INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING
ICPP/2021/CHICAGO/USA
acm In-Cooperation
sighpc
AUGUST 9-12, 2021

SPMFS: A Scalable Persistent Memory File System on Optane Persistent Memory

Yang Yang, Qiang Cao

Huazhong University of Science and Technology

Background

- Non-Volatile Memory
  - Byte-addressability
  - Near-DRAM performance
  - Persistent
  - High density

- Intel Optane Persistent Memory (PMM)
  - The first commercial NVM in 2019
  - Performance not behave as expected

Background

- Intel Optane Persistent Memory
  - Performance asymmetry
    - 2.3GB/s write
    - 6.6GB/s read
  - Limited write scalability
    - Peak at 4 threads
    - Tailing off

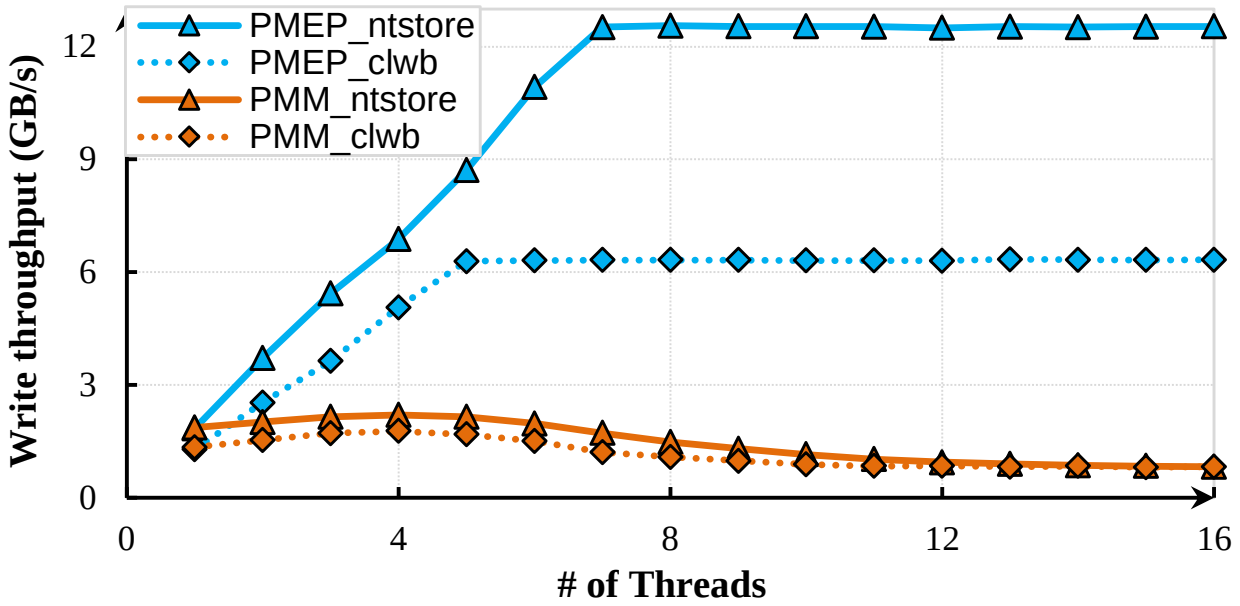


Fig. Write bandwidth vs. thread count on DRAM-based emulator PMEP and Optane PMM.

Background

- Persistent Memory File System
  - Ext4-DAX, NOVA, PMFS, Strata, libnvmmio.
  - Write scalability issue
    - ➢ Hardware bottleneck
    - ➢ Global resource management
    - ➢ Coarse-grained inode-level lock

