All Tomorrow’s Memory Systems
(ver. 2018)

Bruce Jacob (& Don Yeung)

Keystone Professor
University of Maryland
Some Background: Wish List

- Fine-Grained Access
- Bandwidth
- Capacity
- Low Power
- Nonvolatility

* Things we did and/or are doing now (I’ll cover in talk)

- DRAM - HBM/HMC*
- Flash, 3DXP, RRAM, PCM, etc - NVMM*
- HBNV*
Some Background: Wish List

- Fine-Grained Access
- Bandwidth
- Capacity
- Low Power
- Nonvolatility

Major implications for OS* & apps

* Things we did and/or are doing now (I’ll cover in talk)
Hybrid Memory Cube

Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is 160 GB/s per (640 total)

Total concurrency = 16 x 8 x 2..8 (256–1024)
Hybrid Memory Cube

Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is 160 GB/s per (640 total)

Total conc'y = 16 x 8 x 2..8 (256–1024)
Hybrid Memory Cube

Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is 160 GB/s per (640 total)

Total conc’y = 16 x 8 x 2..8 (256–1024)
Hybrid Memory Cube

Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is 160 GB/s per (640 total)

Total conc’y = 16 x 8 x 2..8 (256–1024)
Hybrid Memory Cube

Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is 160 GB/s per (640 total)

Total conc’y = $16 \times 8 \times 2..8$ (256–1024)
Hybrid Memory Cube

Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is 160 GB/s per (640 total)

Total conc’y = 16 x 8 x 2..8 (256–1024)
Hybrid Memory Cube

Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is 160 GB/s per (640 total)

Total conc’y = 16 x 8 x 2..8 (256–1024)
Hybrid Memory Cube

Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is 160 GB/s per (640 total)

Total conc’y = 16 x 8 x 2..8 (256–1024)
Hybrid Memory Cube

Off-chip: high speed SerDes and generic protocol

4 I/O Ports, up to 80 GB/s each

Next gen is 160 GB/s per (640 total)

Total concurrency = 16 x 8 x 2..8 (256–1024)
HMC Die

1Gb Partition, with internal banks

Source: Micron
Logic Die
Logic Die

Vault Controller
Logic Die

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller
Logic Die

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

Vault Controller

IO Port

IO Port

IO Port

IO Port
Logic Die

IO Port | IO Port | IO Port | IO Port
---|---|---|---
Vault Controller | Vault Controller | Vault Controller | Vault Controller
Vault Controller | Vault Controller | Vault Controller | Vault Controller
Vault Controller | Vault Controller | Vault Controller | Vault Controller
Vault Controller | Vault Controller | Vault Controller | Vault Controller
Vault Controller | Vault Controller | Vault Controller | Vault Controller
Vault Controller | Vault Controller | Vault Controller | Vault Controller
Vault Controller | Vault Controller | Vault Controller | Vault Controller
Vault Controller | Vault Controller | Vault Controller | Vault Controller

4x16 Crossbar Switch
HMC Performance

Execution can be several times faster than DDR3-1600

All Tomorrow’s Memory Systems

Bruce Jacob

University of Maryland

Slide 7

Hybrid Memory Cube

CPU

4-Link Cube

CPU

4-Link Cube

CPU

4-Link Cube

CPU

4-Link Cube

CPU

4-Link Cube

CPU

4-Link Cube

CPU

4-Link Cube

CPU

4-Link Cube

CPU

4-Link Cube

CPU
High Bandwidth Memory

Uses a simple ‘2.5D’ instead of full 3D stacking

- TSV Stack Up to 4 or 8 DRAM dies
- 1024-bit 8-Channel Wide Interface
- HBM DRAMs
- HBM Interface
- 1024-bit x 1Gtpfs = 128GB/sec
- TSV Interposer
High Bandwidth Memory

Uses a simple ‘2.5D’ instead of full 3D stacking

- **HBM DRAMs**
- **HBM Interface**
- **1024-bit x 1Gbps = 128GB/sec**
- **1024-bit 8-Channel Wide Interface**
- **TSV Stack Up to 4 or 8 DRAM dies**
- **TSV Interposer**
High Bandwidth Memory

