Memories of Tomorrow

Coughlin Associates

Tom Coughlin, Coughlin Associates

&

Jim Handy, Objective Analysis
REMEMBER WHEN PERSISTENT MEMORY REALLY CAUGHT ON BACK IN THE TEENS?

THAT WAS A LONG TIME AGO. I'M NOT SURE...
How Have New Memories Gotten Funding?

A new technology will cost less than the incumbent if it continues to scale.
The Scaling Limit Has Been Looming For Years
New Memories Extend Scaling

Limited Scaling
- Flash
- DRAM

Extended Scaling
- ReRAM
- MRAM
- PCM
- FRAM
- CBRAM
- NRAM
But Wait! There’s More!

- New memories are all much better than Flash & DRAM:
  - Nonvolatile (persistent)
  - Bit writes
  - No pre-erase requirement
  - Much faster than flash
  - Lower power (no refresh, low write energy)
  - Cheaper (through scaling)

Storage Class Memory!
What is Storage Class Memory?

Storage-class memory (SCM) combines the benefits of a solid-state memory, such as high performance and robustness, with the archival capabilities and low cost of conventional hard-disk magnetic storage.

IBM Almaden Research Labs
What is Storage Class Memory?

Storage-class memory (SCM) combines the benefits of a solid-state memory, such as high performance and robustness, with the archival capabilities and low cost of conventional hard-disk magnetic storage.

*IBM Almaden Research Labs*
Well, That’s Just Core!
Goal: Rethink Everything
  - From Turing & von Neuman to now

Why IEEE?
  - Encompasses the whole computing stack
Tom Conte, Co-chair IEEE Rebooting Initiative, Professor, Georgia Tech
Potential Approaches vs. Disruption in Computing Stack

LEGEND: No Disruption | Total Disruption

Algorithm
Language
API
Architecture
ISA
Microarchitecture
FU
logic
device

“More Moore”
Level 1

2
Hidden changes
3
Architectural changes
4
Non von Neumann computing

IEEE rebooting COMPUTING
More Moore

- Further scaling
  - New device structures:
    - FinFET, 3D NAND, new switch, unreliable switch, etc.

- Non-silicon device types:
  - MRAM
    - Spintronic systems
  - PCM
  - Memristors
  - FRAM
Potential Approaches vs. Disruption in Computing Stack

Legend: No Disruption | Total Disruption

Level 1: “More Moore”

Levels 2, 3, and 4: Architectural changes

Non von Neumann computing
Hidden Changes

- Changes that don’t impact the Instruction Set
  - Devices
  - Logic
  - Functional Units
  - Microarchitecture
- Context switch issue fits here (described shortly)
- Microarchitectural support for non-deterministic logic devices
- No significant impact to memory or storage
Potential Approaches vs. Disruption in Computing Stack

LEGEND: No Disruption | Total Disruption

- Algorithm
- Language
- API
- Architecture
- ISA
- Microarchitecture
- FU
- Logic
- Device

- Non von Neumann computing
- Architectural changes
- Hidden changes
- "More Moore"

Level 1
Level 2
Level 3
Level 4
Architectural Changes

- Beyond x86
  - GPU
  - FPGA
- Moving computation to the data
  - In-memory compute
  - \textit{In-situ} processing
- Special purpose computing
- Requires new programing
Potential Approaches vs. Disruption in Computing Stack

LEGEND: No Disruption  Total Disruption

1. Algorithm
2. Language
3. API
4. Architecture
5. ISA
6. Microarchitecture
7. FU
8. Logic
9. Device

"More Moore" vs. Architectural changes

Level 1 vs. Hidden changes

Non von Neumann computing
Non von Neumann Architectures

- Quantum computers
- Neuromorphic computing
- Analog Computing
- Very immature, lots of investment
Artificial Intelligence useful in:

- Internet of Things
- Voice and Image recognition
- Autonomous Vehicles
- Cybersecurity
- Industrial automation
- And many other uses
All of These Need Storage & Memory
Two Major Problem Areas:

- Memory loses its data (volatility)
- Storage is slow (latency)
Curing Latency

- Move compute close to the data
- Use lower-latency storage

More on these later
Curing Volatility

- Old Way: Frequent verified copies to storage
  - Faster storage helps this a lot!
- New Way: Make the memory persist
  - Batteries
  - Flash backups
  - New persistent memory types

That’s Storage Class Memory (SCM)!
Solution 1: Persistent Memory

- NVDIMM
- Optane
- Other emerging memories
- Issues: Context switch, cost, maturity, S/W support...
NVDIMM
## Today’s Memory Types

<table>
<thead>
<tr>
<th></th>
<th>SRAM</th>
<th>DRAM</th>
<th>ROM</th>
<th>EEPROM</th>
<th>NOR</th>
<th>NAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonvolatile</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Erasable</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Programmable</td>
<td>Yes</td>
<td>Yes</td>
<td>Factory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Smallest Write</td>
<td>Byte</td>
<td>Byte</td>
<td>N/A</td>
<td>Byte</td>
<td>Byte</td>
<td>Page</td>
</tr>
<tr>
<td>Smallest Read</td>
<td>Byte</td>
<td>Page</td>
<td>Byte</td>
<td>Byte</td>
<td>Byte</td>
<td>Page</td>
</tr>
<tr>
<td>Read Speed</td>
<td>V Fast</td>
<td>Fast</td>
<td>Fast</td>
<td>Fast</td>
<td>Fast</td>
<td>Slow</td>
</tr>
<tr>
<td>Write Speed</td>
<td>V Fast</td>
<td>Fast</td>
<td>N/A</td>
<td>Slow</td>
<td>Slow</td>
<td>Slow</td>
</tr>
<tr>
<td>Sleep Power</td>
<td>V Low</td>
<td>High</td>
<td>Zero</td>
<td>Zero</td>
<td>Zero</td>
<td>Zero</td>
</tr>
<tr>
<td>Price/GB</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Med</td>
<td>V Low</td>
</tr>
<tr>
<td>Applications</td>
<td>Small Fast</td>
<td>Main Memory</td>
<td>Stable Code</td>
<td>Serial #, Trim</td>
<td>Code</td>
<td>Data</td>
</tr>
</tbody>
</table>