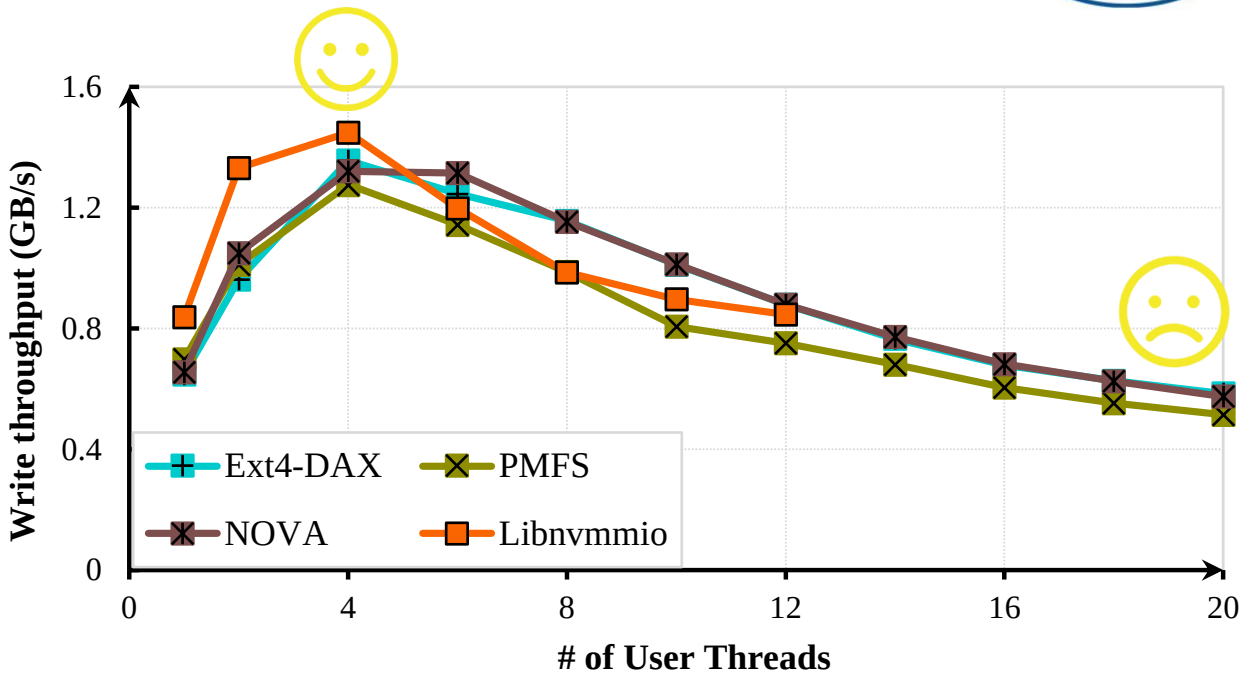
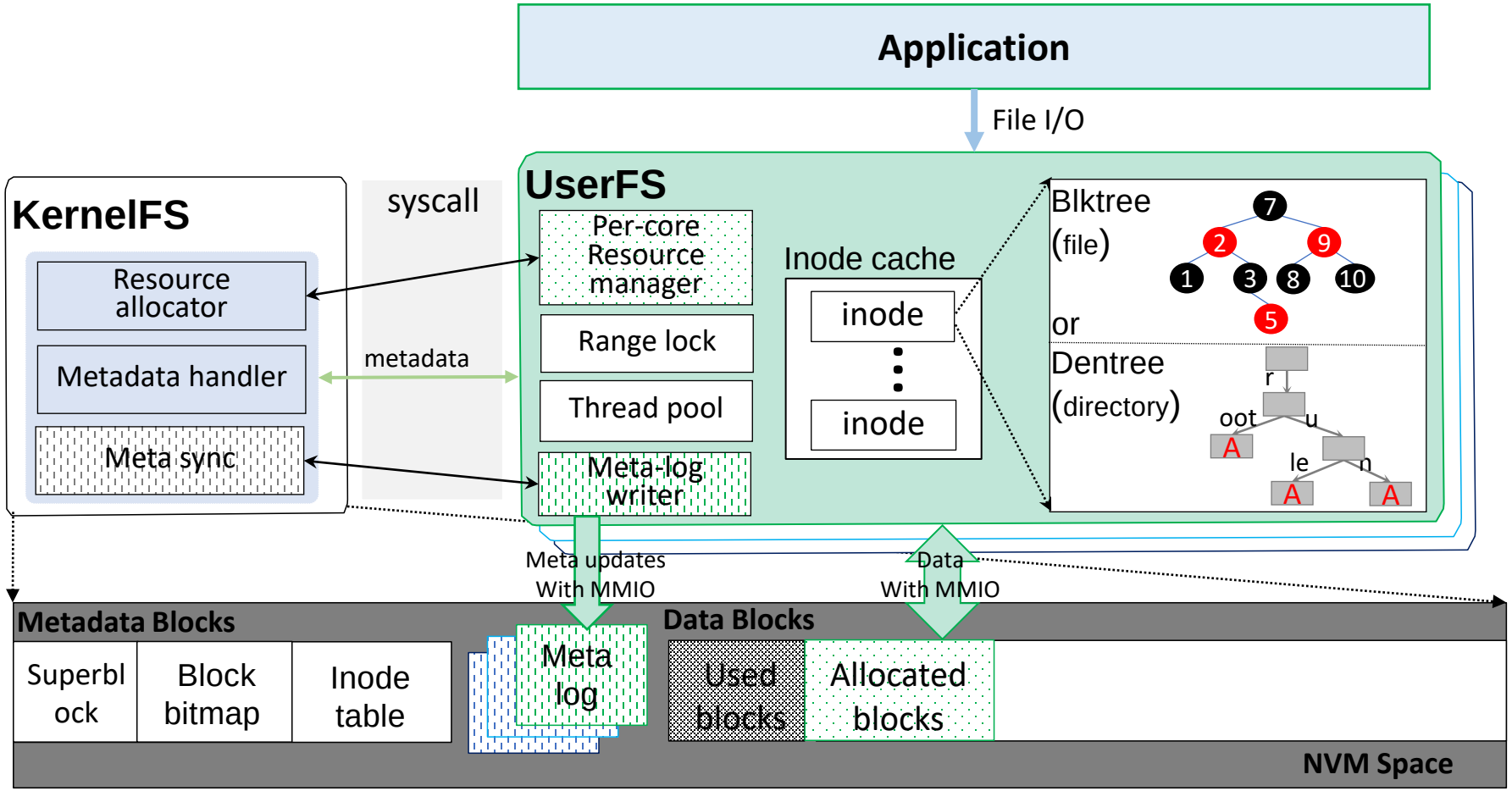


