

Introduction to the Special Issue on NVM and Storage

Non-volatile memory (NVM) technologies have demonstrated great potentials on improving many aspects of present and future memory hierarchy, offering high integration density, larger capacity, zero standby power, and good resilience to soft errors. The recent research progress of various NVMs, e.g., NAND flash, PCM (Phase-change memory), STT-RAM (Spin-Transfer Torque Random Access Memory), RRAM (Resistive random-access memory), FeRAM (Ferroelectric RAM), and so on, have drawn tremendous attention from both the academy and industry. The unique characteristics of these NVM technologies, such as read-write asymmetry, stochastic programming behavior, access through memory rather than I/O instructions, the performance/power/non-volatility tradeoff, introduce many opportunities and challenges for novel system and storage design optimizations.

This special issue addresses the recent developments on non-volatile memory in the context of storage systems. After a rigorous review process, a set of articles was selected for this special issue. These articles are briefly discussed in the rest of the editorial.

In “UnistorFS: A Union Storage File System Design for Resource Sharing between Memory and Storage on Persistent RAM-Based Systems,” Chen et al. propose a union storage file system that jointly manages the NVM resource for both main memory and storage to eliminate unnecessary memory accesses and improve performance.

In “HiNFS: A Persistent Memory File System with Both Buffering and Direct-Access,” Chen et al. propose a high-performance file system for non-volatile main memory to combine both buffering and direct access for fine-grained file system operations, which improves throughput as well as execution time.

In “clfB-tree: Cacheline Friendly Persistent B-Tree for NVRAM,” Kim et al. propose a B-tree structure on NVM storage whose tree node fits into a single cache line. The proposed outperforms existing B-tree in terms of insertion and search performance.

In “Persisting RB-Tree into NVM in a Consistency Perspective,” Wang et al. propose a NVM friendly RB-tree that is all-time consistent and scalable, without the need of recovery procedure after system crashes.

In “DudeTx: Durable Transactions Made Decoupled,” Liu et al. describe optimizations for transactions on NVM to improve throughput.

When RRAM is used for storage of database items, Sun et al. propose to take advantage of the data that can be read both rowwise and columnwise. “Bidirectional Database Storage and SQL Query Exploiting RRAM-Based Process-in-Memory Structure” shows an exploration that can improve four to six orders of magnitudes compared with the traditional architecture.

RRAM can also be used for both computation and storage. In “A Novel ReRAM-Based Processing-in-Memory Architecture for Graph Traversal,” Han et al. achieve significant graph traversal speed-up with RRAM to persistently store and process graph data.

Flash memories are now used everywhere. In “An Analysis of Flash Page Reuse with WOM Codes,” Yadgar et al. applied WOM (Write Once Memory) codes in Flash to reduce writes on Flash and improve performance and endurance. In “SLC-Like Programming Scheme for MLC Flash Memory,” Ho et al. propose a new programming scheme for Flash memory to improve both programming speed as well as reduce error rate for Flash memories.

Finally, the guest editors thank all the reviewers for the hard work and timely reviews. The guest editors also thank the Editor-in-Chief, who made this special issue possible by providing the vision to advocate NVM for storage and systems.

Chun Jason Xue
Michael Swift
Guest Editors