Wen-bing HAN, Xiao-gang CHEN, Shun-fen LI, Ge-zi LI, Zhi-tang SONG, Da-gang LI, Shi-yan CHEN, 2018. A novel non-volatile memory storage system for I/O-intensive applications. *Frontiers of Information Technology & Electronic Engineering*, 19(10):1291-1302. https://doi.org/10.1631/FITEE.1700061

A novel non-volatile memory storage system for I/O-intensive applications

Key words: In-storage processing; File system; Non-volatile memory (NVM); Storage system; I/O-intensive applications

Corresponding author: Xiao-gang Chen
E-mail: chenxg@mail.sim.ac.cn
ORCID: Wen-bing HAN, http://orcid.org/0000-0002-0370-7023

Introduction

1. The fast, byte-addressable, and non-volatile memory (NVM) technology, such as phase change memory (PCM), offers new chances for I/O-intensive applications.

2. Existing software stack and hardware architecture need to be optimized to exploit NVM performance.

3. In-storage processing (ISP) technology effectively reduces the number of I/O requests through moving computation into storage devices side.

4. This study proposes a novel PCM-based storage system, including a in-storage processing enabled file system (ISPFS) and a configurable parallel in-storage processing engine.

System architecture

1. PCM is directly attached to processors and can be accessed via CPU load/store instructions.

2. A configurable parallel computation fabric called ‘ISP engine’ is stacked on top of the PCM controller.

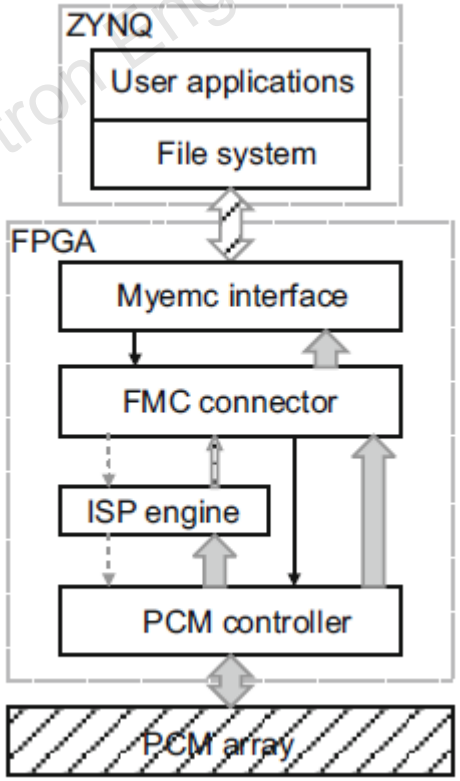

Fig. 1 Architecture of the proposed storage system

Design of ISPFS

1. ISPFS eliminates data copies through bypassing page cache and modifying page fault handler.

2. ISPFS stores instructions and results into ISP registers, which provides an ISP instructions channel.

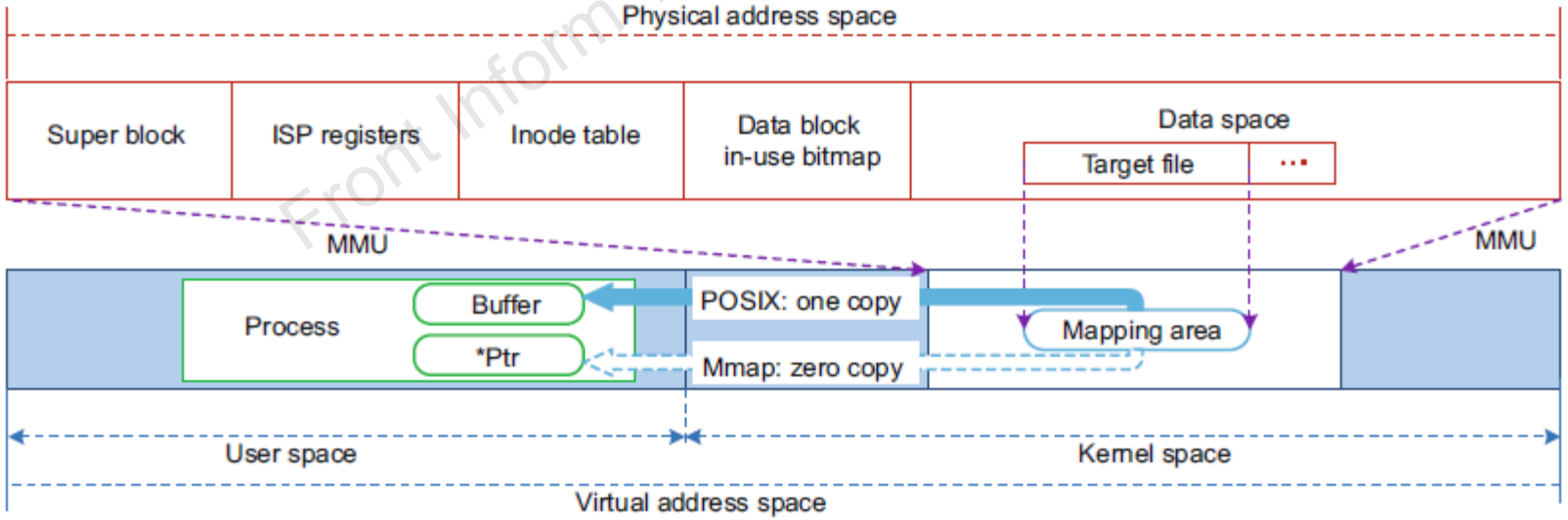

Fig. 2 Space layout of the in-storage processing enabled file system (ISPFS)

In-storage processing mechanism

1. A command file is created in every directory of ISPFS.
2. Applications can send ISP commands or receive ISP results by writing or reading command files.
3. ISPFS transforms the read/write operations of command files into ISP instructions.
4. ISP engine accesses file data and executes tasks according to ISP instructions.

