Persistent Memory: Accessed as Memory, Managed as Storage

Nisha Talagala
Fellow, SanDisk

August 7, 2014

SanDisk®

Forward-Looking Statements

During our meeting today we may make forward-looking statements.

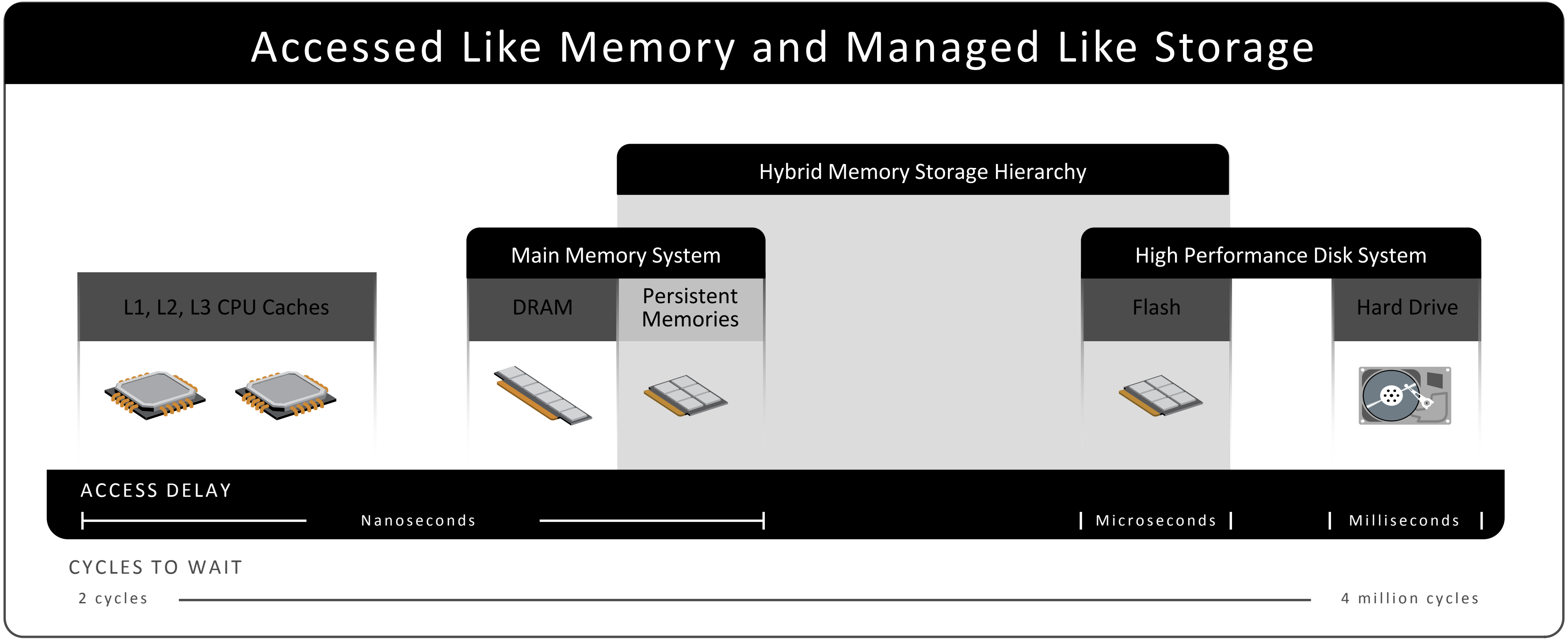
Any statement that refers to expectations, projections or other characterizations of future events or circumstances is a forward-looking statement, including those relating to industry trends, future memory technology and product capabilities and performance. Information in this presentation may also include or be based upon information from third parties, which reflects their expectations and projections as of the date of issuance.

Actual results may differ materially from those expressed in these forward-looking statements due to factors detailed under the caption “Risk Factors” and elsewhere in the documents we file from time to time with the SEC, including our annual and quarterly reports.

We undertake no obligation to update these forward-looking statements, which speak only as of the date hereof or as of the date of issuance by a third party, as the case may be.

