High-Performance Synchronous Logging using Low Latency MRAM Storage Accelerators

Arthur Sainio

August 6, 2018

Latency in Financial Systems

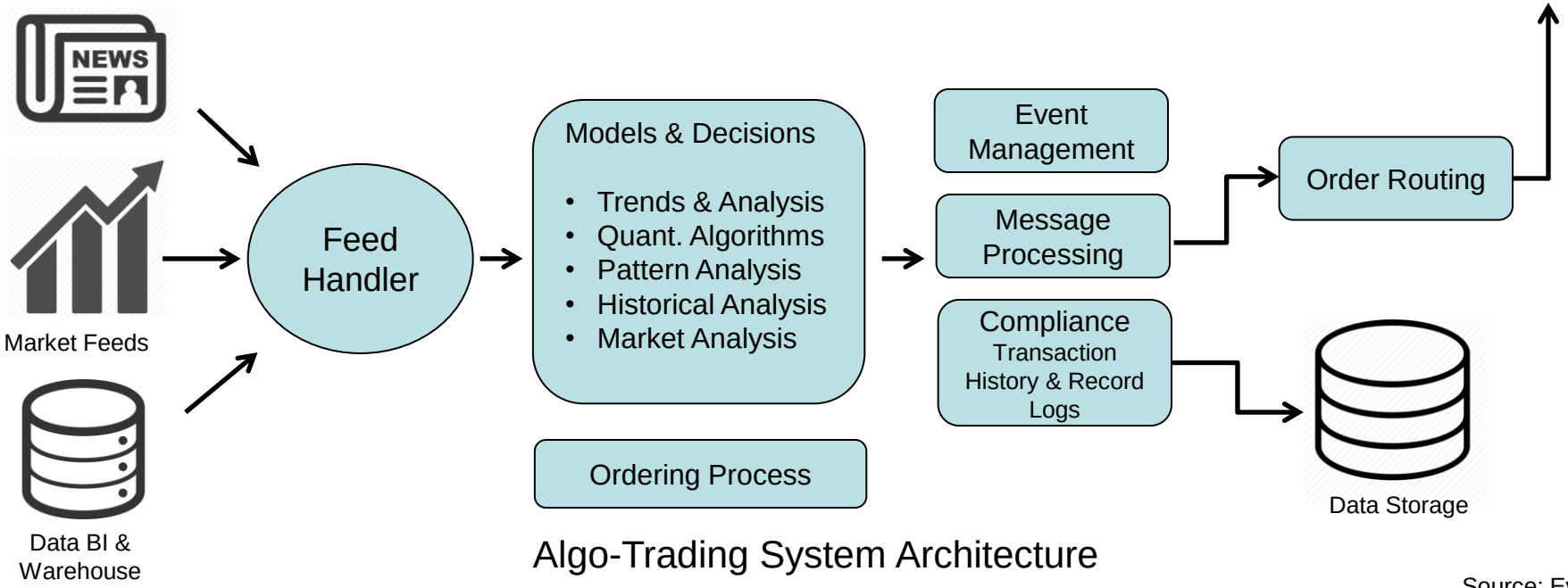
Network Latency ~1µs+

Application Latency (<1µs to 100µs+)

Logging Latency 1ns – 1ms+

Network Latency ~1µs+

Proximity Latency <1µs to100ms+



Algo-Trading System Architecture

Source; Everspin

Synchronous Logging and Trading Performance

- Lowering latency in the logging phase can drastically improve overall trading performance

- Today, log data has to be written on a **storage device** such as high performance SSDs to ensure persistence of log for compliance and client protection. Lower latency and higher write performance will improve trading performance

- Higher trading performance may imply more revenue

- It is possible to do asynchronous logging but DRAM buffers based log data could be lost at power loss

- For compliance reasons, trading of some financial assets may be required to be synchronously logged