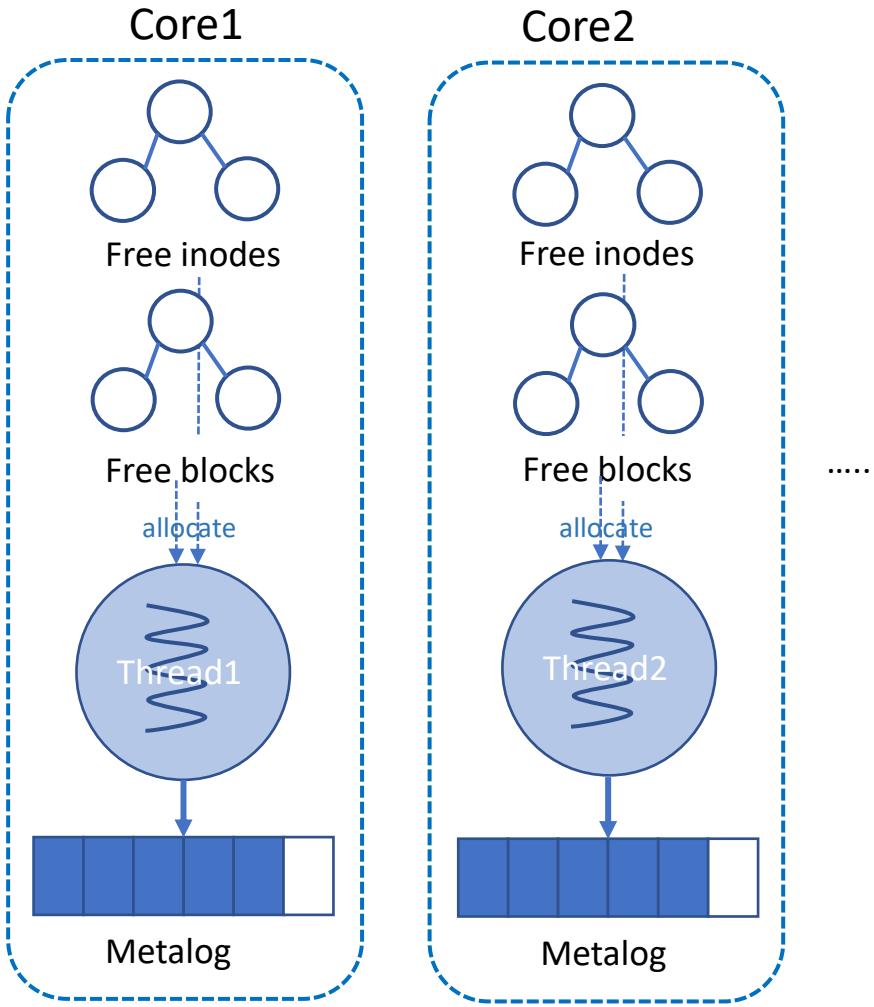
Fig. Write throughput (32KB IO) vs. thread count on NVM-based file system.