SanDisk®

Hybrid Memory-Storage Hierarchy



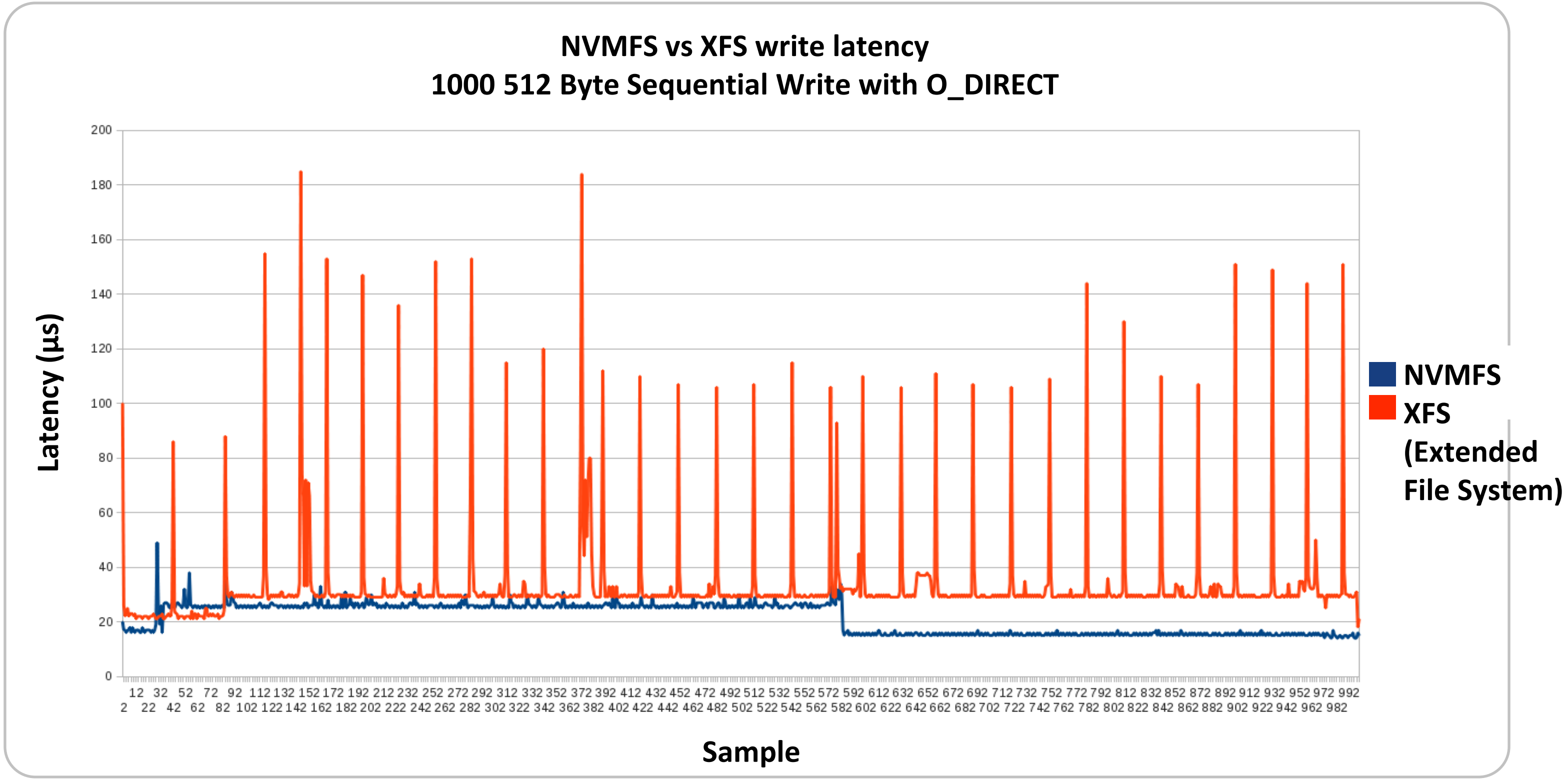
ACCESSED WITH THE
SPEED AND SIMPLICITY OF MEMORY

MANAGED WITH THE
FAMILIARITY AND CAPACITY OF STORAGE

NVMFS (Non-Volatile Memory File System)

