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.

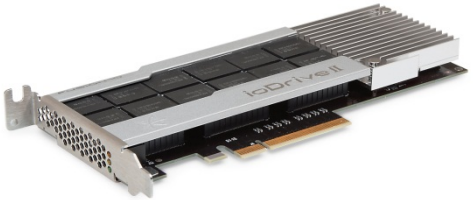
Actual results may differ materially from those expressed in these forward-looking statements due to factors detailed under the caption “Risk Factors” and elsewhere in the documents we file from time to time with the SEC, including our annual and quarterly reports.

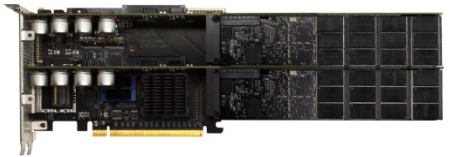
We undertake no obligation to update these forward-looking statements, which speak only as of the date hereof or as of the date of issuance by a third party, as the case may be.

SanDisk®

Non Volatile Memory (NVM)

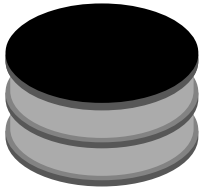
- Flash (NAND)
  - 100s GB to 10 TB per device
  - Trends - Higher capacity, lower cost/GB, lower write cycles, SLC->MLC->3BPC
  - 100K to millions of IOPS, GB/s of bandwidth
- PCM/MRAM/STT/Other NVMs
  - Still in research

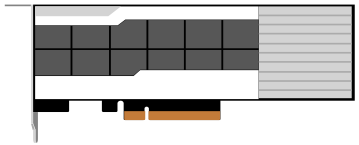




Why Flash?

- Perfect match for Databases
- Low latency – good for low queue I/O
- Handles mixed sequential and random I/O at various block sizes much better than disk

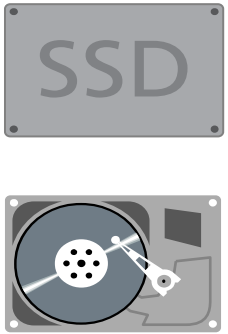




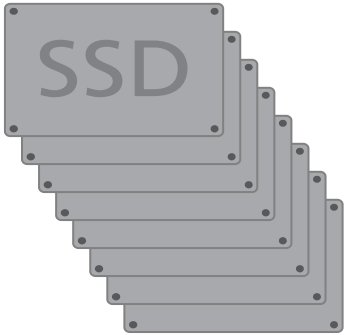
Capacity	4TB	3TB
IOPS	15	200,000
Cost/IOP	$$$$	¢¢¢¢

Evolution of Flash Usage

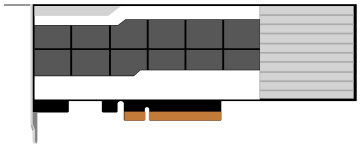
FLASH + DISK



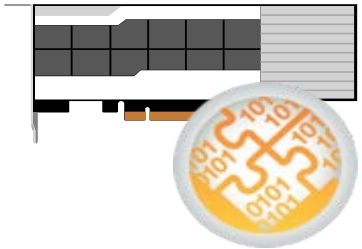
FLASH AS DISK



FLASH BEYOND DISK



FLASH AS MEMORY



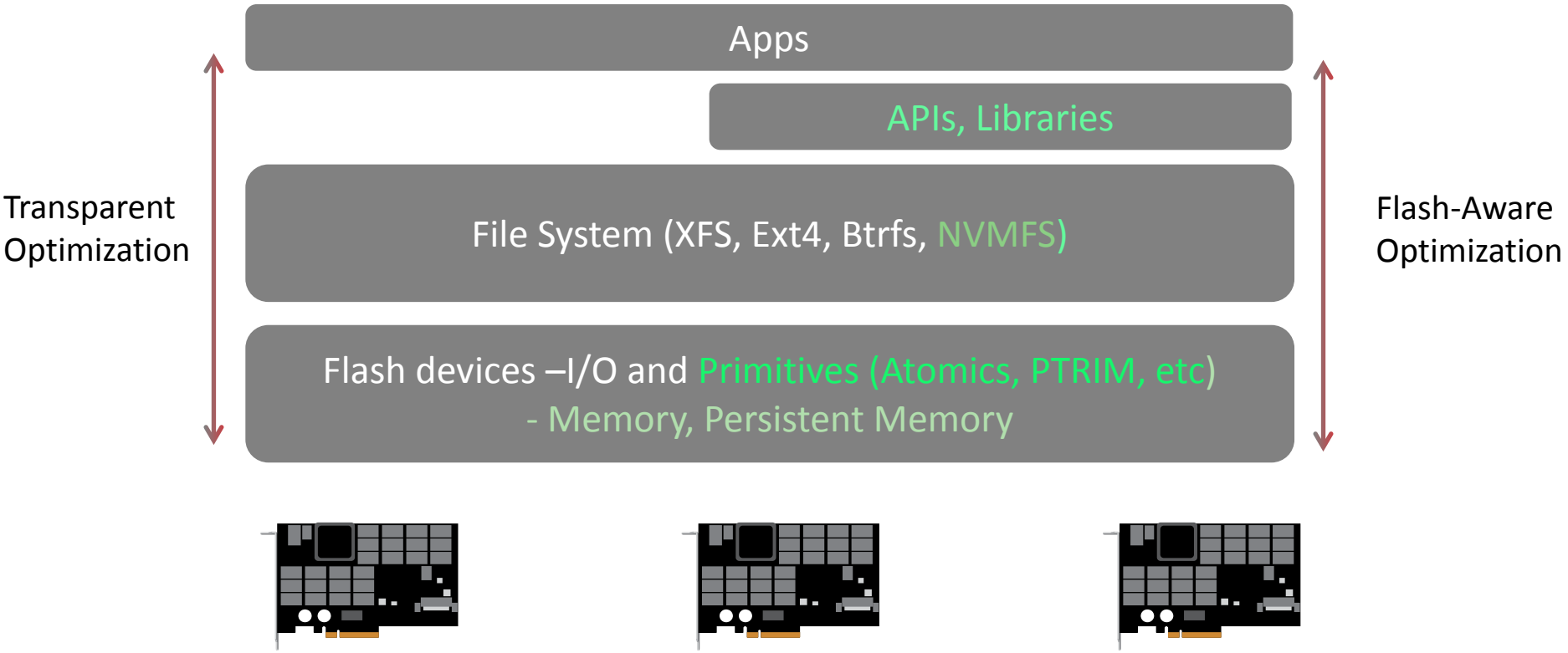
Better Transactions/$/Watt

Increased flash awareness

Flash Beyond Disk: Not Just Fast Disk

Area	Hard Disk Drives	Flash Devices
Read/Write Performance	Largely symmetrical	Heavily asymmetrical. Additional operation (erase)
Sequential vs Random Performance	100x difference. Elevator scheduling for disk arm	<10x difference. No disk arm – NAND die
Mapping and Background ops	Rare	Regular occurrence – like a log structured file system
Wear out	Largely unlimited	Limited writes
IOPS	100s to 1,000s	100Ks to Millions
Latency	10s ms	10s-100s us

