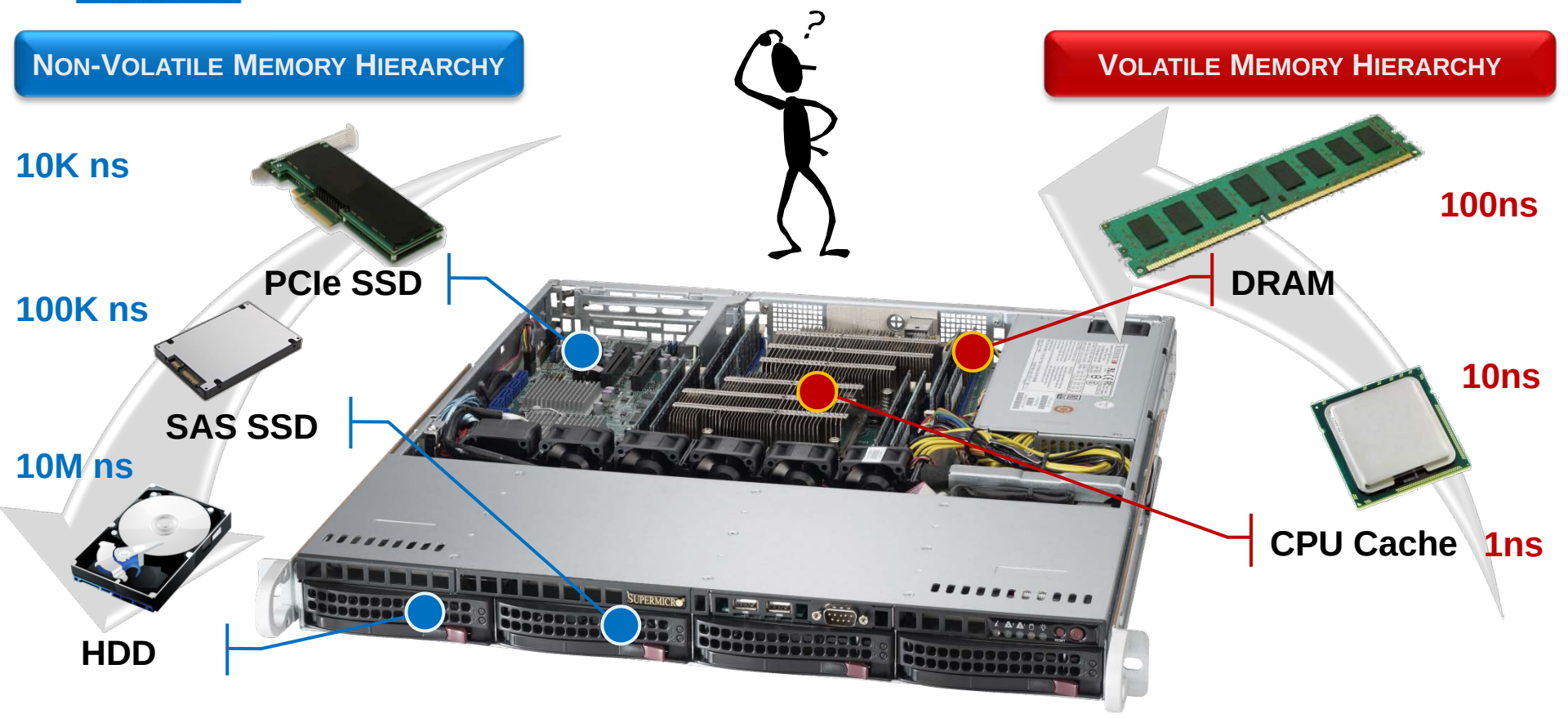
Benchmark: In-Memory Database System (IMDS) Deployed on NVDIMM

