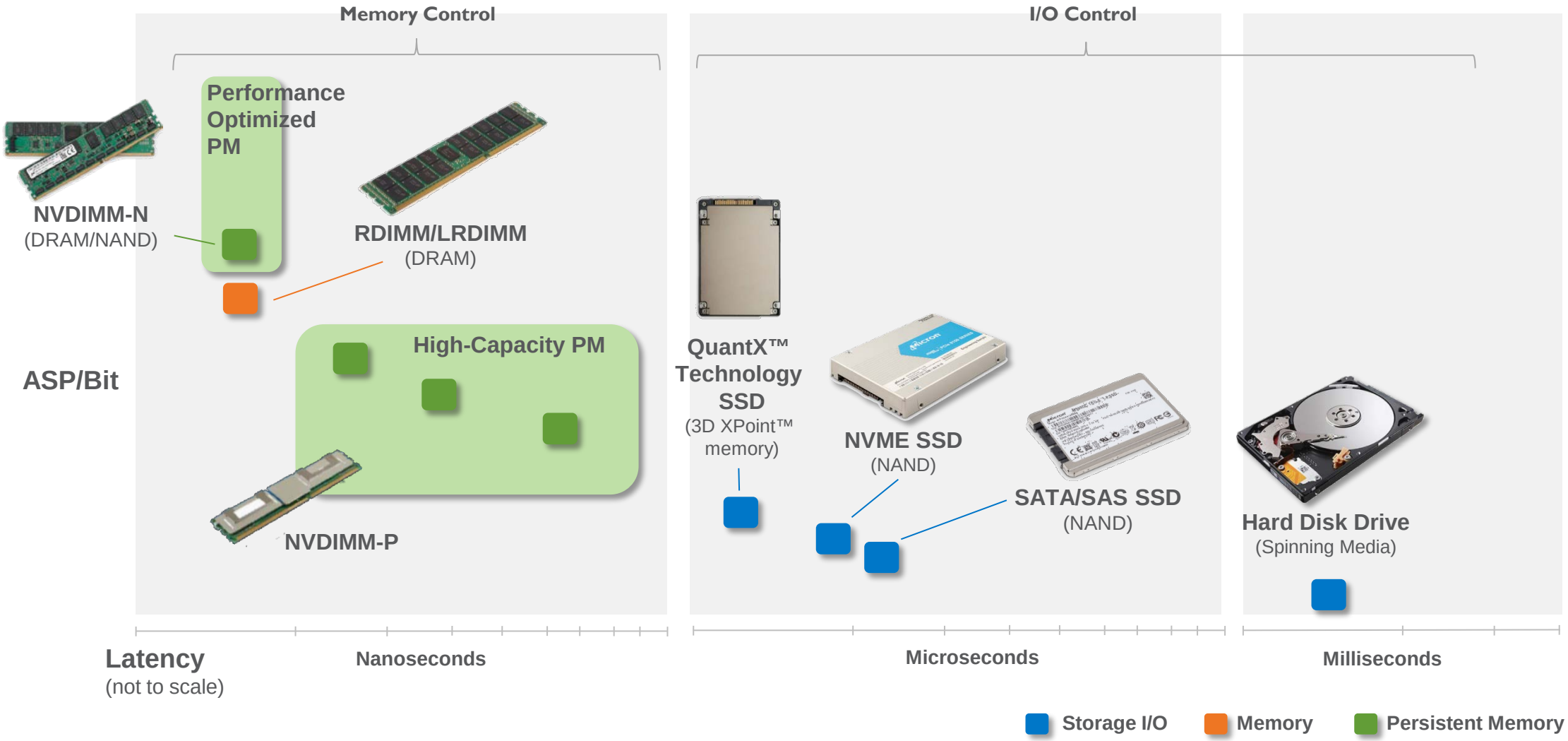
Memory and Storage-Side Processing

How persistent memory will bring an entirely new structure to large data computing