Challenge of Regulations on Trading Performance

- Financial Industry Regulatory Authority (FINRA), Markets in Financial Instruments Directive (MiFID II) and many other financial regulatory bodies or requirements are now demanding that transacting clients log & timestamp (global) order details, transaction communications, as well as log all order status changes such as:

  - Accepted for bidding
  - New
  - Rejected
  - Pending new
  - Partially filled
  - Canceled
  - Expired
  - Suspended
  - Stopped
  - Filled
  - Calculated
  - Done for day
  - Pending replace
  - Pending cancel

Question: *What happens if order execution information and transaction data (logs) are lost?*

MRAM = Low Latency + Persistence

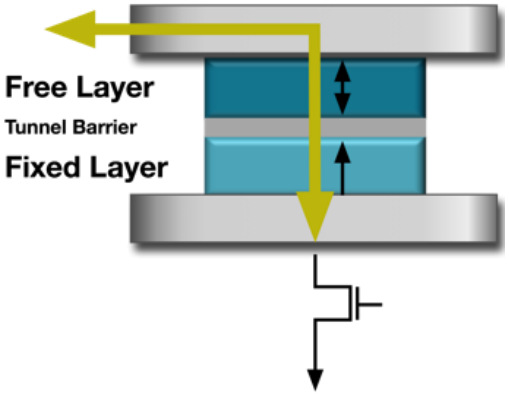
Low Latency

Data Persistence

Data Integrity

Performance

- Writes to a memory array by manipulating electron spin with a polarizing current
- Performs like DRAM but requires no refresh
- Highly scalable, enabling higher density memory products



MRAM Storage Accelerators

- Lowest latency (~6µs) and highest performing NVMe products for 100% 4KB RW IOPs (~1.5M)

- Removes system bottlenecks by accelerating synchronous logging data being written to SSDs

- No need for backup power

- Plug and Play - No need for additional drivers

- Save money by not needing to use Enterprise SSDs
- Save money by not requiring a high core count CPU to increase performance

MRAM Storage Accelerators

- NVMe 1.2.1
- Form Factors
  - HHHL PCIe
  - U.2
  - Others
- PCI Gen3 x8 HHHL PCIe card
- PCI Gen3 x4 U.2
- Ultra-low latency ~6µs
- High Random Write (100%) 4KB IOPS ~1.5M
- Durable, persistent memory
- Two access modes:
  - Memory mapped (MMIO)
  - Block based (NVMe 1.1+)
- Peer to peer (PeerDirect) for RDMA access
- Out of box performance with standard Windows/Linux NVMe drivers
- No need for backup power

Low Latency Synchronous Logging

Incoming Messages

- Variable Rate
- Bursts
- Latency Sensitive
- Written in bytes

Apache LOG4J™

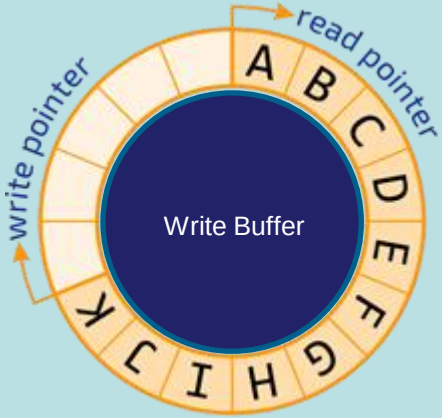
Java

Logging Applications

Logging Accelerator Software

Custom Appender

JNA



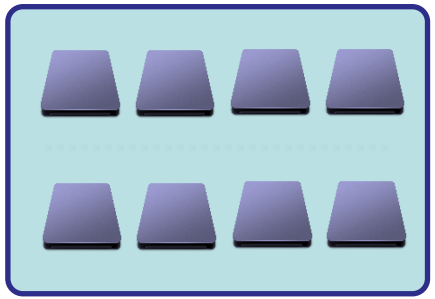
Raw I/O Writer (written in C)

Raw I/O Reader (written in C)