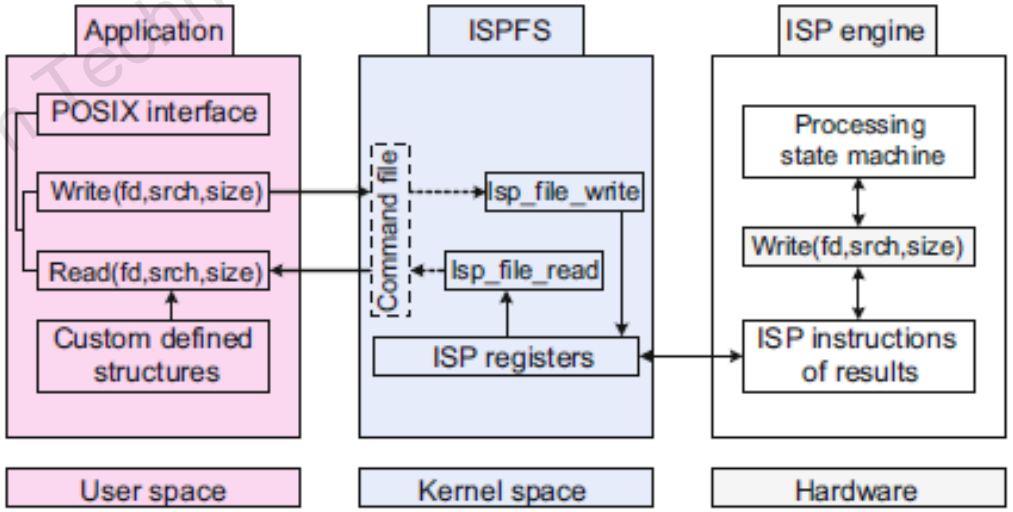

Fig. 3 In-storage processing mechanism

Major results

1. ISPFS achieves 2 to 10 times throughput compared to EXT4 file system.

