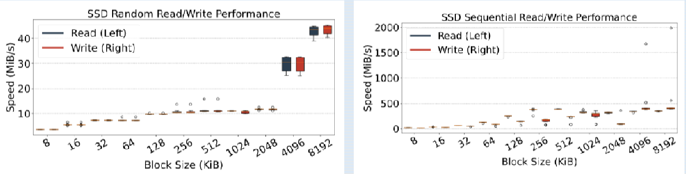
(a) IO Workflow Redirection

Conclusion

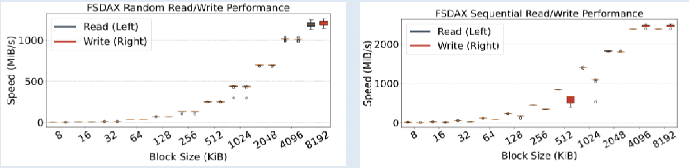
- ➢ We demonstrate the PM to optimize large-scale DL training jobs.
- ➢ By leveraging PM's byte addressability, we achieved zero-copy data handling, which significantly reduces I/O operations.
- ➢ Using PM in devdax mode offers performance comparable to system RAM, making it suitable for high-demand applications.

Preliminary Results

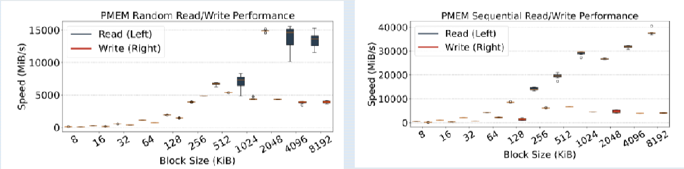
➢ Performance Comparison - Speed Tests:



(b) SSD Random and Sequential Read/Write

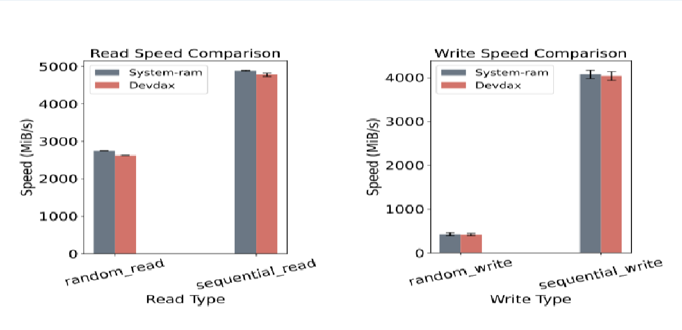


(c) Fsdax Random and Sequential Read/Write



(d) PM Random and Sequential Read/Write

➢ Performance Comparison - Devdax vs System-RAM



(e) Read/write speeds between System-ram and Devdax

➢ Takeways:

- The read/write speed of fsdax is up to ~6X that of an SSD, while PM is up to ~9X faster than fsdax.
- The devdax mode of PM achieves performance comparable to system RAM.