Outgoing Messages

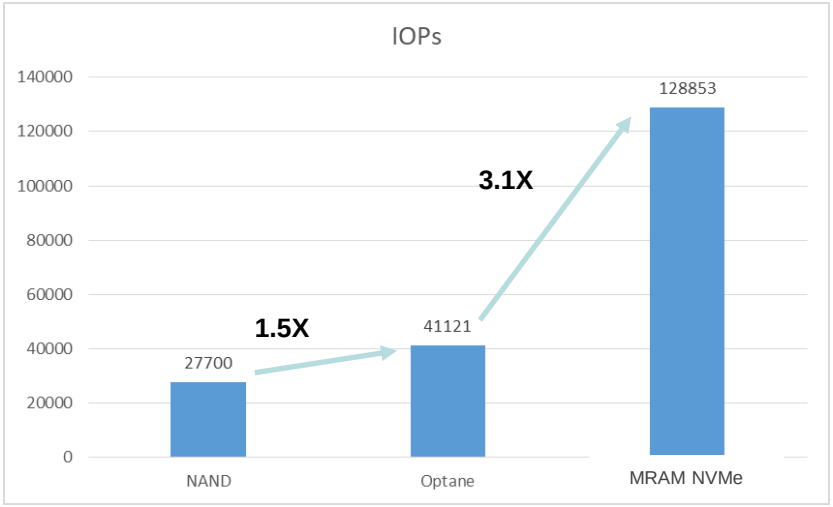
- Written to in big blocks

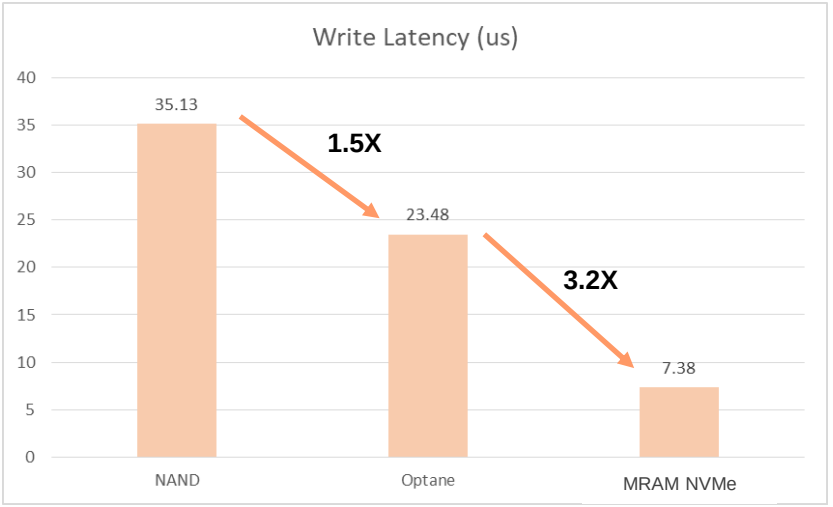
Array of client SSD's



HIGHER LOGGING PERFORMANCE WITH MRAM STORAGE ACCELERATION

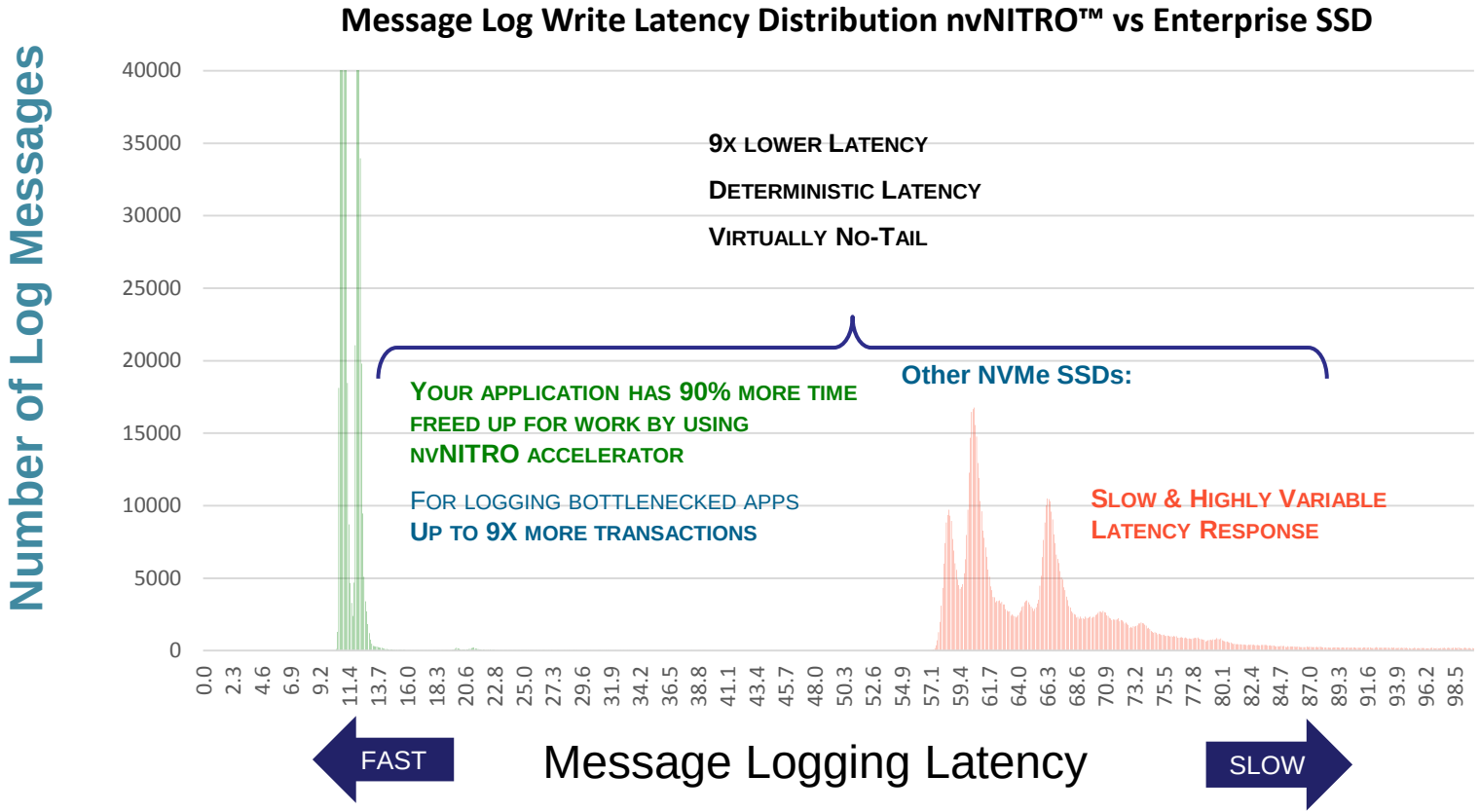
Random Write Performance Comparison
QD1, T1, 4KB BS





- Live demo system; Supermicro X11DPI system with a 2.1GHz 8160 Platinum Skylake processor comparing a MRAM Storage Accelerator vs. an Optane NVMe SSD

Higher Logging Performance



MRAM Storage Accelerator Applications

Synchronous Logging	• Low Latency transactions, application or error logging for industries such as FinTech etc.
Power Loss-Protected Circular Buffers	• Standard and custom I/O applications requiring high endurance write buffering for high performance, randomized and low latency data stream patterns to slower storage devices
NVMe over Fabric	• Low latency write buffer for all flash storage arrays (AFA) to front-end the NVMe drives in the NVMe flash array connected over 100Gb RDMA networks
Application Checkpointing	• Front-end storage for saving application state with extremely low I/O write latency and very high write bandwidth. Bit coining etc. could benefit
Database	• Log Cache Buffer/ Journals less than 1GB size (Possibly for MySQL ib_logfile0 & ib_logfile1 files)

Thank You!