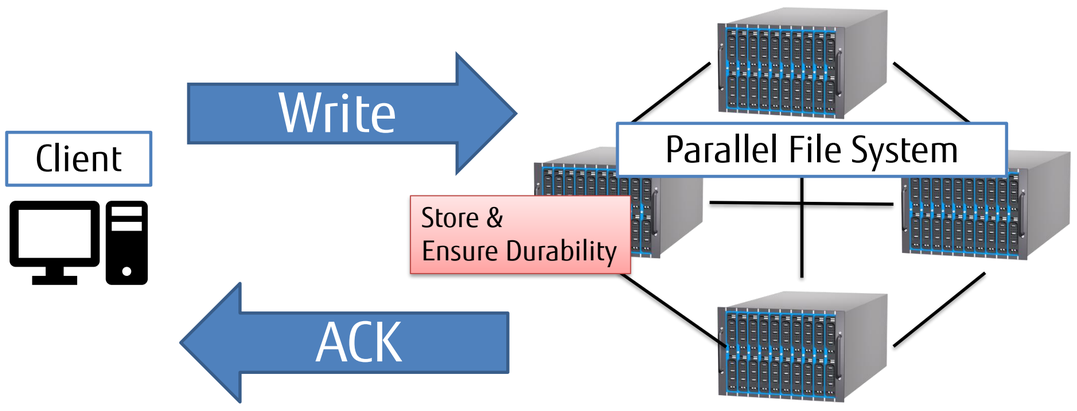
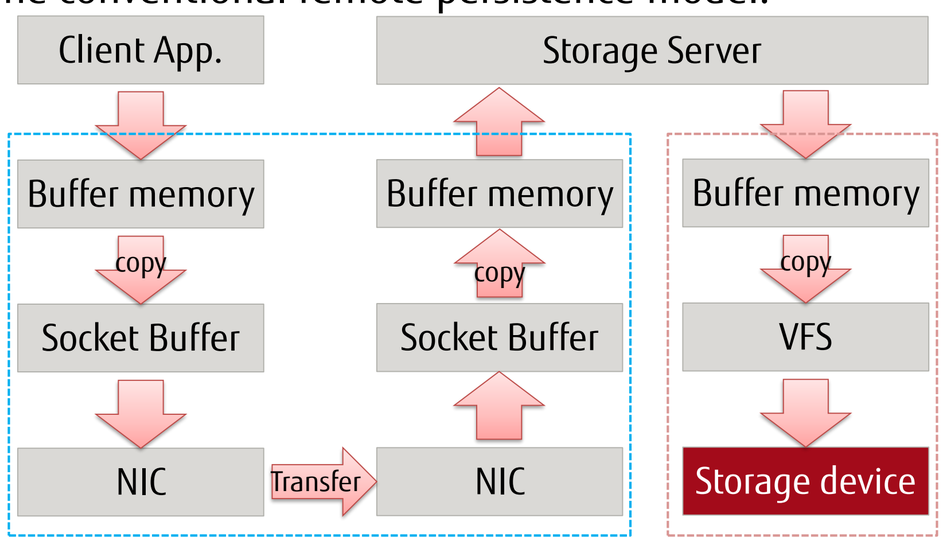
Low-overhead Remote Persistence for Scalable File Systems

Hiroki Ohtsuji[1], Takuya Okamoto[2], Erika Hayashi[1], Eiji Yoshida[1]
[1]Fujitsu Laboratories Ltd., [2]Fujitsu Ltd.

Background and Motivation



The conventional remote persistence model:

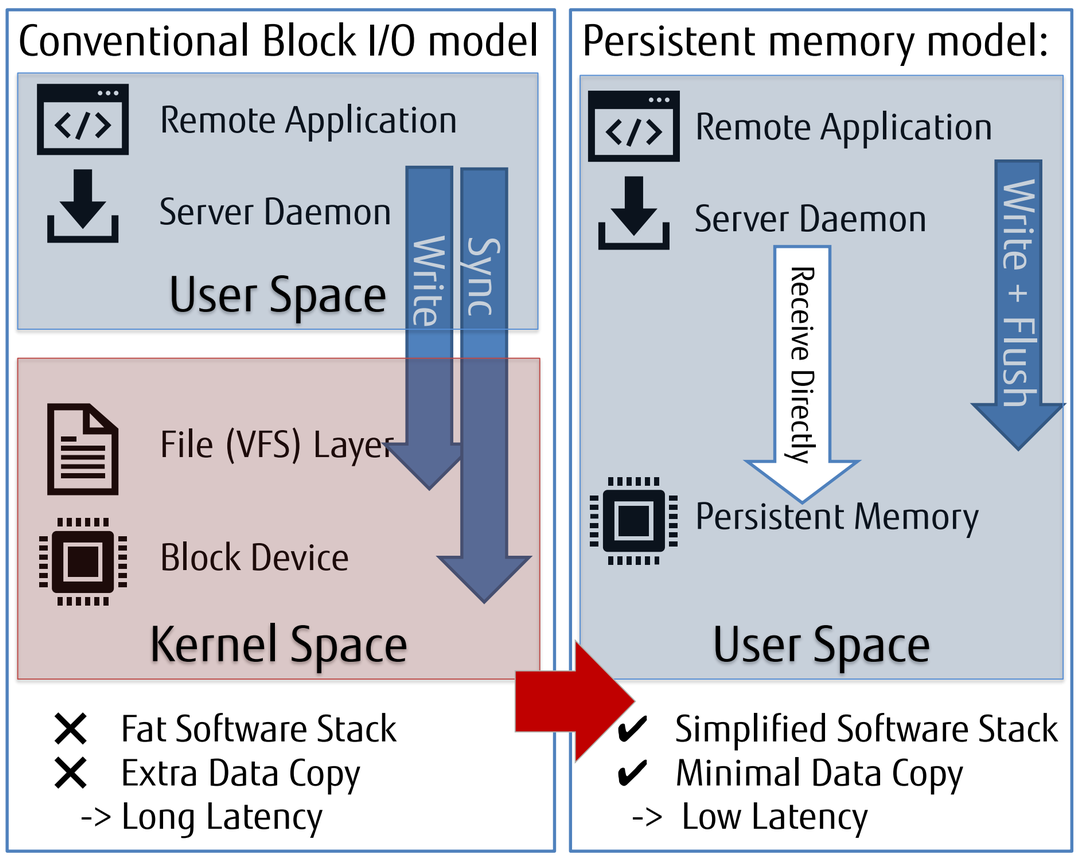


Remote persistence operations require multiple software layers and memory copy operations.

Reducing the overhead is required to leverage the performance advantages of persistent memory devices.

Low-overhead Remote Persistence

Eliminating Block I/O Operations with fine-grained persistency guarantee

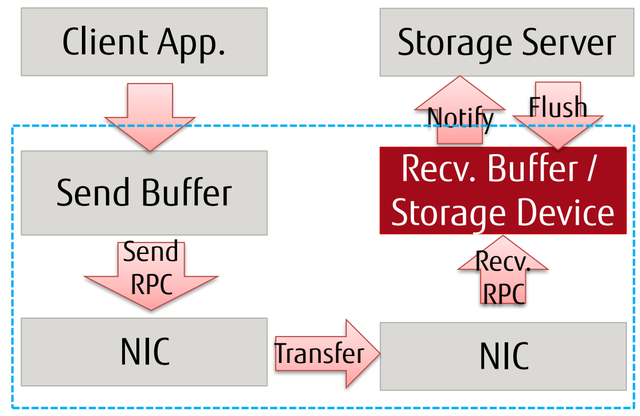


I/O Architecture Transformation is necessary

Reducing the overhead is required to leverages the performance advantages of persistent memory devices.

Optimized Remote Persistence Model

The optimized remote persistence model:



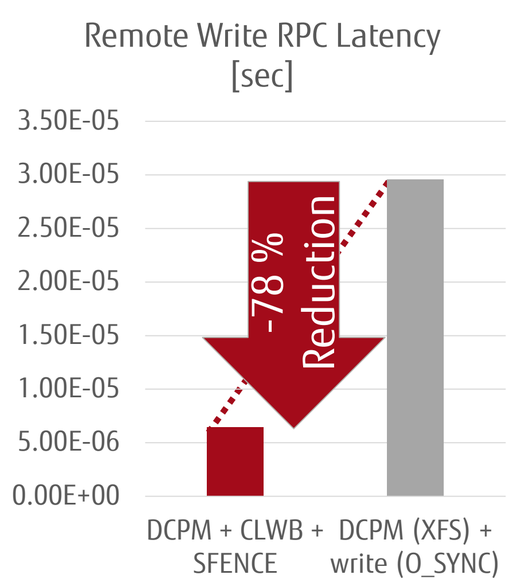
6.5 µs total latency

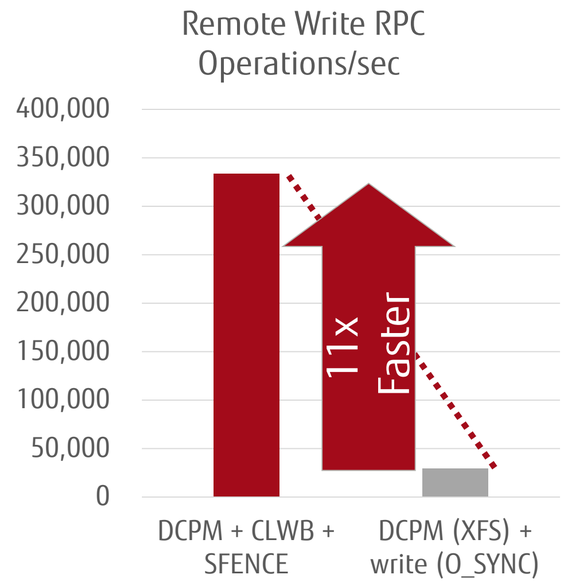
Implemented a light-weight RPC mechanism with InfiniBand Send/Recv Operations

Supporting direct persistence of RPC receiving buffer.

Evaluation

FUJITSU Server PRIMERGY RX2540 M5
Intel(R) Xeon(R) Gold 6240M CPU @ 2.60GHz x2
Intel DC Persistent Memory 128 GB x12
DDR4 DRAM 32 GB x12
Mellanox Connect-X5 EDR HCA Dual Ports





DCPM (Block) + write (O_SYNC)
-> Using DCPM as a block device (XFS)
-> Writing data blocks with synchronization
DCPM + CLWB + SFENCE
-> Binding DCPM to a specific NUMA node
-> Writing data with cache flush and fence operations
Evaluated these two methods with 2 KB write RPC operations.

DCPM + CLWB + SFENCE case reduced **78 % of remote write latency** compared to the case of block I/O on XFS.
Both cases use the same storage devices.

Conclusion and Future Work

Conclusion
-> Optimized write/flush mechanism for NVMM can reduce the remote persistence overhead.
Future Work
-> Implementing a complete set of File I/O operations on top of the proposed remote persistent mechanism.