Uses a simple ‘2.5D’ instead of full 3D stacking

- TSV Stack Up to 4 or 8 DRAM dies
- 1024-bit 8-Channel Wide Interface
- HBM DRAMs
- HBM Interface
- 1024-bit x 1Gtps = 128GB/sec
- TSV Interposer
High Bandwidth Memory

Uses a simple ‘2.5D’ instead of full 3D stacking

- TSV Stack
  Up to 4 or 8 DRAM dies

- 1024-bit 8-Channel Wide Interface

- HBM DRAMs

- HBM Interface

- 1024-bit x 1Gbps = 128GB/sec

- TSV Interposer
High Bandwidth Memory

Uses a simple ‘2.5D’ instead of full 3D stacking

- TSV Stack
  - Up to 4 or 8 DRAM dies

- HBM DRAMs

- 1024-bit x 1Gtps = 128GB/sec

- 1024-bit
  - 8-Channel Wide Interface

- HBM Interface

- TSV Interposer

- GPU/CPU

- All Tomorrow’s Memory Systems
  - Bruce Jacob
  - University of Maryland
High Bandwidth Memory

Uses a simple ‘2.5D’ instead of full 3D stacking

1024-bit x 1Gtps = 128GB/sec

TSV Interposer

HBM DRAMs

HBM Interface

1024-bit 8-Channel Wide Interface

TSV Stack
Up to 4 or 8 DRAM dies

GPU/CPU

1024-bit 8-Channel Wide Interface

m

m
High Bandwidth Memory

Each Link is 128 Bits Wide: \textit{1024 Total}
### Non-Volatile Main Memory

<table>
<thead>
<tr>
<th>Technology</th>
<th>Cost for 10 GB</th>
<th>Size of 10 GB</th>
<th>Power for 10 GB</th>
<th>Power per GB/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off-Chip SRAM</td>
<td>$1,000</td>
<td>1 bucket</td>
<td>0.1–1 W</td>
<td>0.1 W</td>
</tr>
<tr>
<td>DDR4 SDRAM</td>
<td>$100</td>
<td>1 DIMM</td>
<td>1 W</td>
<td>0.1 W</td>
</tr>
<tr>
<td>NAND Flash</td>
<td>$10</td>
<td>&lt;1 chip</td>
<td>0</td>
<td>0.1 W (?)</td>
</tr>
<tr>
<td>3D XPoint</td>
<td>$40</td>
<td>&lt;1 chip</td>
<td>0</td>
<td>0.1 W (?)</td>
</tr>
</tbody>
</table>

*Note:* wear-out mitigated by using MANY devices (thousands). A single device would wear out in under two days; therefore, 1000 devices should last for at least a year.

Next, you can trade off longevity for access time and wearout: if the data need only last hours or minutes, wearout is reduced.
A Comparative Example

- **SSD**
  - 1 TB NAND Flash PCIe SSD (I/O)
  - 32 GB DDRx SDRAM
  - 8MB LLC SRAM
  - 1 TB NAND Flash PCIe SSD (I/O)
  - 32 GB DDRx SDRAM
  - 8MB LLC SRAM
  - 1 TB NAND Flash PCIe SSD (I/O)
  - 32 GB DDRx SDRAM
  - 8MB LLC SRAM
  - 1 TB NAND Flash PCIe SSD (I/O)
  - 32 GB DDRx SDRAM
  - 8MB LLC SRAM
  - Cost: $500 – 10W

- **NVMM**
  - 1 TB NAND Flash Main Memory
  - 32 GB SDRAM
  - 8MB SRAM
  - 1 TB NAND Flash Main Memory
  - 32 GB SDRAM
  - 8MB SRAM
  - 1 TB NAND Flash Main Memory
  - 32 GB SDRAM
  - 8MB SRAM
  - Cost: $500 – 10s of W

- **Ideal**
  - 1 TB DDRx SDRAM Main Memory
  - 32 GB SDRAM
  - 8MB SRAM
  - 1 TB DDRx SDRAM Main Memory
  - 32 GB SDRAM
  - 8MB SRAM
  - 1 TB DDRx SDRAM Main Memory
  - 32 GB SDRAM
  - 8MB SRAM
  - Cost: $10,000 – 100W
All Tomorrow’s Memory Systems
Bruce Jacob
University of Maryland