SPMFS Design



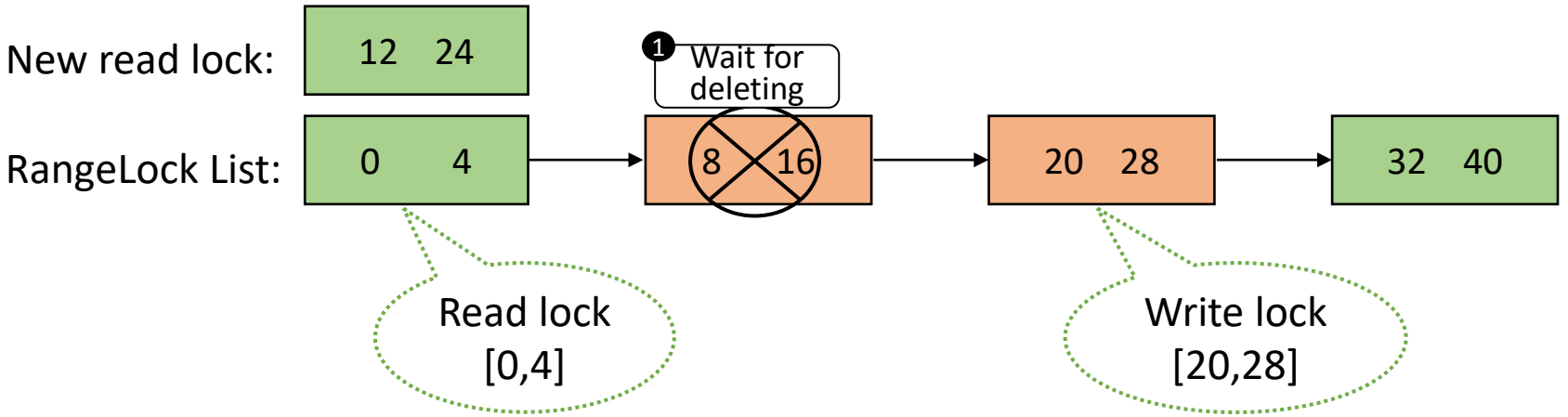
SPMFS Design

- Per-Core Resource manager
  - Per-core inode lists
  - Per-core free block lists
  - Per-core meta-log