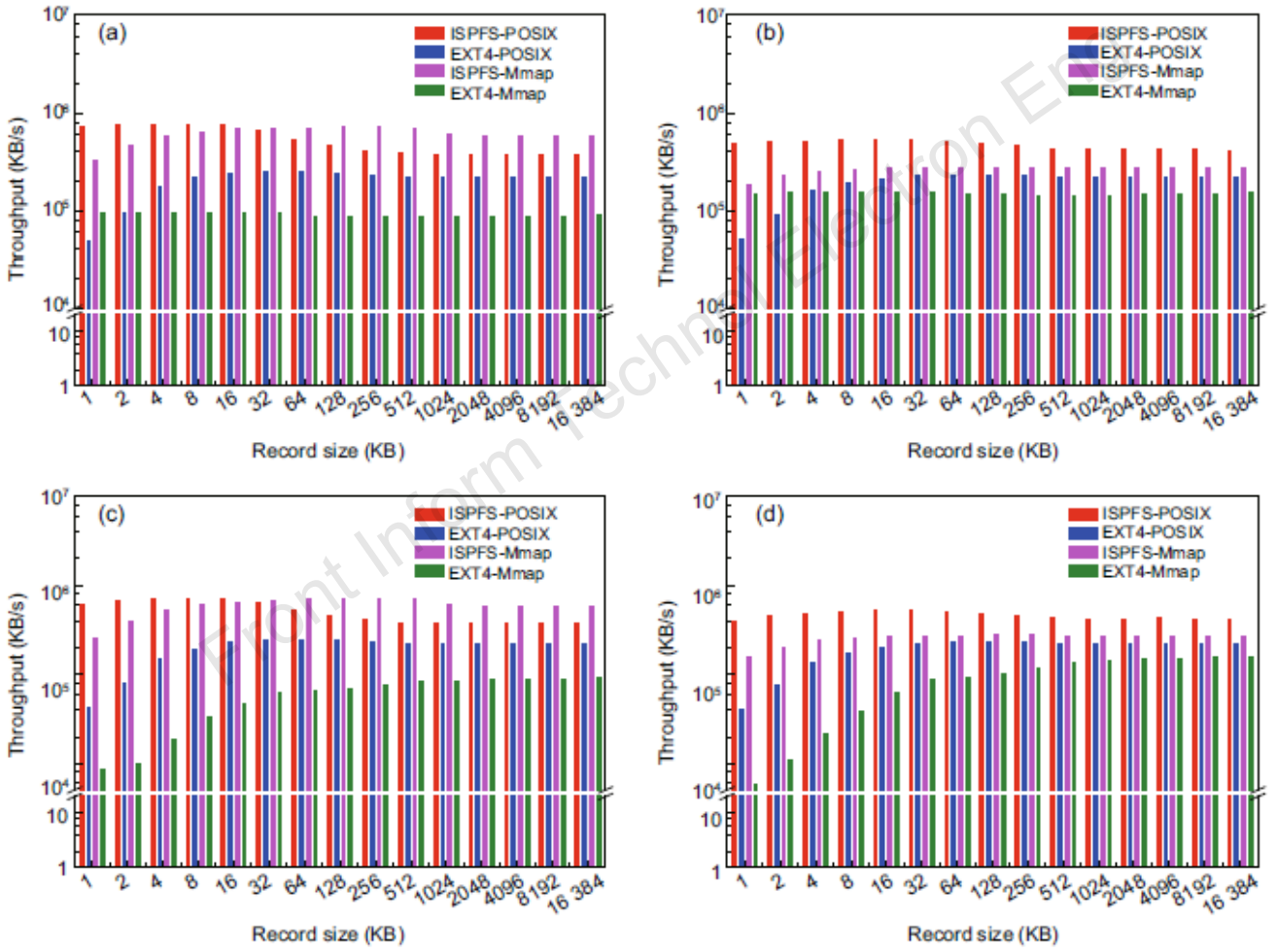

Fig. 7 Comparing throughput for ISPFS using IOZONE: (a) rewrite; (b) reread; (c) random write; (d) random read

