

Capacity. Persistence. Performance.
Scale-in Software for Capital Markets Computing



Challenge: Maximizing modern hardware

Negative impacts of RAM limits, disk I/O and accelerator I/O:



1

RAM limits require more hardware and more time splitting data and distributing jobs

2

Persisting work-in-progress relies on storage I/O, which slows everything down

3

Low-level programming is needed to optimally feed HW accelerators

Solution: RAM-level performance using NVM



Helium is an ultra-fast system software that solves all three problems:



1

High-performance system-level data access software that emulates RAM using SSDs, expanding addressable memory into the tens of terabytes

2

Provides persistence with negligible performance penalty while allowing multiple uses of data dictionaries

3

Lets applications bypass OS to address FPGAs & GPUs without complex, low-level coding

HELIUM™

Process More Data, Faster



With Helium, you can:



1

**Dramatically expand
direct access to data
without adding
servers, or...**

2

**...achieve current
performance levels at
a fraction of the price*,
and...**

3

**...put server-level
capability on
workstations or
laptops**

* A server with 1TB of SSD is about ¼ the cost of one with 1TB of RAM

Use Cases



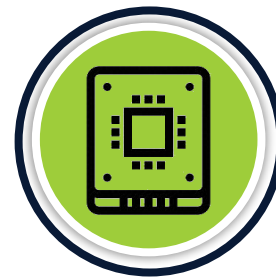
Financial risk analytics



Persistent data frames
(Python/Spark)



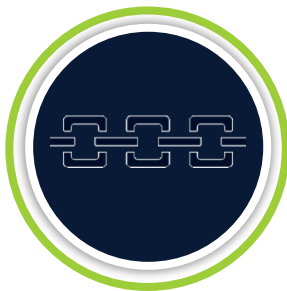
Financial back testing



Storage engine for SSD
optimized DBs



Time series DBs



Blockchain and DL



Java object persistence
(hibernate)



Trade matching engines

Use Case: Blockchain / Distributed Ledger



Distributed ledger technology is built on blocks and transactions pointing to each other



These pointers are typically saved using databases like RocksDB or LevelDB



A single transaction might consist of thousands of DB lookups and inserts



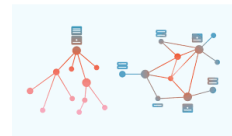
Helium as the storage engine dramatically increases transaction rates, reduces latency and jitter



RocksDB



levelDB

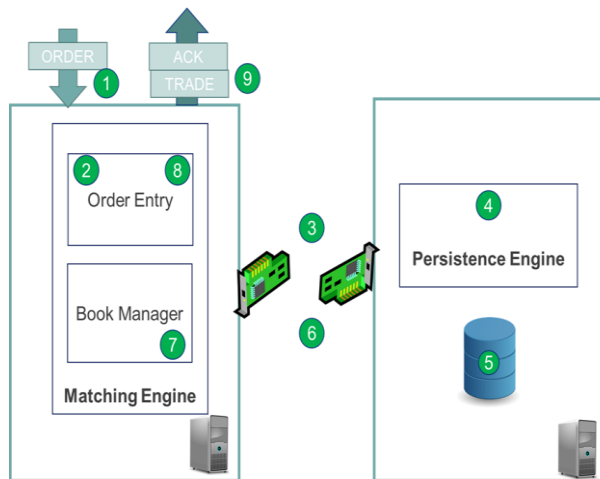


>10X Increase

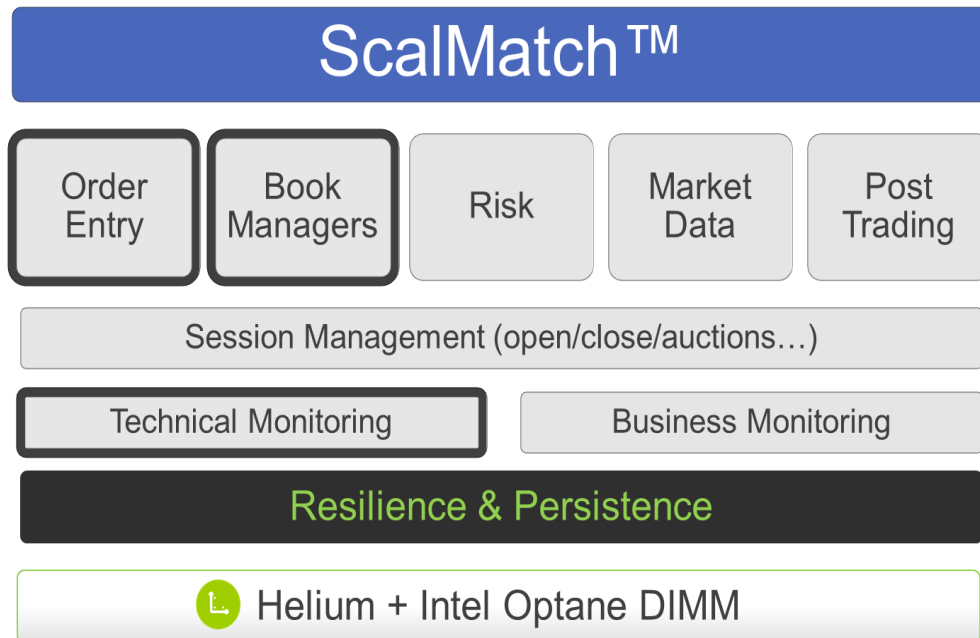
Use Case: Highly Available Matching Engine with Near Zero Risk of Data Loss