Presented by Steve Graves, McObject

and Jeff Chang, AgigA Tech

The Problem: Memory Latency

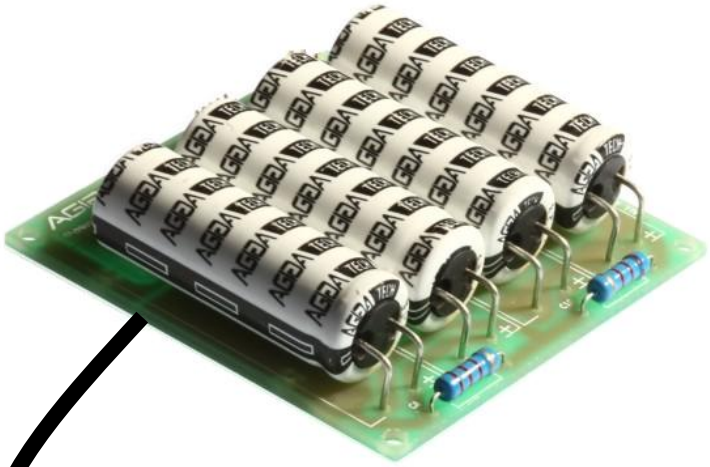


As CPU technology scales with Moore's Law, memory IO creates significant performance bottlenecks

Huge latency gap in memory hierarchy between volatile and non-volatile technologies