- **Appears as Linux file system**
  - Provides performance to applications “as is”
  - Focuses only on file namespace
  - POSIX (Portable Operating System Interface) compatible
- **Flash**
  - **Employs existing flash translation layer** for:
  - Large virtualized addressed space
  - Direct flash access
  - Crash recovery mechanisms
  - **Exports primitives through file namespace**
- **Persistent Memory:** Application access through filesystem or straight to device

NVMFS: Consistent Low Latency

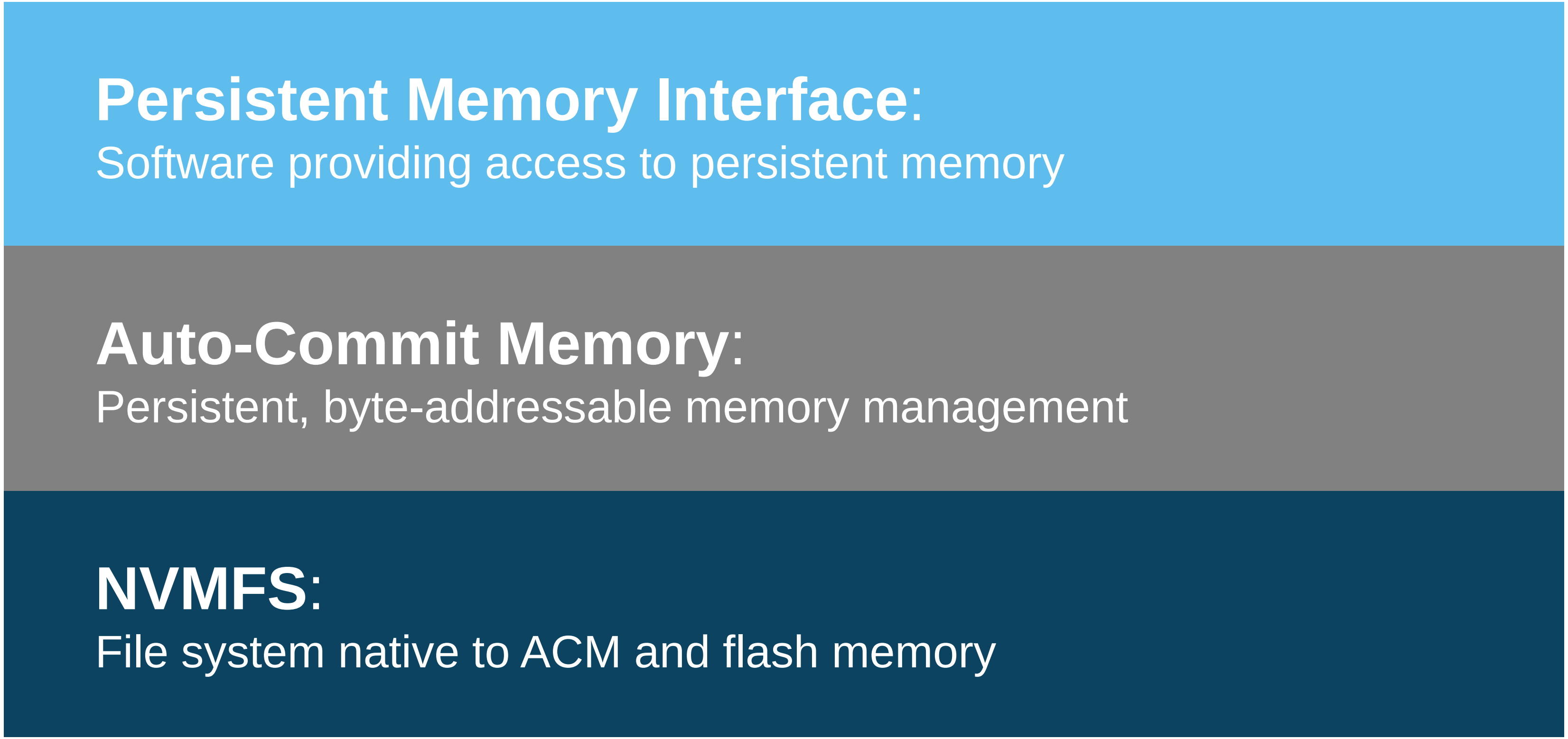


ACM (Auto Commit Memory): A Persistent Memory

- Persistent, granular, byte and cacheline size updates
  - Latency reductions 10x-20x*
  - 32x-64x less data written to media
- Direct memory access reduces CPU overhead
- Integrated with NVMFS for use with regular files
- Automatic tiering across persistent memory and flash
- Application integrated Storage Class Memory