Flash-Awareness: Opportunities in the I/O Stack



Examples covered in this talk

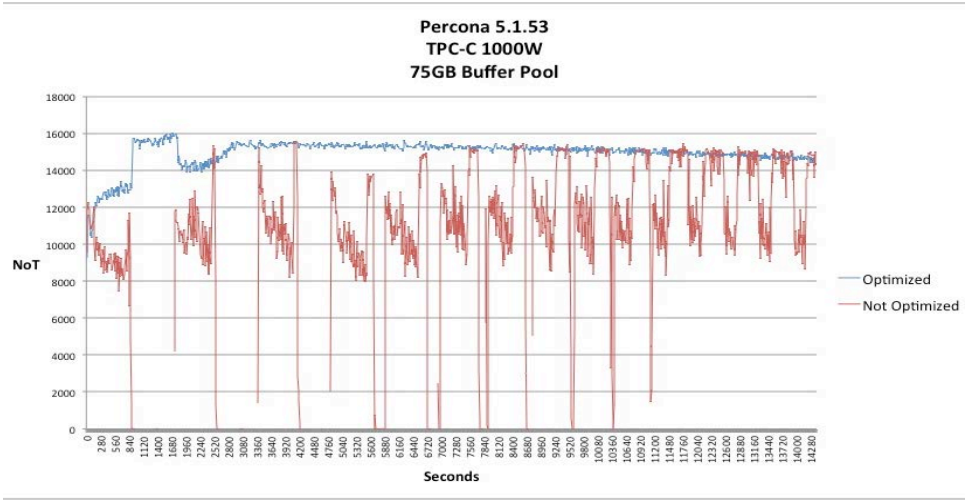
- MySQL
- MongoDB
- LevelDB

Gigaspaces with ZetaScale™ Software – See OPEN Session 301-A: Flash Changes the Game in Application Performance and TCO (Enterprise Applications Track)

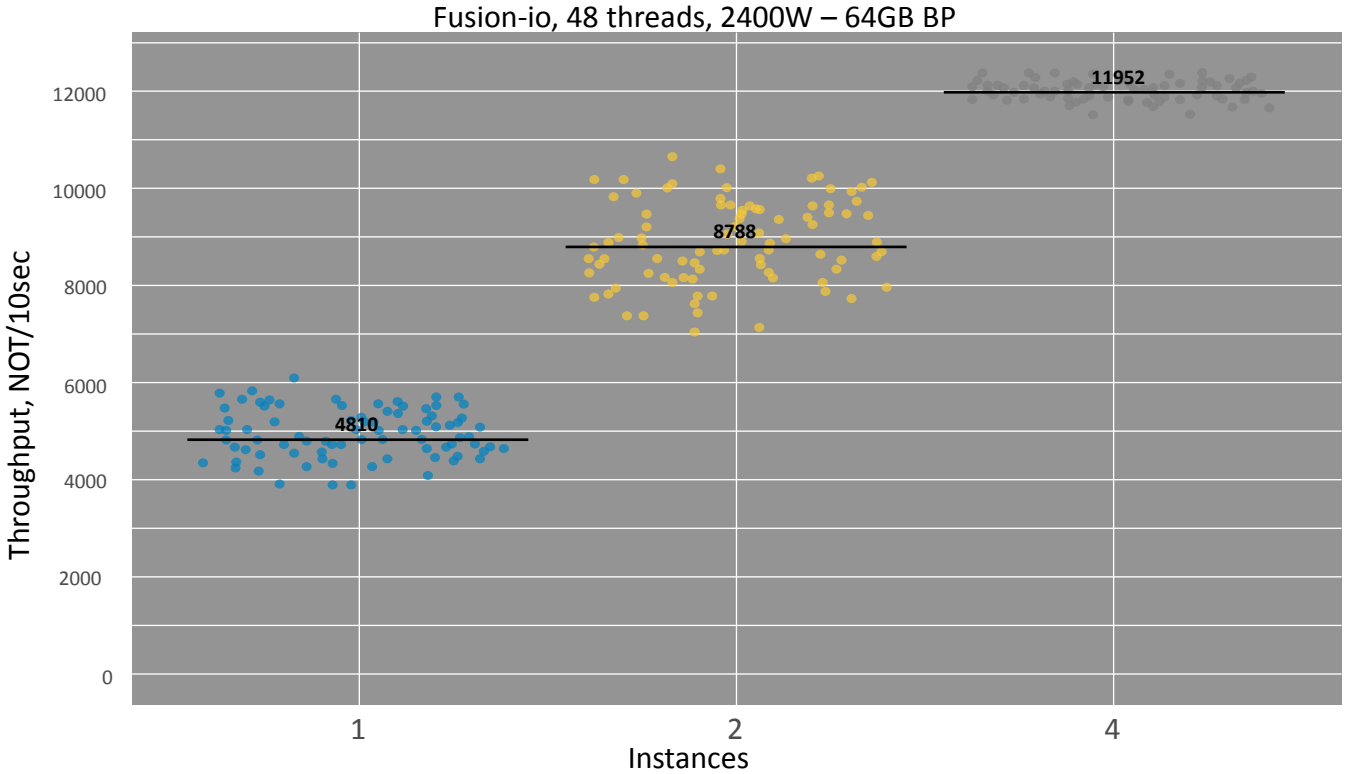
Example: MySQL

Flash as Disk: Tune for a Really Fast Disk

- Has been going on now for several years with great results
- Targeted data placement, NOOP scheduler, seek-less optimizations, concurrency improvements, etc.
- Make the block I/O stack faster
- Find the fastest file-system

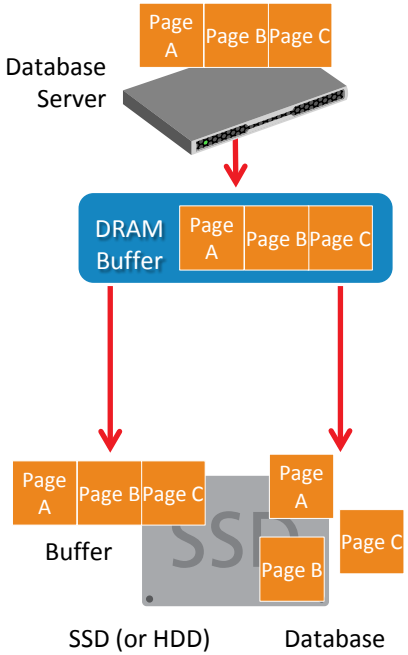


Multi-Instance: Using all the IOPS



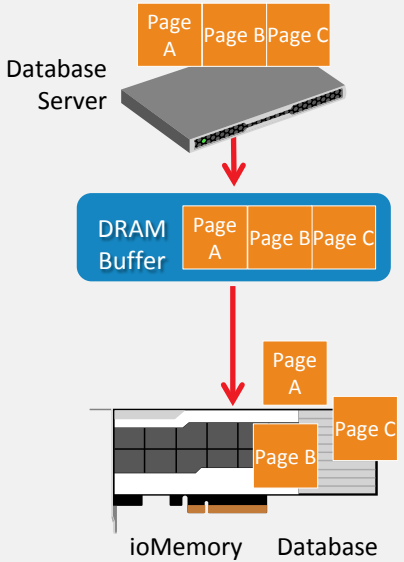
Atomic Writes

Traditional MySQL Writes



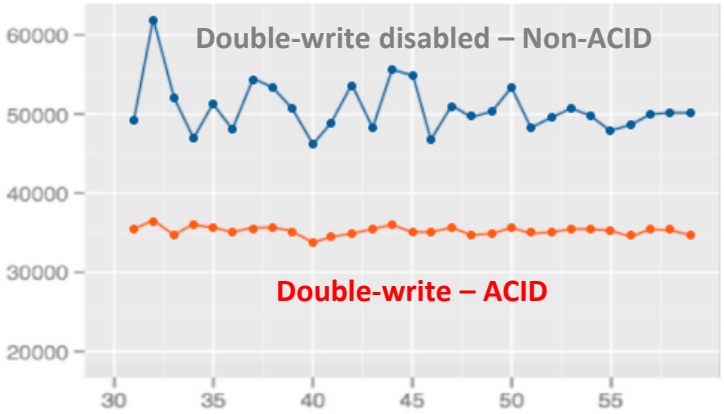
1. Application initiates updates to pages A, B, and C.
2. MySQL copies updated pages to memory buffer.
3. MySQL writes to double-write buffer on the media.
4. Once step 3 is acknowledged, MySQL writes the updates to the actual tablespace.