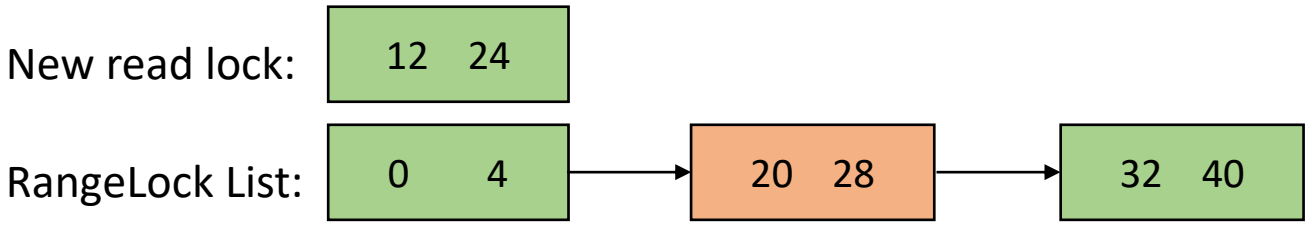
SPMFS Design

- Fine-Grained Range Lock
  - Read Lock



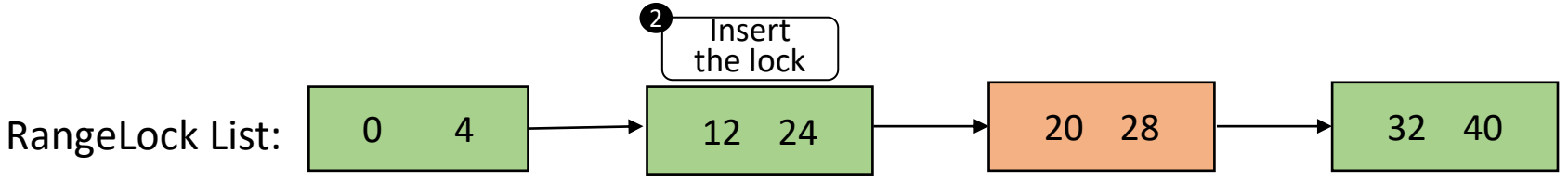
SPMFS Design

- Fine-Grained Range Lock
  - Read Lock



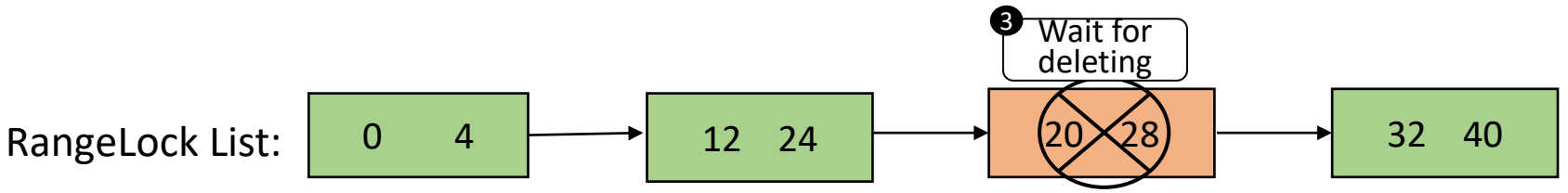
SPMFS Design

- Fine-Grained Range Lock
  - Read Lock



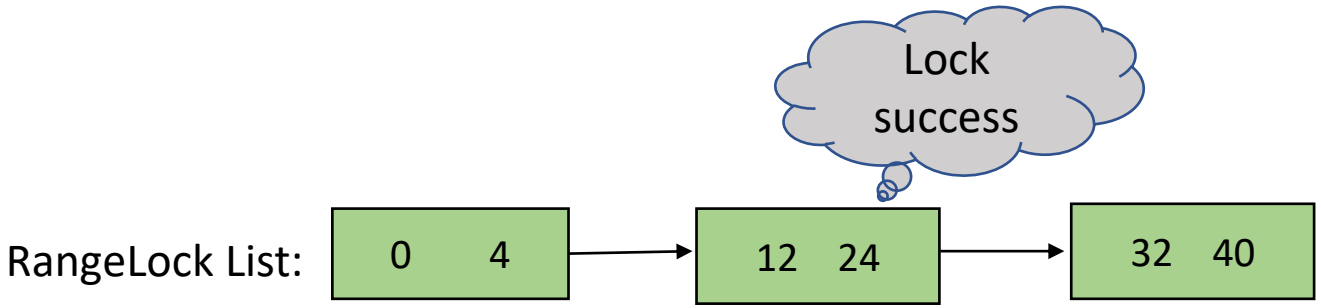
SPMFS Design

- Fine-Grained Range Lock
  - Read Lock