Typical Trading Architecture



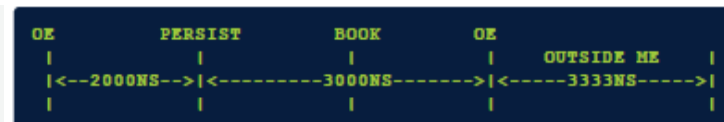
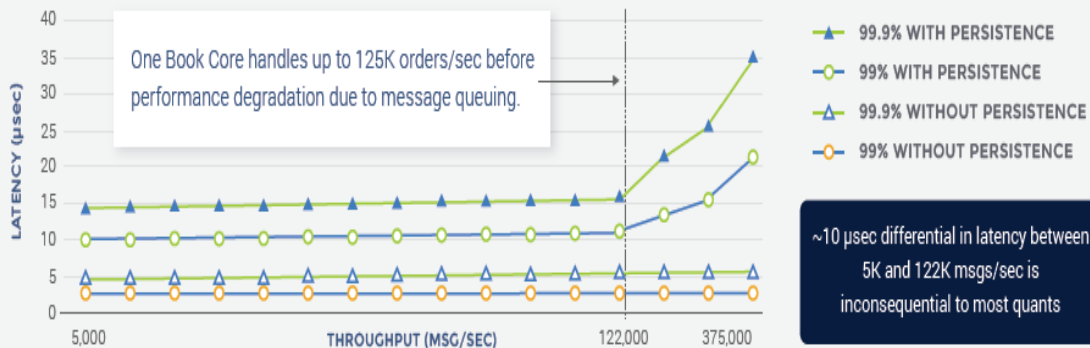
Trading Architecture: Combining Levyx, Scalnyx & Intel Cascade Lake with Optane DIMMS



Use Case: Highly Available Matching Engine with Near Zero Risk of Data Loss



LATENCY COMPARISON OF 1 BOOKMANAGER CORE, WITH AND WITHOUT PERSISTENCE, I.E. HELIUM

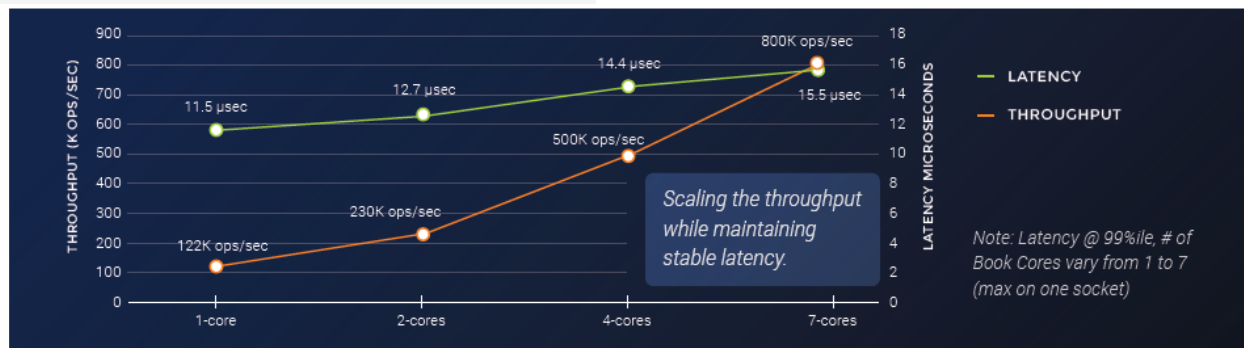


Messages lost only if the messages are between OE and PERSIST

Messages Lost $\geq 2,000/8,333$ or **0.24 messages** on average because:

- Messages that are in OE are refused (so not lost)
- Messages that are only in PERSIST can be replayed

* NOT STAC BENCHMARKS



Business Benefits



Freedom

Through dramatically expanded addressable application memory



Confidence

Through automatic persistence



Efficiency

Through concurrent access to the same data dictionaries by multiple users



Speed

Through optimized data flow for maximum CPU/FPGA/GPU processing utilization



Savings

Through leveraging lower cost SSD over higher cost RAM

Thank You

Try the Free community version at <https://helium.levyx.com>



Levyx