MySQL with Atomic Writes

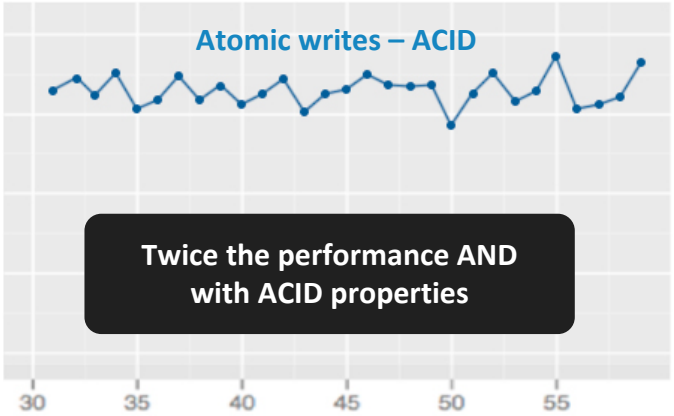


1. Application initiates updates to pages A, B, and C.
2. MySQL copies updated pages to memory buffer.
3. MySQL writes to actual tablespace, bypassing the double-write buffer step due to inherent atomicity guaranteed by the (intelligent) device.

Performance with Atomic Writes



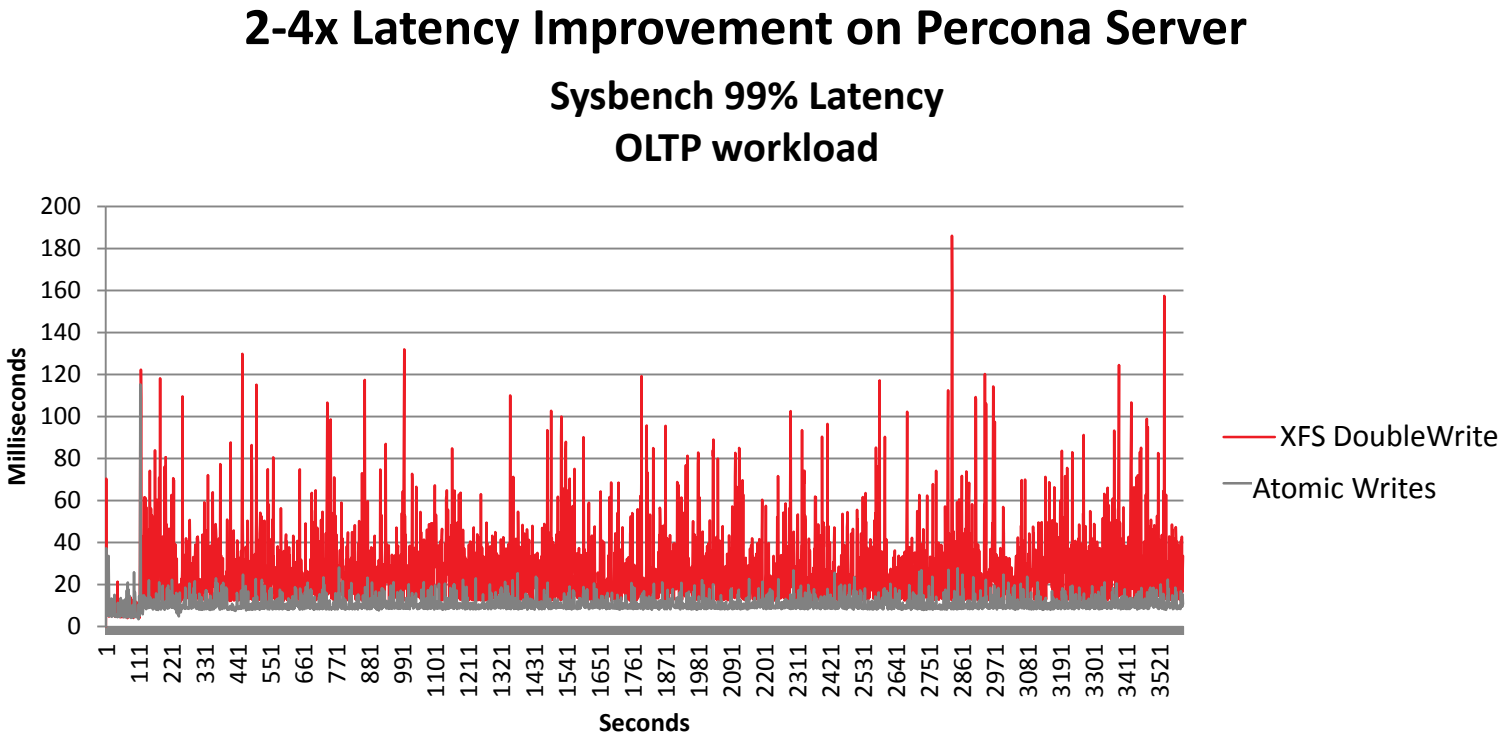
Without Atomic Writes



With Atomic I/O

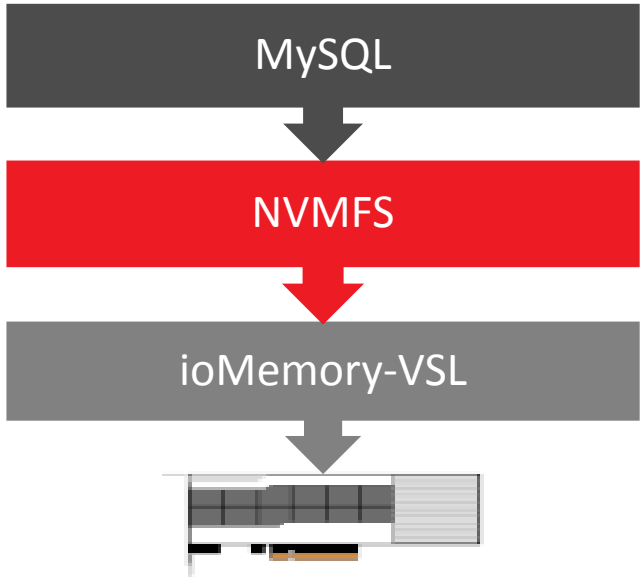
- Atomic writes at 99% of the performance of raw writes
- 2x flash device endurance improvement