**Legend:**
- **V Fast:** Very Fast
- **Fast:** Fast
- **N/A:** Not Applicable
- **Slow:** Slow
- **High:** High Power
- **Low:** Low Power
- **Med:** Medium Power
- **Zero:** Zero Power
New Memories Perform Better

<table>
<thead>
<tr>
<th></th>
<th>MRAM</th>
<th>ReRAM</th>
<th>FRAM</th>
<th>PCM</th>
<th>XPoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonvolatile</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Erasable</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Programmable</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Smallest Write</td>
<td>Byte</td>
<td>Byte</td>
<td>Byte</td>
<td>Byte</td>
<td>Byte</td>
</tr>
<tr>
<td>Smallest Read</td>
<td>Byte</td>
<td>Byte</td>
<td>Byte</td>
<td>Byte</td>
<td>Byte</td>
</tr>
<tr>
<td>Read Speed</td>
<td>Fast</td>
<td>Fast</td>
<td>Fast</td>
<td>Fast</td>
<td>Fast</td>
</tr>
<tr>
<td>Write Speed</td>
<td>Fast</td>
<td>Fast</td>
<td>Fast</td>
<td>Fast</td>
<td>Fast</td>
</tr>
<tr>
<td>Active Power</td>
<td>Low</td>
<td>Med</td>
<td>Low</td>
<td>High</td>
<td>High?</td>
</tr>
<tr>
<td>Sleep Power</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Price/GB</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High?</td>
</tr>
<tr>
<td>Applications</td>
<td>Niche</td>
<td>TBD</td>
<td>Low Power</td>
<td>Obsolete</td>
<td>Main Memory</td>
</tr>
</tbody>
</table>
Solution 2: Bring Compute to Storage

- In Situ Processing: Kinetic, WDlabs Microserver, NGD, NVXL, ScaleFlux
- Compute in Memory: TOMI, Automata
  - Why it’s failed
- Old failed storage/processing models: Schooner, Virident, Violin
Compute In SSD: *In Situ* Processing

![MySQL Search Performance 8 SSDs](image)

- **Scale In Server**
  - 1% CPU

- **Standard Server**
  - 100% CPU

- **Relative Queries per Second**

- **Number of Drives**

- **Source**: Micron Tented M500 SSD

- **Flash Memory Summit 2013**

- **Legend**:
  - 39% Less Power
  - 2X Performance

- **Notes**:
  - 10 concurrent search records, 40Byte Record
  - Server running MySAM; Scale In Server running MySAM
Compute In Memory

TOMI Borealis Block Diagram

Automata
Compute In Memory

Example: Network Analysis
Compute Appliances

Schooner

MySQL

memcached

Virident

Violin Memory with Microsoft Windows Server
All Require Support Throughout The System

- Software support
- Firmware (BIOS) support
- CPU support
- Even new pins on the bus!

Example: Let’s look at Persistent Memory (PM)
PM Needs New Software

- Move away from “Storage vs. Memory” approach
  - Store at the byte level, not blocks
  - Avoid the storage stack
  - Avoid things like flash translation
The SNIA Persistent Memory Programming Model

- [https://www.snia.org/PM](https://www.snia.org/PM)
PM Needs BIOS Support

Current BIOS
- DRAM corrupt at boot
- Power fail loses everything:
  - Memory
  - Registers
  - Cache

PM BIOS
- DRAM may boot with valid data
- Push DRAM to PM at power fail:
  - Memory
  - Registers
  - Cache
PM Needs Processor Support

- Intel’s new IA instructions
  - Cache & write buffer flush
  - NVDIMM-N “Commit”: Copy DRAM to NAND
- Context switch issue?
Context switches become the issue

Latency (10ns = 1μs = 1ms = 10ms = 100ms = 100μs)

- **HDD**
- **SATA SSD**
- **NVMe Flash**
- **Persistent Memory**

- **Context Switch Zone (Interrupts are A-OK)**
- **Nether Zone**
- **Polling Zone**
Advent of SSDs Was Similar

- New software required
  - Caching/tiering
    - Blossomed into Software-Defined Storage
  - Faster I/O stacks
- New hardware
  - SATA 2 & 3
    - SAS kept pace
  - NVMe for PCIe SSDs
Other Alternatives Will Need Similar Efforts

- In-situ processing:
  - Server vs. drive-based software
  - What to do about HA?
- How does this work with compute and memory virtualization?
Summary

- Change is in the air
  - Persistence everywhere
  - New ways to bring processing to storage
  - New approaches to computing
- A lot of support will be required
  - Software
  - Architecture
  - Hardware
Questions?
Your Presenters

Tom Coughlin, President, Coughlin Associates is a highly-respected storage analyst and consultant with over 30 years in the data storage industry in engineering and management at high profile companies.

Jim Handy is a widely recognized semiconductor analyst, has over 30 years in the electronics industry. His background includes marketing and design positions at market-leading suppliers.
Source Material

- **Tom Conte, Georgia Tech**, The IEEE Rebooting Computing Initiative: Rethinking All Levels of How We Compute, Presentation at Sandia, July 2017
- **SNIA Persistent Memory Programming Model**, [https://www.snia.org/PM](https://www.snia.org/PM)
- **National Council on Aging**, [https://www.ncoa.org](https://www.ncoa.org)