Steve Pawlowski, VP of Advanced Memory Systems

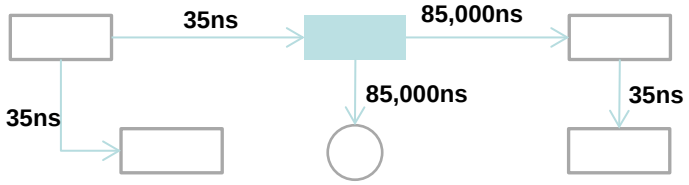
PERSISTENT MEMORY TODAY

Closing the Latency Gap

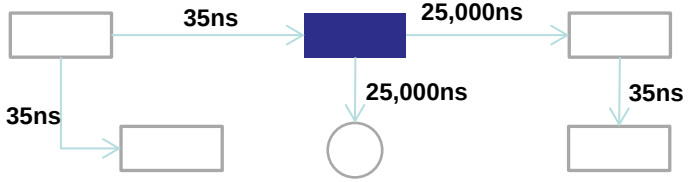


NVDIMM-N and Application Performance

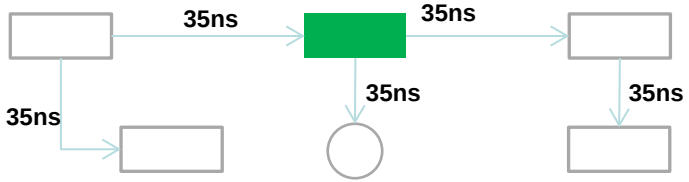
DRAM + NAND-Based NVMe SSD

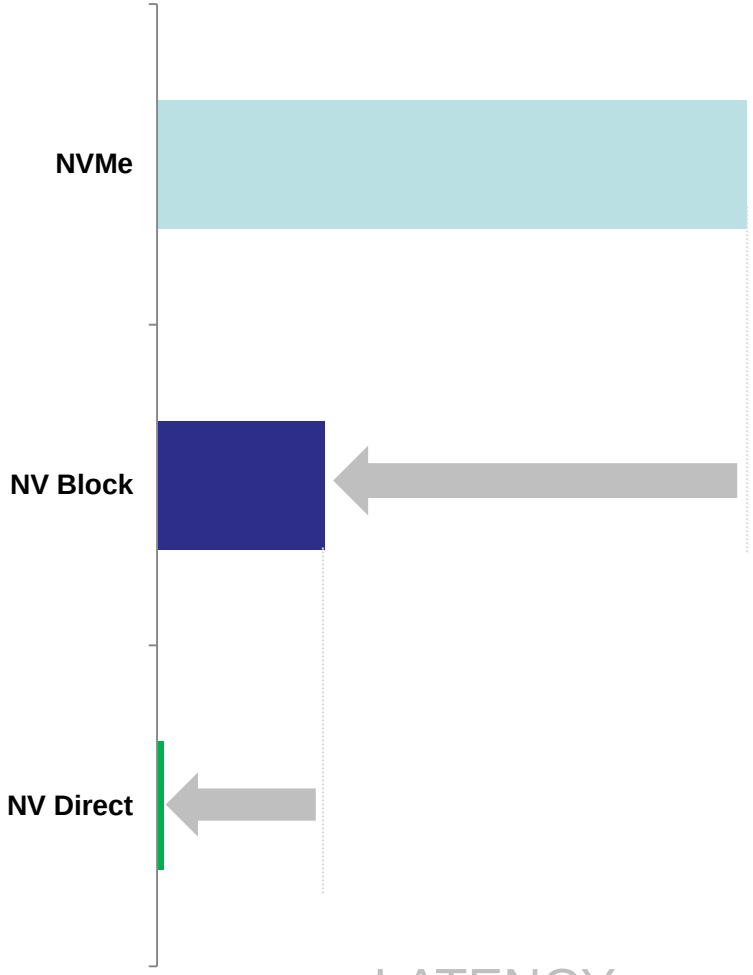


DRAM + NVDIMM-N Block Mode



DRAM + NVDIMM-N Direct Mode



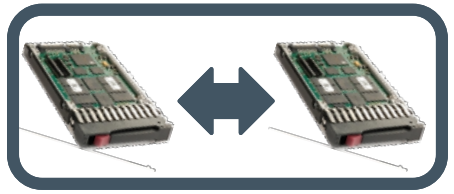


- Data committed to persistent media written to ***NAND*** through the I/O stack

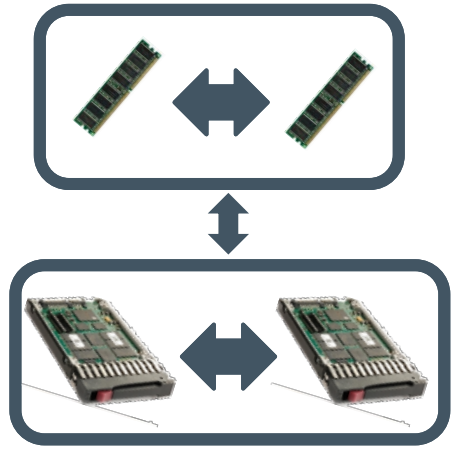
- Data committed to persistent media written to ***DRAM*** on NVDIMM-N through the I/O stack
- Latency improvement due to writing to NVDIMM (25,000ns) versus SSD (85,000ns)