Atomic Writes: Latency Improvement

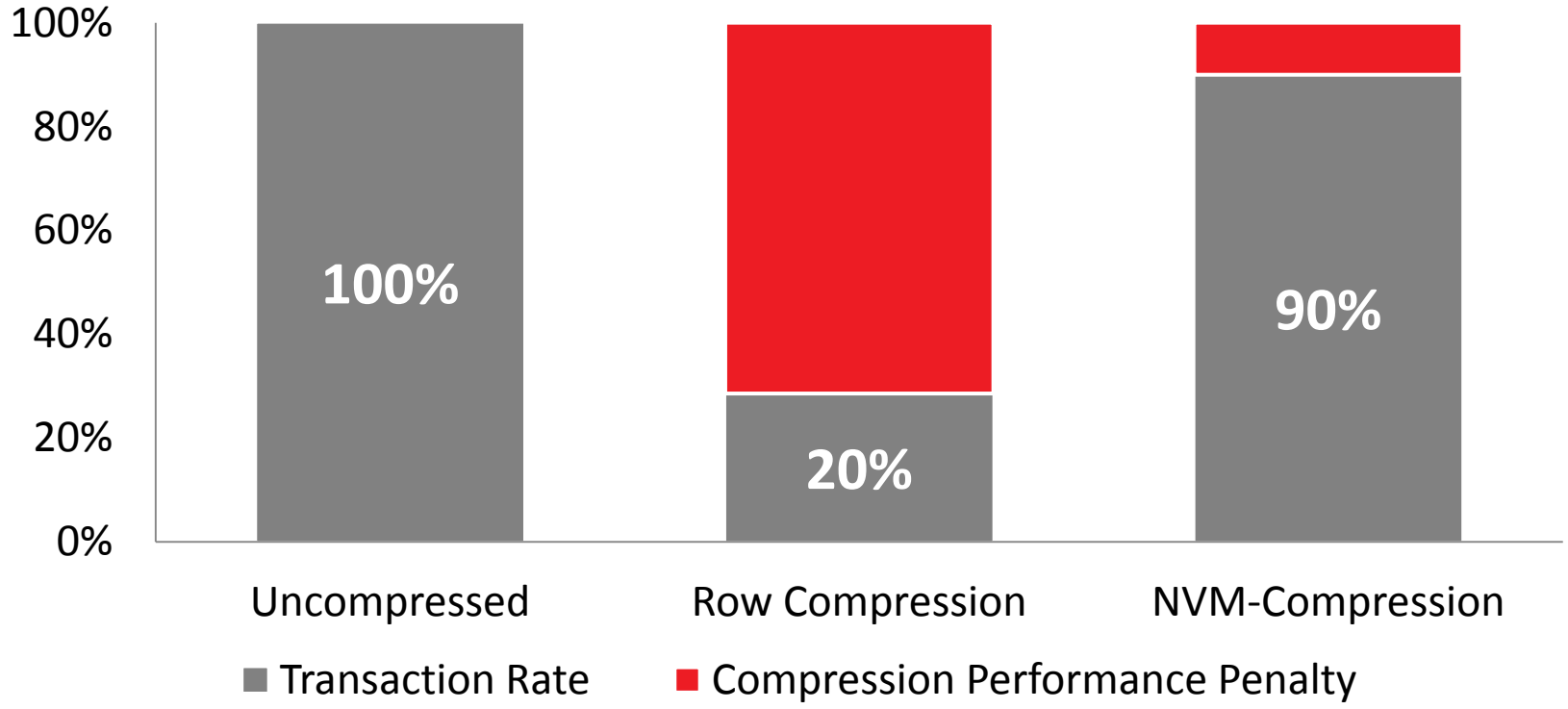


NVM-Compression

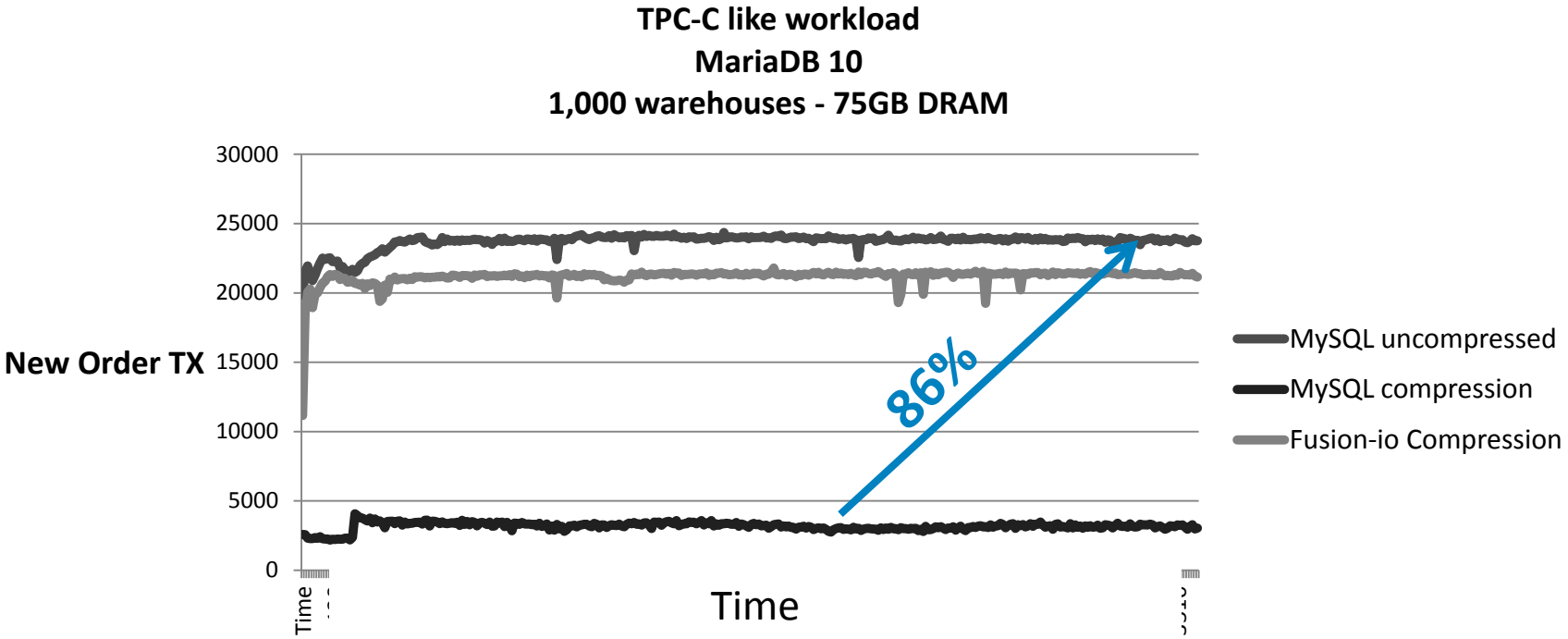
- Uses natural thin-provisioning of the underlying flash
- No bit packing at MySQL, no rebalancing
- Holes in data files are TRIMed/unmapped
- Multi-threaded flush and atomics to reduce latency
- Pluggable compression algorithms



Performance Improvement (LinkBench)

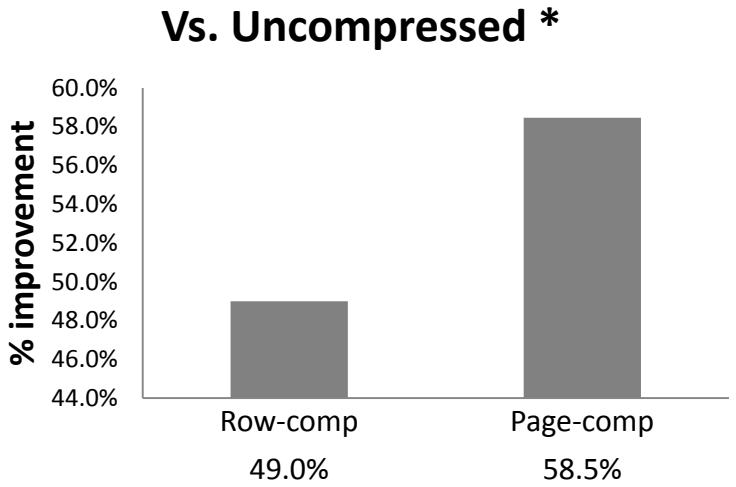


Performance Improvement: (TPCC-like)