Latency gap will widen with the introduction of DDR4

The Solution: Non-Volatile DIMMs

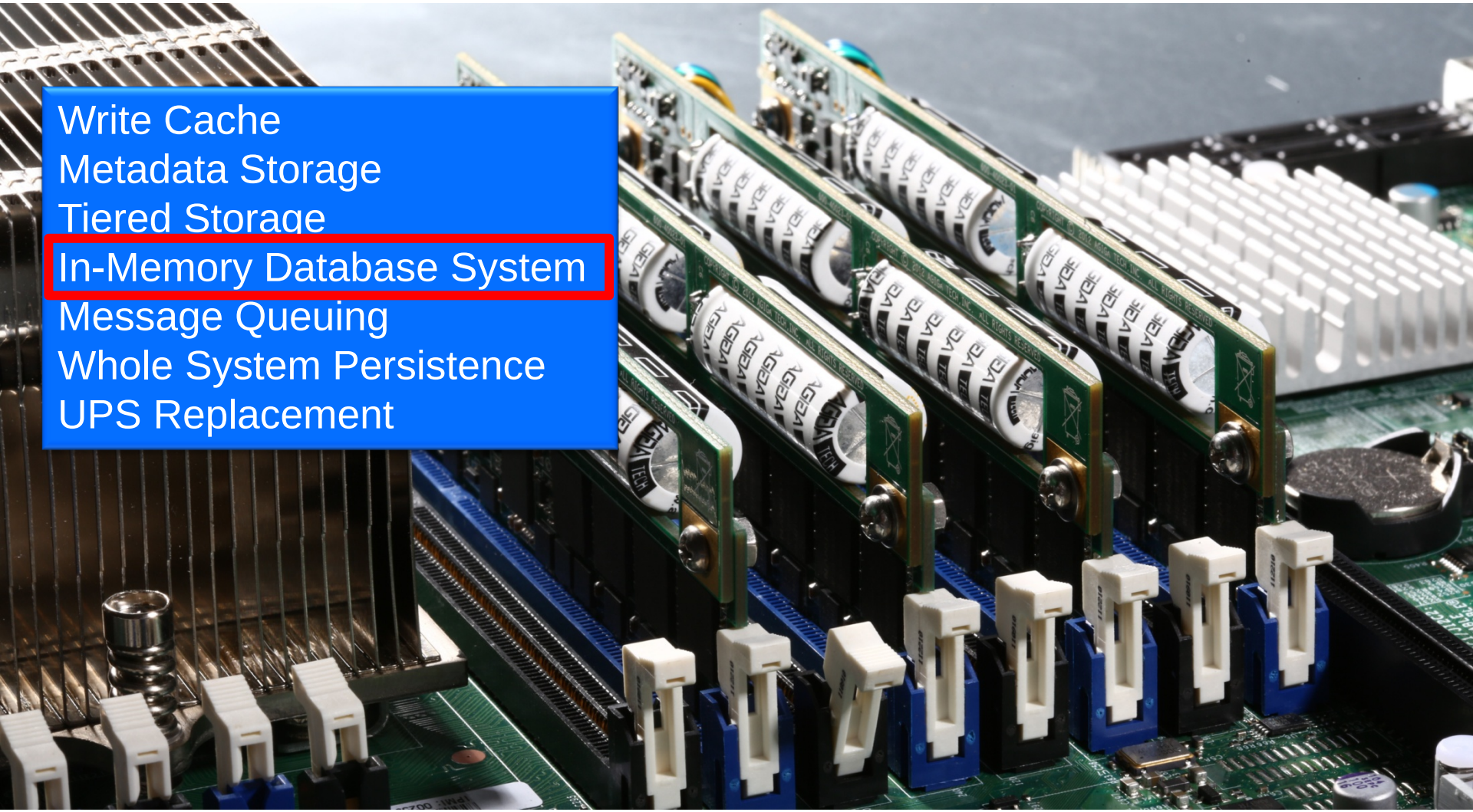


Looks Like DRAM, Acts Like Flash

Ultracapacitors
Provide Power During Backup
Fast charge time
No maintenance
High reliability
Environmentally safe

NVDIMM
Moves DRAM contents to NAND Flash during power loss
Restores data on system recovery
Fits in standard JEDEC DIMM socket

NVDIMM Use Cases



- Write Cache
- Metadata Storage
- Tiered Storage
- In-Memory Database System
- Message Queuing
- Whole System Persistence
- UPS Replacement

In-Memory Database System (IMDS)

- Definition: database management system that stores all records in main memory
- Contrast to traditional DBMSs premised on disk storage for all data
- IMDSs eliminate
  - Disk and file I/O
  - Cache processing
  - Data transfer
- Result: IMDSs perform orders of magnitude faster
- Increasingly popular for business analytics, telecom, capital markets, industrial control and more

IMDSs and Volatility

- DRAM's volatility is viewed as an IMDS disadvantage – how can an IMDS gain data durability?
- Transaction logging
  - Pro: logging enables recovery of committed transactions in the event of system failure
  - Con: logging re-introduces persistent write overhead (still outperforms traditional DBMS – but not as fast as "pure" IMDS)
- Battery-backed RAM
  - Pro: retains data on DRAM chip in event of power failure
  - Con: restrictive temperature requirements, limited shelf life, risk of leaking corrosive and toxic fluids, etc.
- Emerging NVDIMMs: can they lend IMDSs durability without performance overhead or battery disadvantages?

McObject – AgigA Tech IMDS & NVDIMM Benchmark Tests

- Industry's first test of IMDS using NVDIMM as main memory storage
- Sought to measure performance and confirm data recoverability/durability
- Test system:
  - Intel Oak Creek Canyon reference motherboard with Intel Pentium Dual Core CPU 1407 @ 2.8 GHz processor
  - 8 GB Kingston conventional DDR3-1333 DRAM
  - Debian Linux 2.6.32.5
- McObject's *eXtreme*DB In-Memory Database System with AgigA Tech's 4 GB AGIGARAM DDR3 NVDIMM as IMDS storage

Benchmark Test Results

- Performance: NVDIMMs matched conventional DRAM for all database operations – inserts, updates, deletes, index searches and table traversals
- Data durability: Following mid-execution re-boot, benchmark application
  - Re-started automatically
  - Accessed *eXtreme*DB database in pre-failure state (on recovery, NVDIMM had loaded it from its flash into its DRAM)
  - Checked for database consistency and resumed operation
- This recovery leveraged an *eXtreme*DB feature added in 2003 to enable use of the IMDS w/ battery-backed RAM
  - Recovery algorithm can identify, and re-open and re-use memory block of database memory device assigned pre-failure

Discussion

- Tests: NVDIMMs and IMDSs together deliver "persistence without the performance penalty"
- Promising combination for systems that demand both speed and data durability (telecom/networking, capital markets, automation, etc.)
- Other considerations:
  - Cost (conventional DRAM vs. NVDIMM)
  - Hardware platform compatibility
  - Support of NVDIMMs by other IMDS products (benchmark shows it is possible with *eXtreme*DB using "hooks" originally added to support battery-backed RAM)