- Data committed to persistent media written to ***DRAM*** on NVDIMM-N through Load/Store Bus
- Latency improvement due to circumventing overhead associated with I/O stack

Microsoft® Enterprise Performance



Mirrored SSD Pool



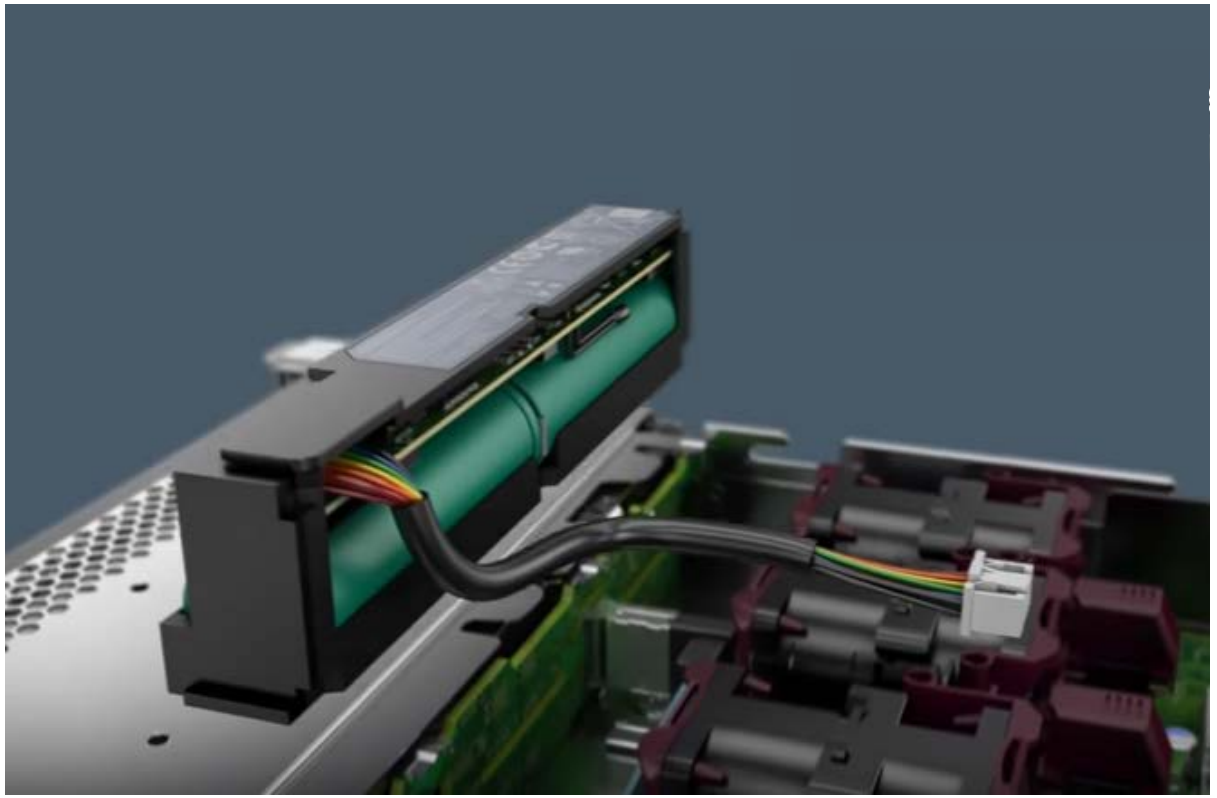
Dual 8GB mirrored NVDIMM cache + mirrored SSD pool

	Mirrored SSD Pool	Dual 8GB mirrored NVDIMM cache + mirrored SSD pool	
SQL Server Transaction Log	970k Tx/min 372us write latency	1.08M Tx/min 181us write latency	**+11% Performance** (Block Mode, Read Centric)
Exchange Server	780k Tx/min	1.3M Tx/min	**+63% Performance** (Block Mode, Write Centric)