Capacity Efficiency and Endurance

- Architecturally more storage efficient than row compression
- When combined with Atomic Writes, can improve flash endurance ~4x

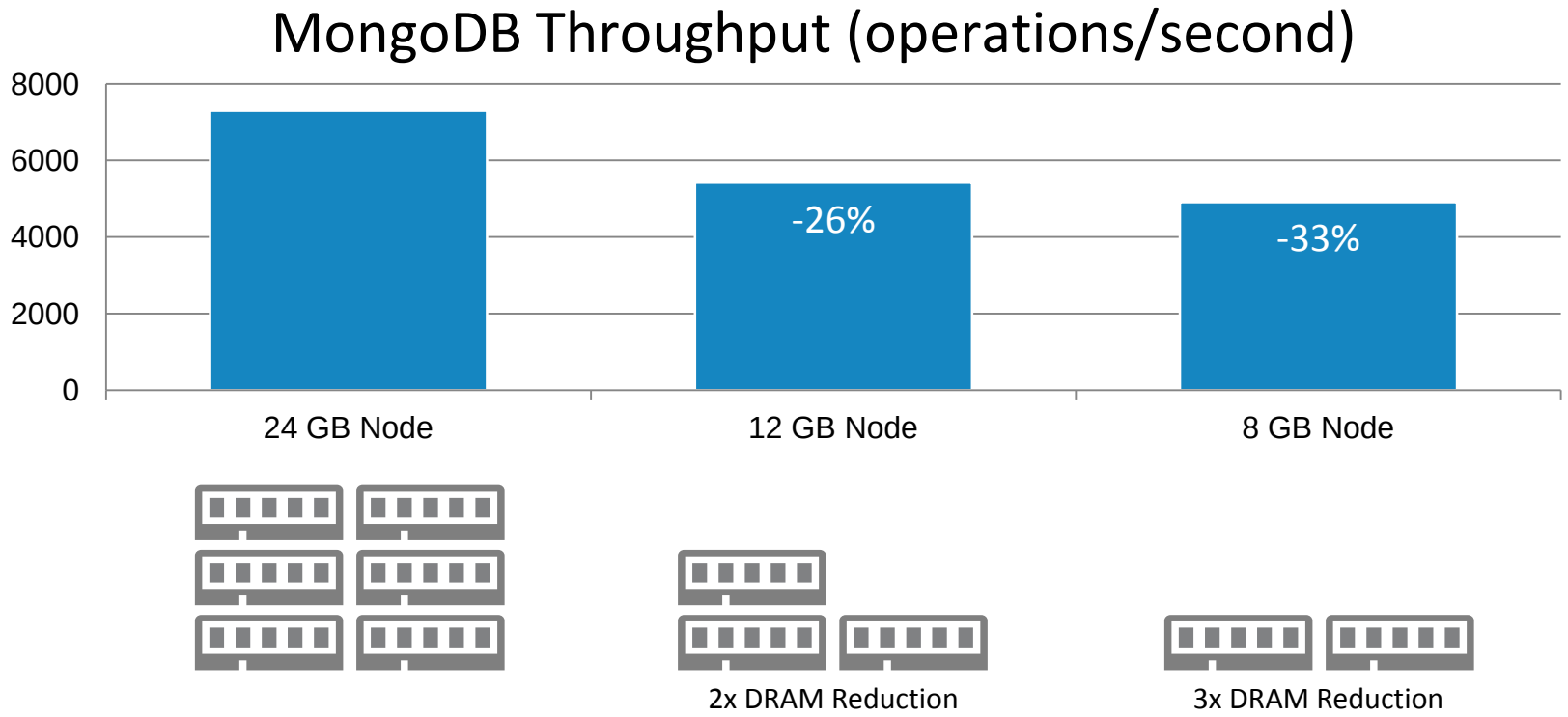


***For LinkBench with lz77. Comparable results with lz4.**

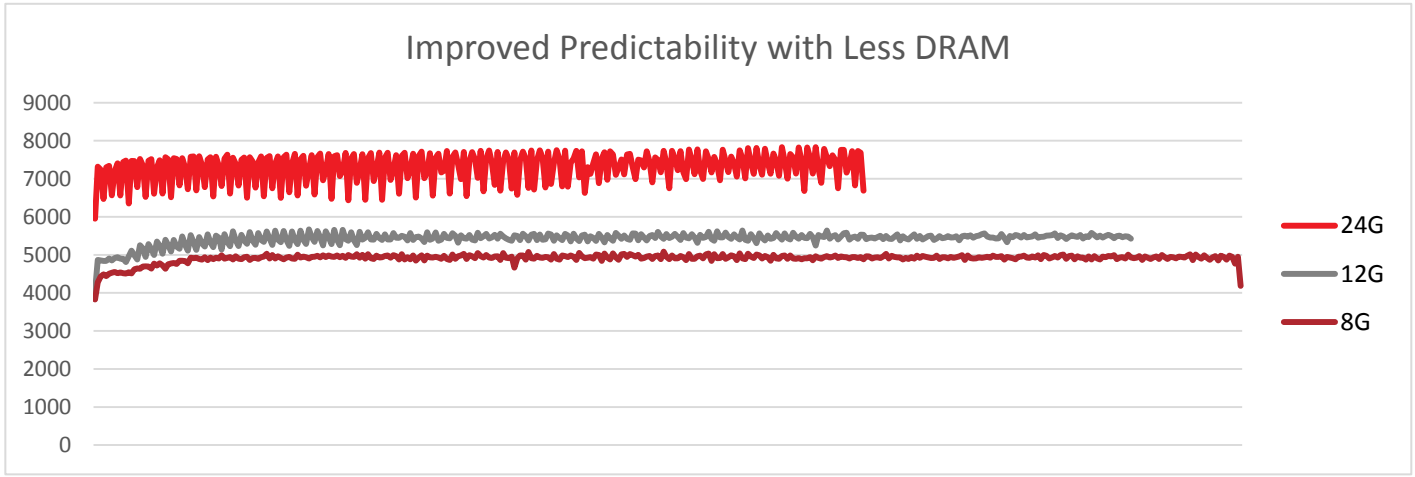
Flash Aware API Integration for Database