*compared to write operations to an ioMemory device though classic block level interfaces

An example use case:

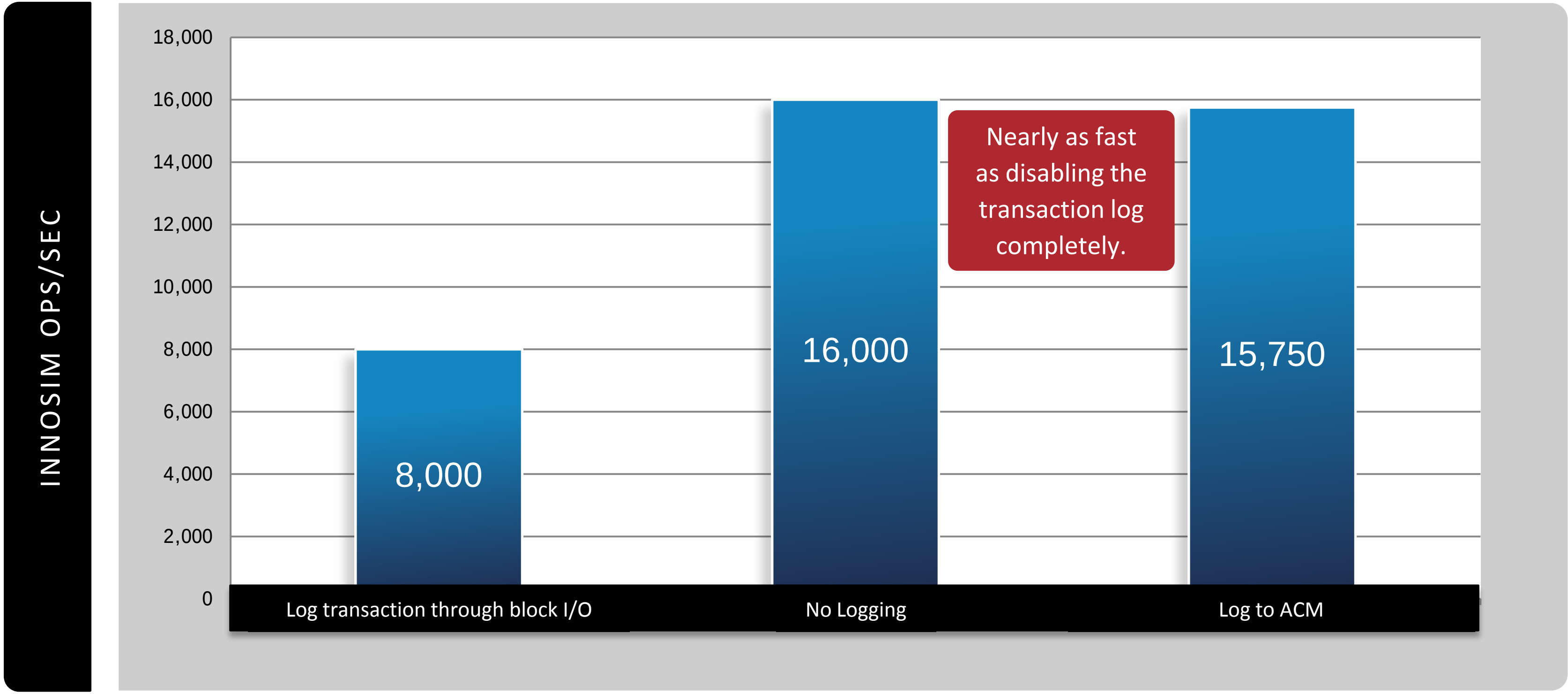


Persistent Memory Interface:
Software providing access to persistent memory

Auto-Commit Memory:
Persistent, byte-addressable memory management

NVMFS:
File system native to ACM and flash memory

Example Use Case: MySQL Database



Nearly as fast as disabling the transaction log completely.

Appendix