CHALLENGES TO OVERCOME

Driving Easier Market Adoption

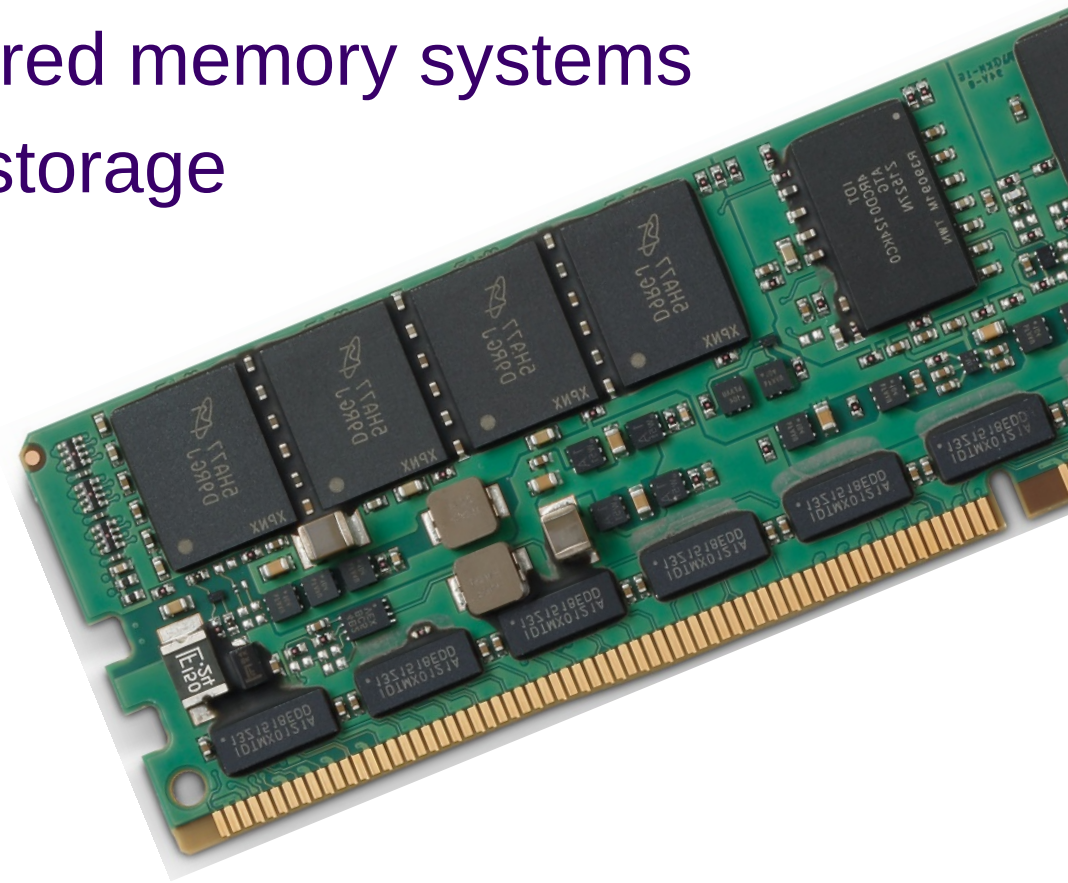
- Simplify the power source
- Minimize SW and programming ecosystem impact
- Appropriate energy/performance tradeoffs
- Address data security
  - HW encryption, AES-256
- Latency optimization



HPE® Smart Storage NVDIMM Integrated Backup Power

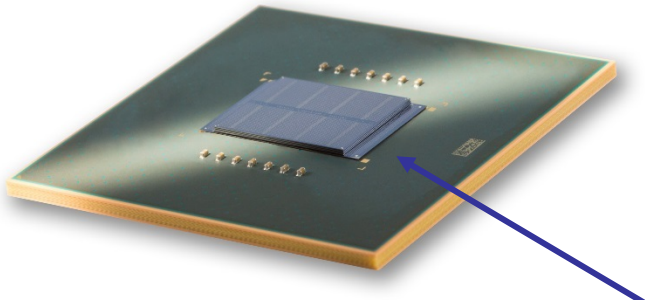
Implications of NVDIMM-P

- High capacities enable truly large in-memory computing
- Latencies introduce opportunity for tiered memory systems
- Introduces the concept of distributed storage
- Compute functionality in memory
  - (more on this in a minute)
- Industry ecosystem should begin preparing today



FUTURE VISION

Hybrid Memory Cube



NVDIMM-N

Integrated on-board controller

This opens interesting new possibilities for managing data

Moving Data to the Processor is Costly

One floating-point calculation

17 picojoules

Moving data from DRAM to CPU

17,000 picojoules

Opportunities for **1000x improvement** are increasingly rare

Why Hasn't it Been Done Before?