Example: MongoDB

Reducing DRAM with Flash



Improving Performance Predictability while reducing DRAM

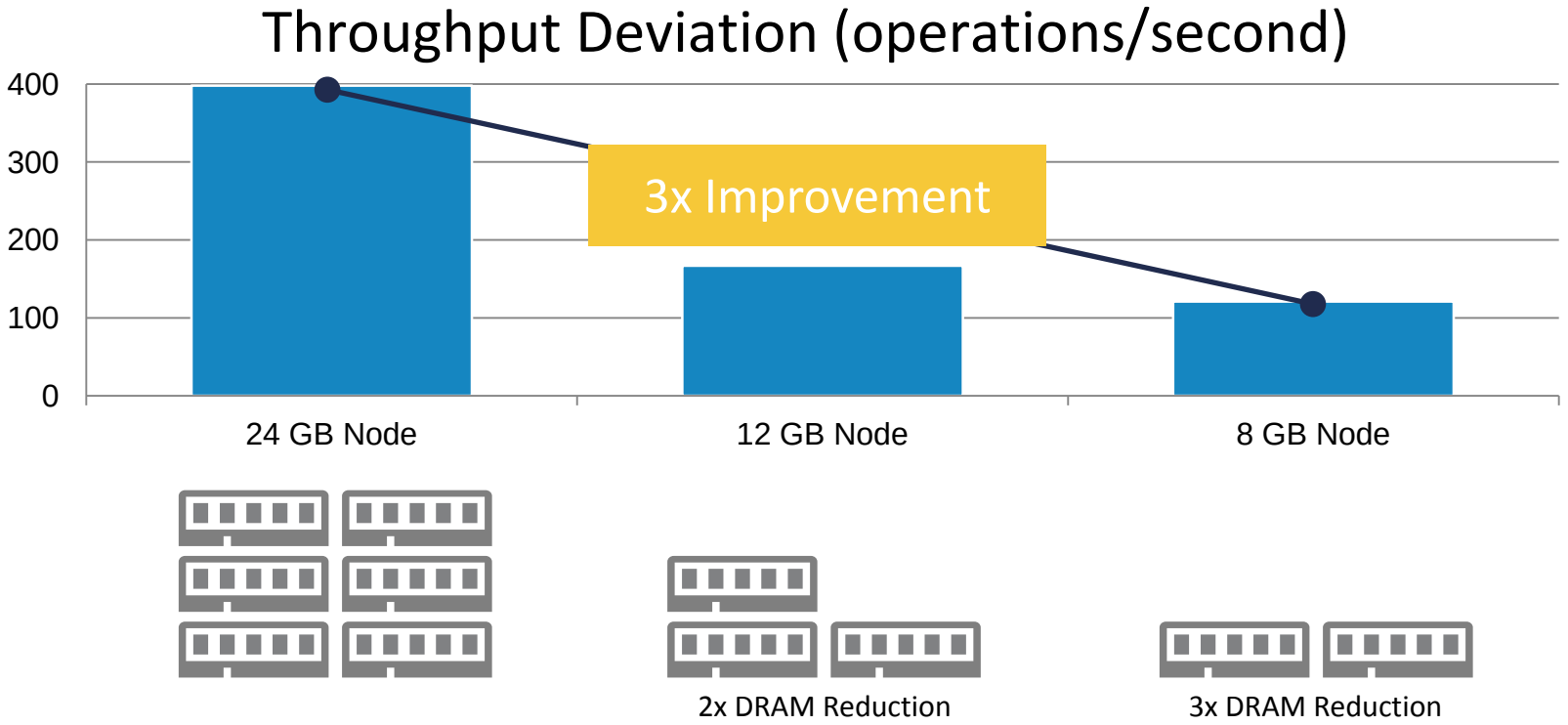


24GB Baseline

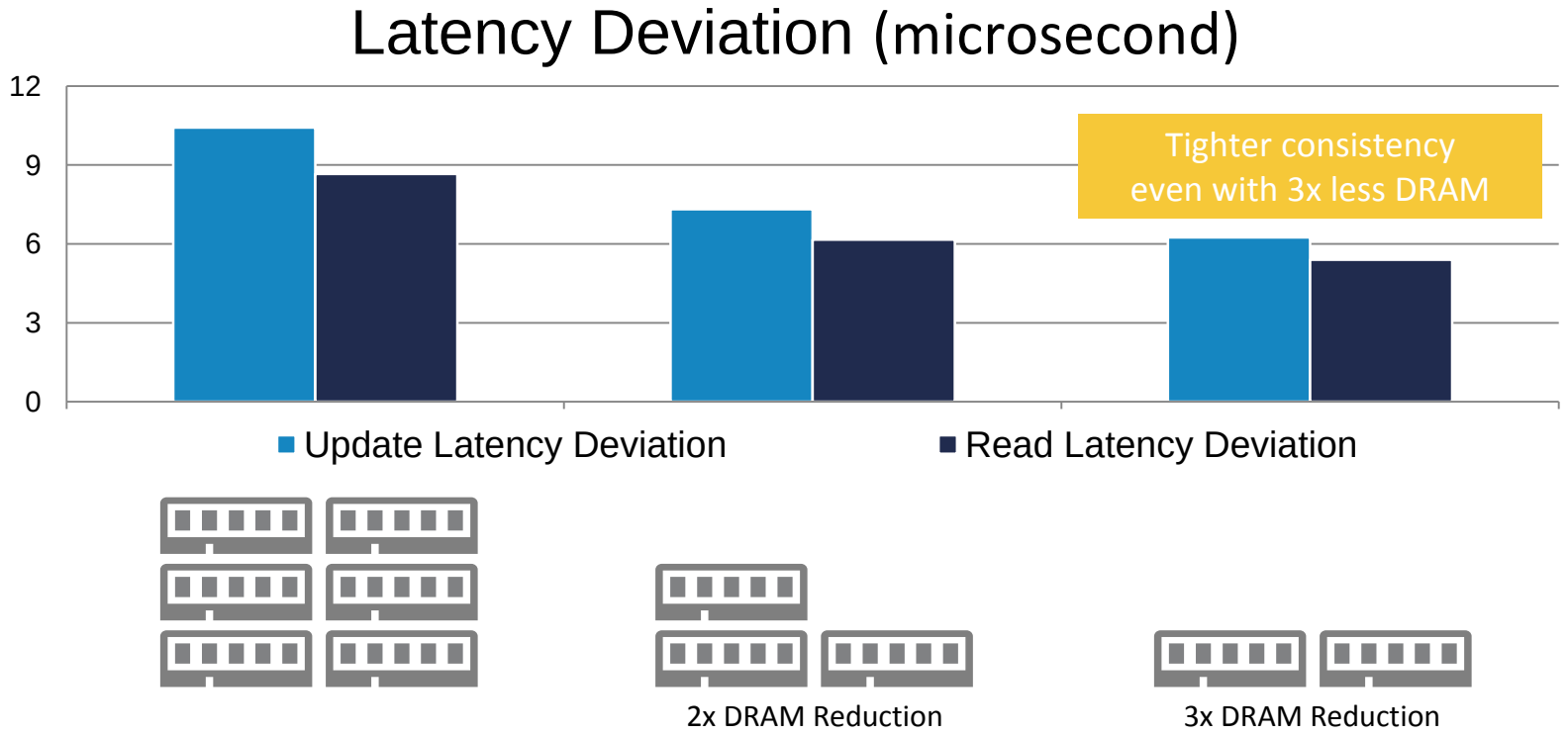
12GB 2x DRAM Reduction

8GB 3x DRAM Reduction

Improving Performance Predictability while reducing DRAM

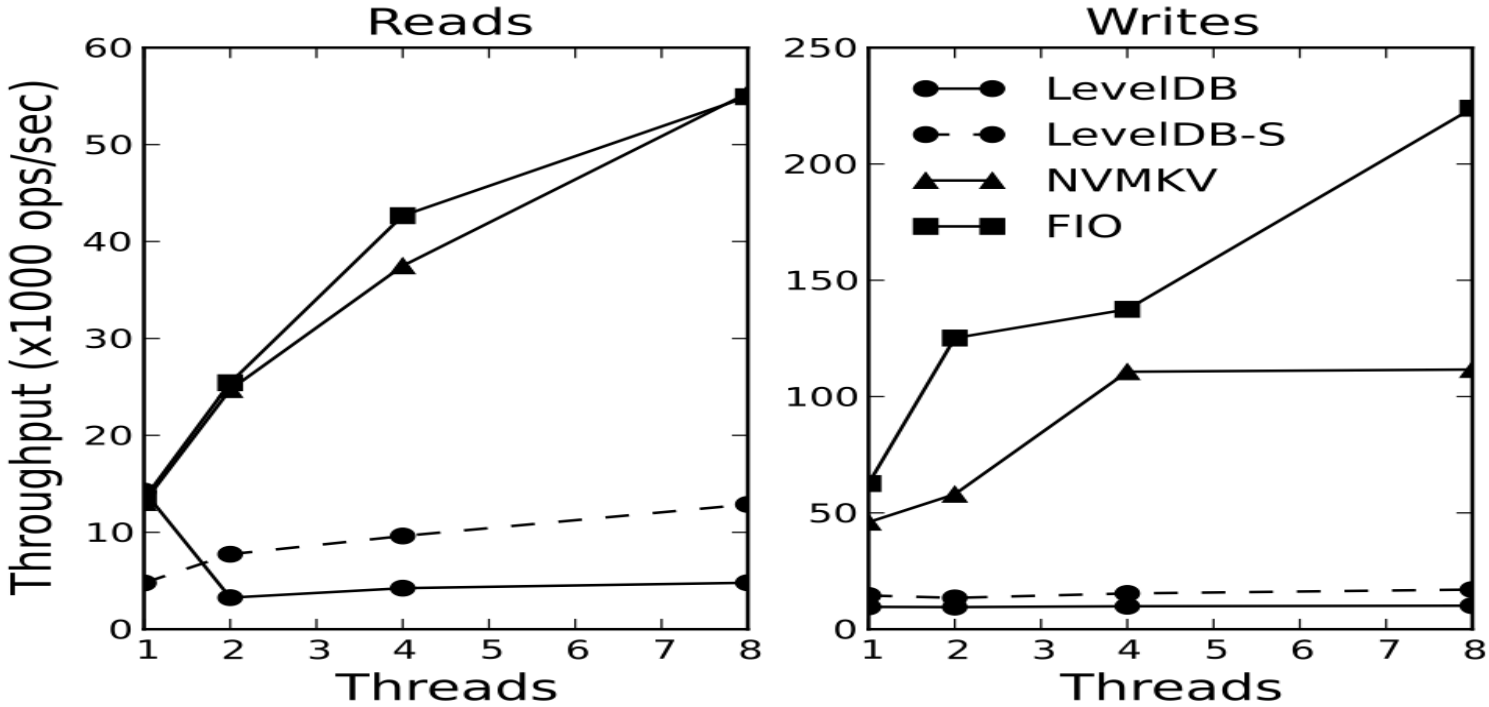


Improving Performance Predictability with Less DRAM



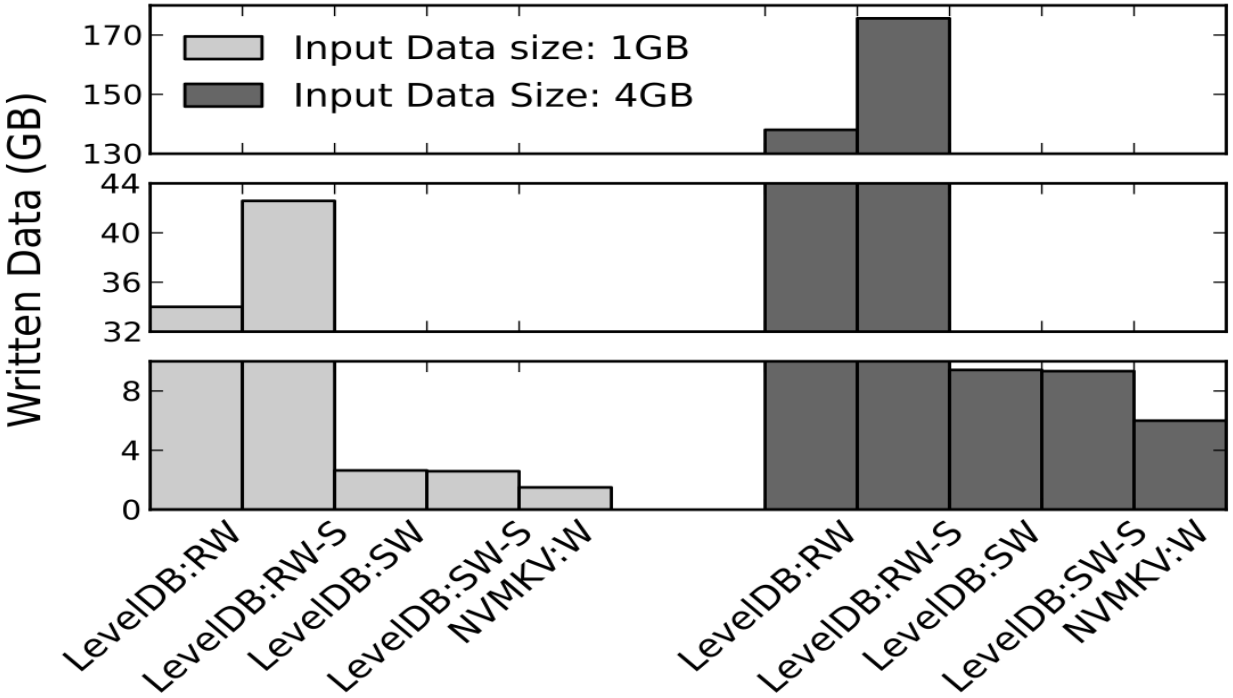
Example: LevelDB

LevelDB, NVMKV, and Raw Block Device



Fusion-io flash aware stack can bring full performance of flash to KV stores

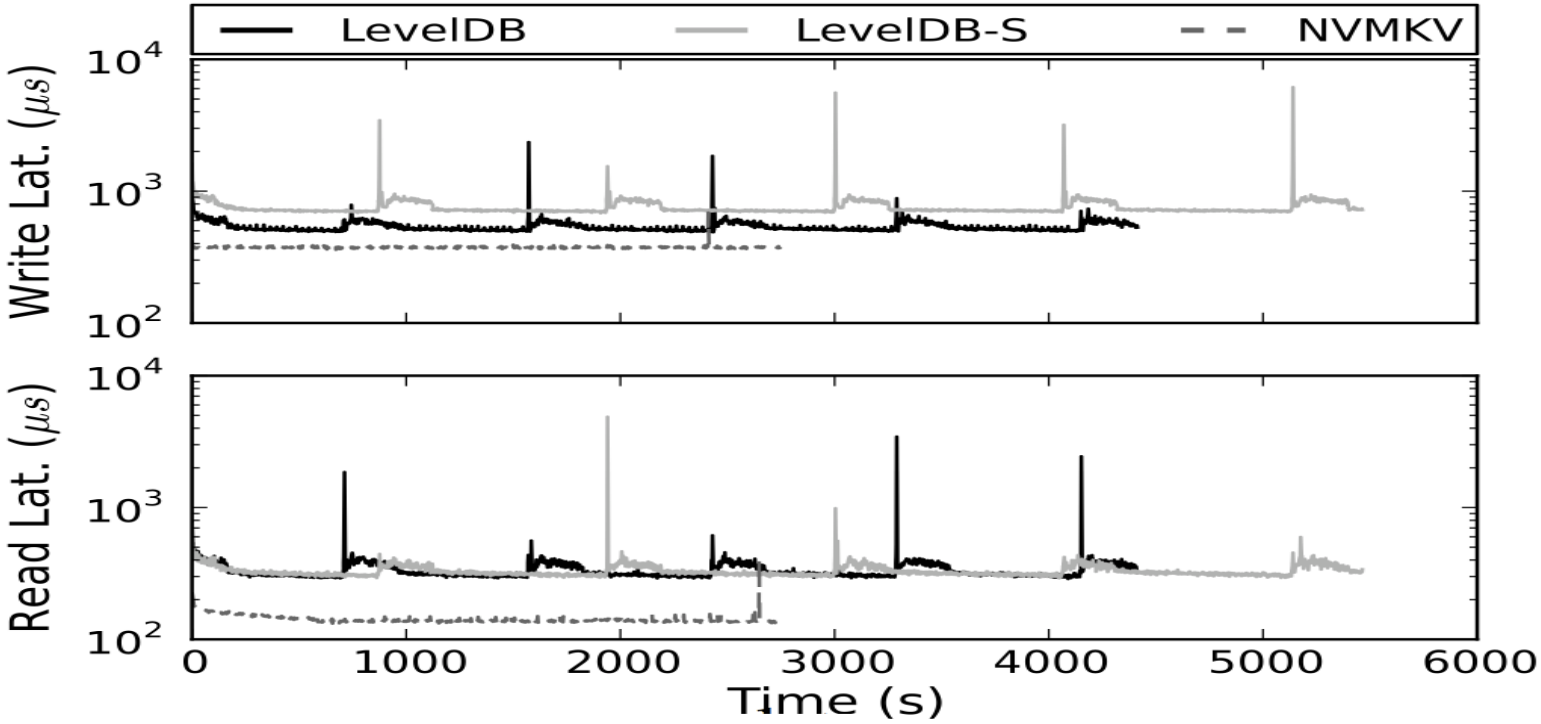
Improving endurance and lifetime with Flash Awareness



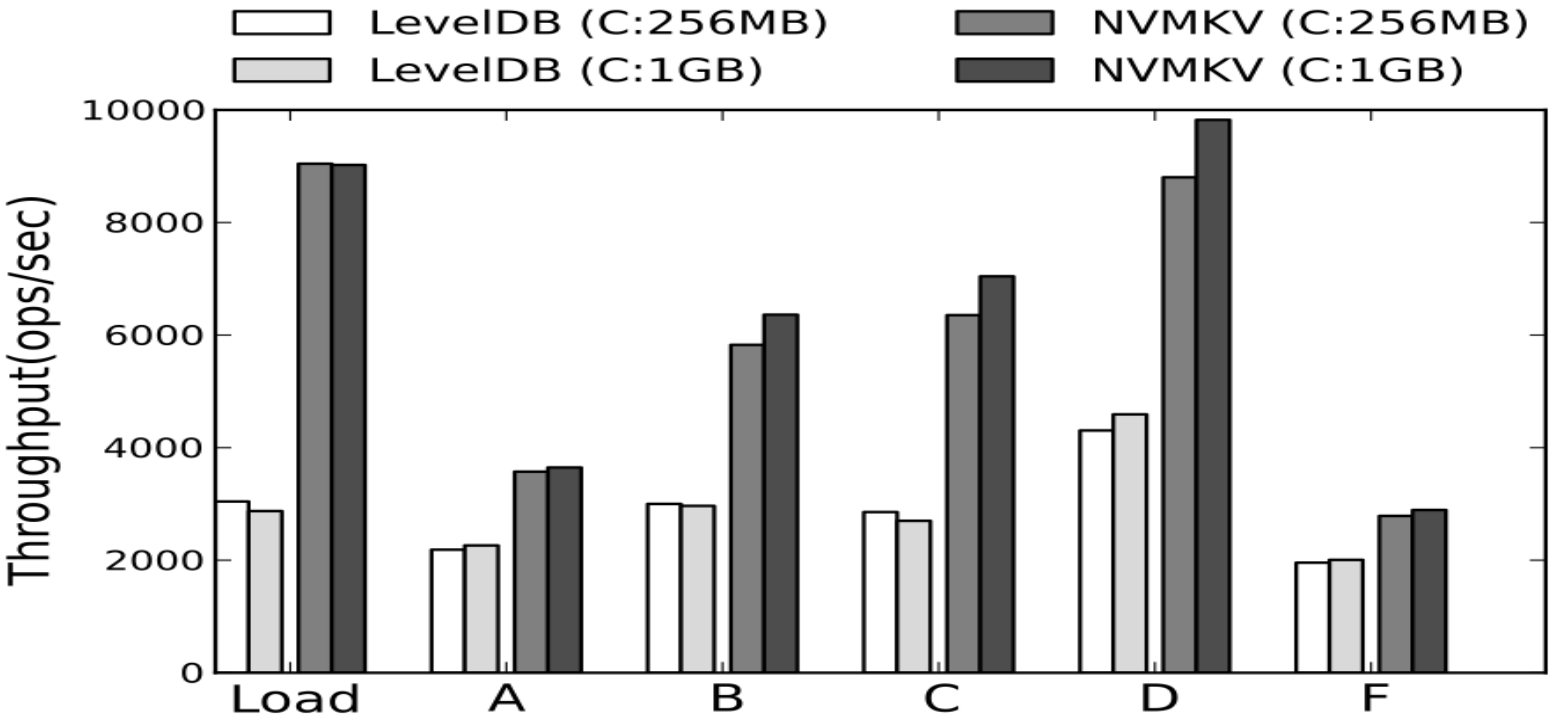