Major results (Cont'd)

1. Our ISP solution is about 19 times more efficient than pure software solution and reduces I/O requests by 97%.

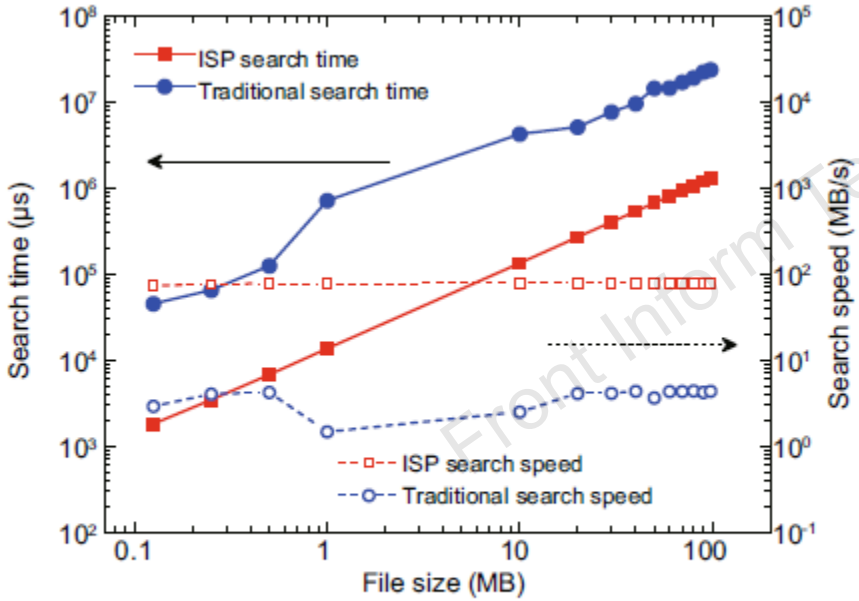

Fig. 8 Word query application performance of ISPFS and software solution

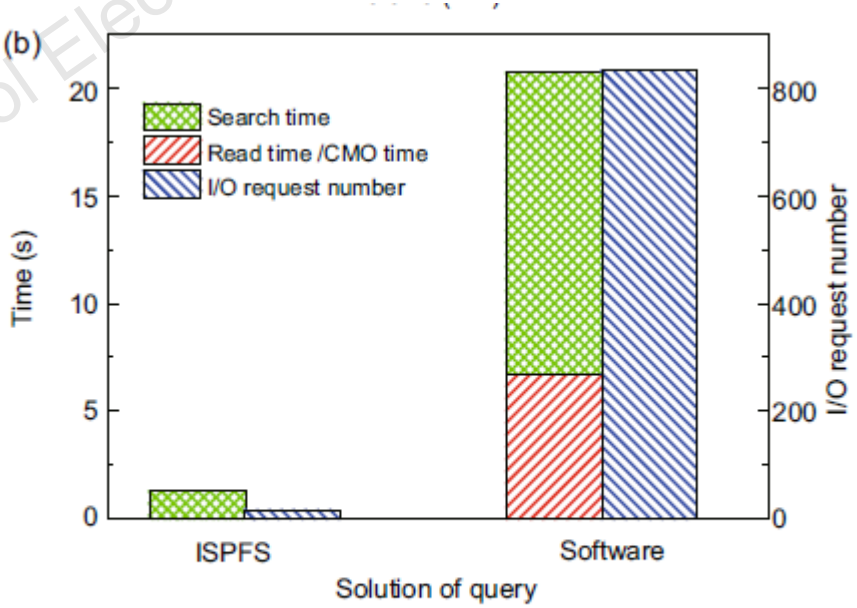

Fig. 9 The resource usage (a) and access time and I/O requests (b)

Conclusions

1. The proposed storage system incorporates in-storage processing technology into file system through a command file.

2. ISPFS enhances throughput by reducing data migration and software overhead.

3. Experiments demonstrate that the I/O efficiency of ISPFS solution outperforms pure software solution for I/O-intensive applications.