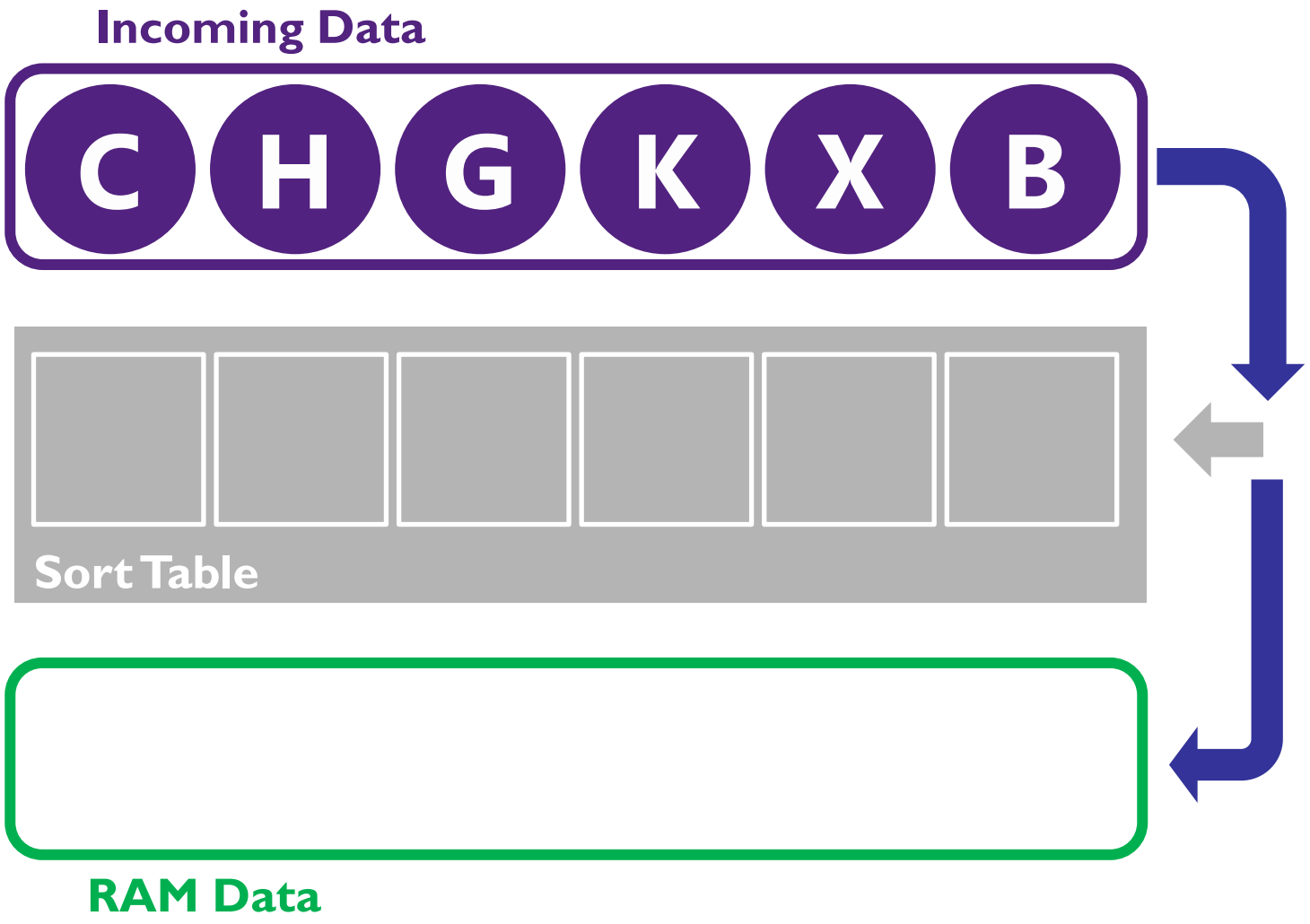
Memory Technology	Processor Technology
Low leakage transistors	High speed
Low power	High power

- Processors and memory have distinct abilities—they must act symbiotically
- DRAM transistors are best when:
  - In-memory processing is very simple
  - Processing is intrinsically data-centric
  - Actions are complementary to CPU / Custom Processors

Where Do These Conditions Exist?

- Data preparation
- Data sort
- Databases

An Illustration



- Sort keys are paired with data as it is written
- Data can be read:
  - As written
  - Based on sort keys

What Can Be Done with Sort in Memory?

- Data-centric problems
  - Data sort
  - Databases
  - Machine learning (90% of the problem is feeding the MACs)
- Computationally lightweight problems
- Solutions that can be smoothly transitioned:
  - Replace a library call (qsort ->msort)
  - Replace a subsystem (MySQL)

Questions?