RW: Random Async Write
RW-S: Random Sync Write
SW: Sequential Async Write
SW-S: Sequential Sync Write

Improving endurance by 2x-40x with flash aware key value storage

Performance Predictability



Transaction throughput (YCSB)



Throughput improvements of 50%-300% while using less DRAM

Availability

- Atomic Writes
  - MariaDB mainline >= 5.5.31
  - Percona Server >= 5.5.31
  - Oracle MySQL >= 5.7.4
- NVM Compression
  - MariaDB – 10.0.9
  - Percona Server – 5.6
  - Oracle MySQL – labs release (http://labs.mysql.com/)
- NVMFS available early access
- MongoDB – works with existing MongoDB releases
- NVMKV – available at opennvm.github.io

https://opennvm.github.io

http://www.opencompute.org/projects/storage

Thank You

SanDisk®

Appendix

SanDisk®

File System Support

NVM-Compression uses POSIX compliant Interfaces

POSIX Interface	Functionality
fallocate(offset, len)	Preallocation and extending files/table space
fallocate(PUNCH_HOLE)	Unmap/Punch hole operation. (issue Persistent TRIMs)
io_submit()	AIO Transparent Atomic writes

NVM-Compression speed from NVMFS file system

NVMFS

- **Non Volatile Memory FileSystem**
- A POSIX compliant filesystem, designed by Fusion-io

- **Strengths**
  - Efficient pre-allocation of large files
  - Pre-allocation of large files is efficient
  - No fragmentation/degradation with aging of filesystem
  - Near raw flash speeds for I/O, atomic writes and TRIM

SanDisk®