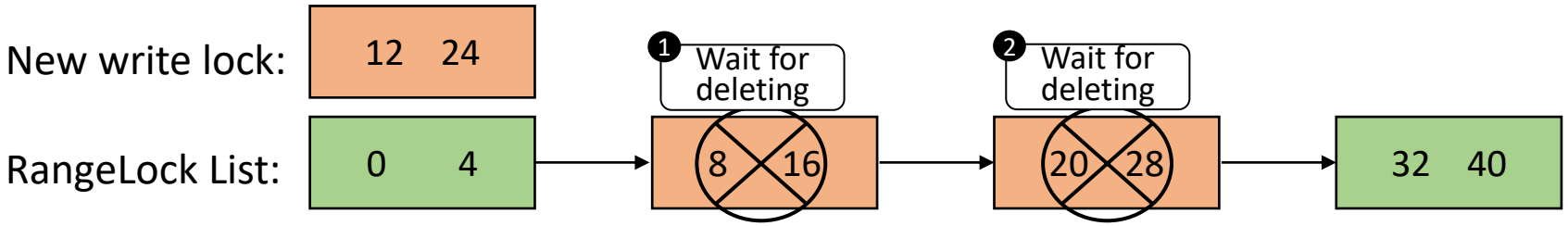
SPMFS Design

- Fine-Grained Range Lock
  - Read Lock



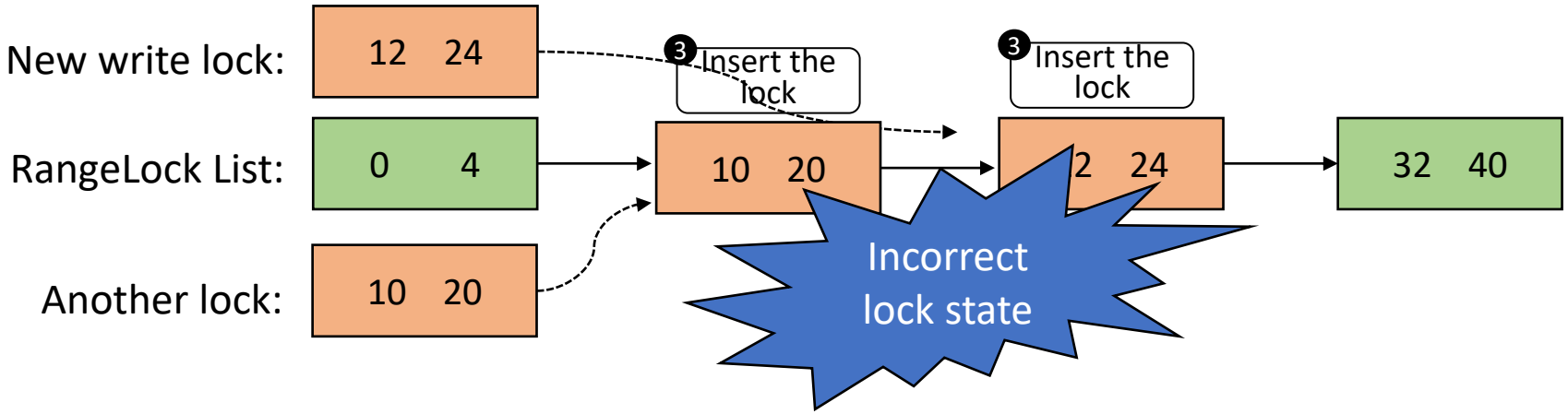
SPMFS Design

- Fine-Grained Range Lock
  - Write Lock



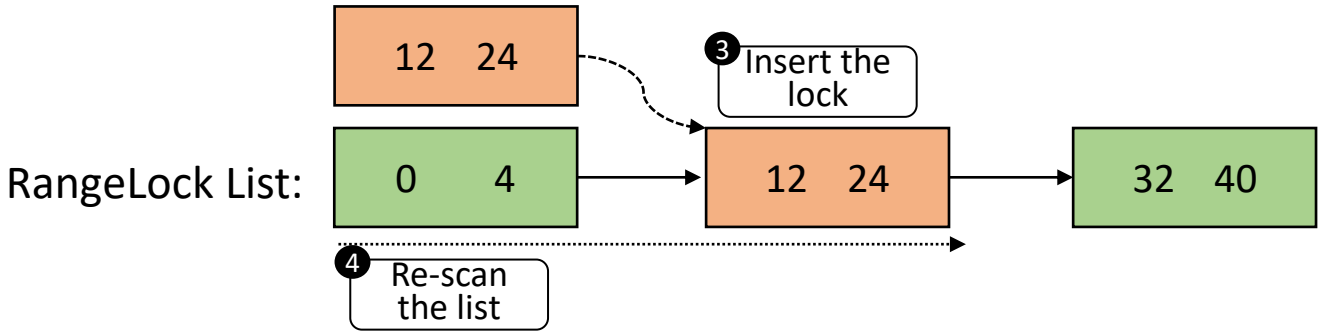
SPMFS Design

- Fine-Grained Range Lock
  - Write Lock



SPMFS Design

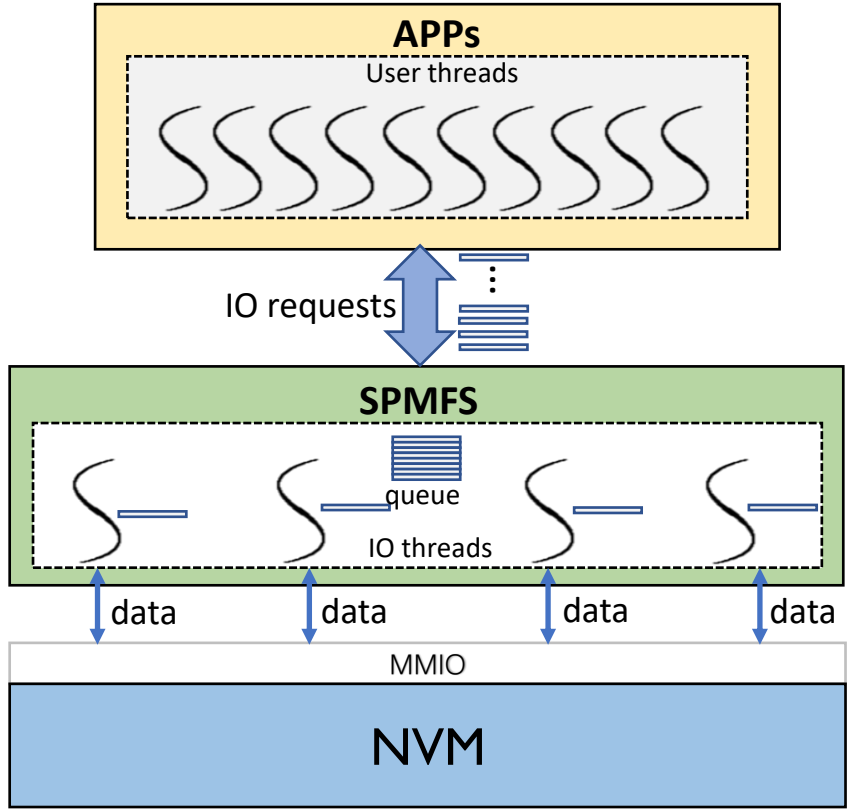
- Fine-Grained Range Lock
  - Write Lock



SPMFS Design

- Dedicated IO thread pool
  - Putting write requests to queue
  - Dedicated threads process the requests from queue
  - Write inconsistency
    - Data from user space to NVM directly
    - Data in user space may be updated again

Please see our paper for more details.



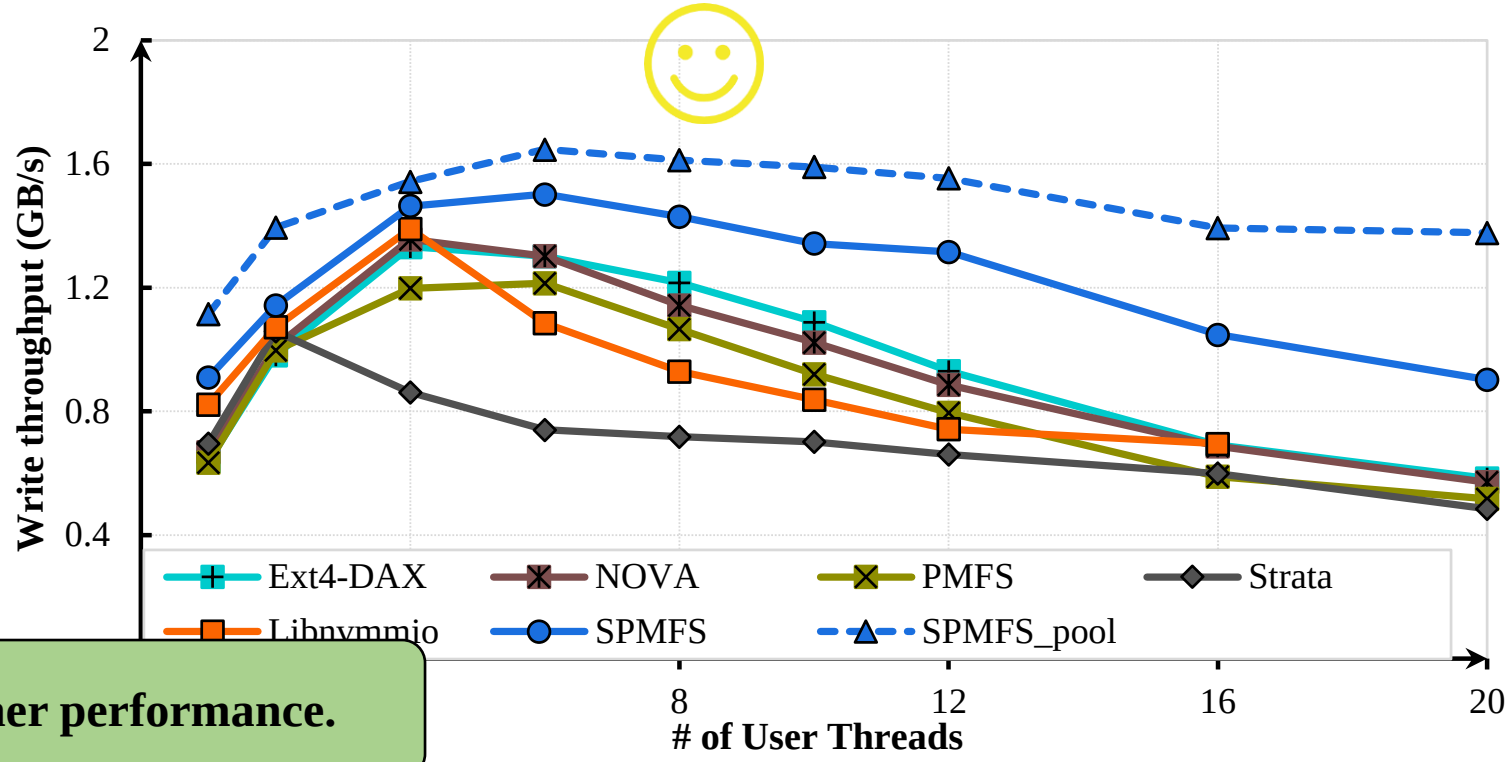
Experimental setup

- Hardware platform
  - Dual-socket 16-core Intel Xeon CPU @2.30GHz with 22MB of L3 cache.
  - 128GB Intel Optane DC PMM.

- State-of-the-art NVM file systems
  - Ext4-DAX, PMFS [Eurosys'14], NOVA [FAST'16] (Kernel-level filesystem)
  - Strata [SOSP'17], Libnvmmio [ATC'20] (User-space filesystem)

Performance Evaluation

- Write Performance
  - FIO-similar Workload
    - 32KB-sized IO
    - Sequential write
    - Varying numbers of threads
    - Each thread accesses its private file
  - Results



) vs. thread count on

SPMFS achieves higher performance.

SPMFS_pool further improves the write performance.

Performance Evaluation

- Range lock Performance
  - FIO workload
    - 32KB-sized IO
    - randomly writing data to a shared file
    - varying numbers of FIO threads
  - Results

SPMFS achieves higher write performance.

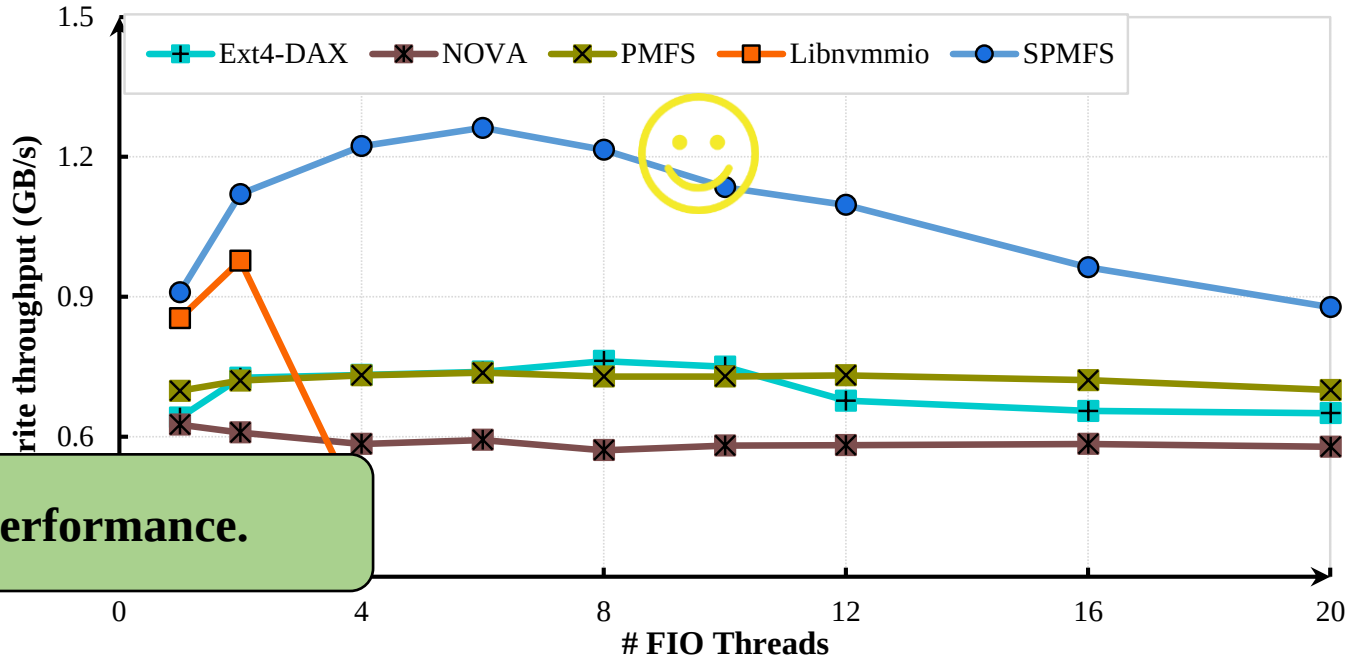


Fig. Write throughput (32KB IO) vs. FIO thread count on NVM-based file systems.

Conclusion

- PMM and existing file systems have limited scalability
- SPMFS : a scalable file system
  - Per-core resource manager
  - Fine-grained access range lock
  - Dedicated I/O thread pool
- SPMFS exhibits higher scalability against previous systems

Thank you!

Q&A

Yangyang_hust@hust.edu.cn