SLIDE 12

**NVMM Performance**

"Ideal"

Normalized Performance

<table>
<thead>
<tr>
<th></th>
<th>GUPS</th>
<th>DD Read</th>
<th>DD Write</th>
<th>Mmap</th>
<th>Pnum</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSD - SLC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NVMM - SLC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSD - MLC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NVMM - MLC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This is when we realized how good Linux is at prefetching out of SSDs
High Bandwidth Non Volatiles

The problem: You want 1TB @ 320 GB/s

<table>
<thead>
<tr>
<th>Pure DRAM</th>
<th>Pure NAND Flash</th>
</tr>
</thead>
<tbody>
<tr>
<td>64 HMCs</td>
<td>400 ONFI-4 flash chips*</td>
</tr>
<tr>
<td>1TB</td>
<td>300 TB</td>
</tr>
<tr>
<td>20,000 GB/s</td>
<td>320 GB/s*</td>
</tr>
<tr>
<td>100 W static power</td>
<td>0 W static power</td>
</tr>
<tr>
<td>128-byte granularity</td>
<td>16,000-byte granularity</td>
</tr>
</tbody>
</table>

* on a 3200-pin parallel bus
High Bandwidth Non Volatiles

Borrow page from HMC playbook

Network Fabric

MC MC MC MC MC MC...

NV RRAM: up to 1000ns expected*

*trade-offs?
High Bandwidth Non Volatiles

First-order concurrency requirements:

<table>
<thead>
<tr>
<th>bytes</th>
<th>sec</th>
<th>access</th>
<th>byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>320 GB</td>
<td>1000 ns</td>
<td>access</td>
<td>32 B</td>
</tr>
<tr>
<td>320 GB</td>
<td>1000 ns</td>
<td>access</td>
<td>256 B</td>
</tr>
<tr>
<td>160 GB</td>
<td>500 ns</td>
<td>access</td>
<td>128 B</td>
</tr>
</tbody>
</table>

320 GB/sec = 10K

320 GB/sec = 1250

160 GB/sec = 625
So what all does this enable?

**HBM/HMC**: hugely parallel systems (the duality of bandwidth and parallelism), streaming applications, 2x performance

**NVMM**: massive data sets, new OS paradigms such as merged VM+FS and journaled main memory (built-in checkpoint/restart)

**HBNV**: fine-grained operations on enormous sparse data sets
So what all does this enable?

**HBM/HMC**: hugely parallel systems (the duality of bandwidth and parallelism), streaming applications, 2x performance

**NVMM**: massive data sets, new OS paradigms such as merged VM+FS and journaled main memory (built-in checkpoint/restart)

**HBNV**: fine-grained operations on enormous sparse data sets

For example …
**Application Area: SpGEMM**

**a.** Segmented Graph Database

**b.** SpGEMM of NxN matrices (row-row version)

A: Left Input

B: Right Input

C: Output

```
\[ \begin{pmatrix}
    i \\
    u \\
    v \\
    w \\
    j \\
    l \\
    k \\
    m
\end{pmatrix} \times
\begin{pmatrix}
    \cdot \\
    \cdot \\
    \cdot \\
    \cdot \\
    \cdot \\
    \cdot \\
    \cdot \\
    \cdot
\end{pmatrix}
= \\
\begin{pmatrix}
    u \\
    v \\
    w \\
    j \\
    l \\
    k \\
    m
\end{pmatrix}
```

Compressed Sparse Row (CSR) data format:

- **rows**
- **cols**
- **data**

**Memory Requirements:**

- TB/s bandwidth
- TB-scale capacity
- Fine-grained access
- Low power
Connected Components

Goal: label all nodes in connected component w unique ID

Awerbuch-Shiloach algorithm \( O((n+m)\log(n)) \)

Original Graph  Ordered Edges  Shortcuts  Hooking  Shortcuts

Vector w AVX-512, 8-way masked scatter-gather: inner loop 20 instrs w 7.4 traversed edges per iteration per thread

**DRAM Memory System**
(320 GB/s, 55ns latency, 128-byte blocks)

Achieves 2.5 GTEPS (~100W)

**ReRAM Memory System**
(16K-way mem parallelism, 700ns latency, 8-byte blocks)

Achieves 23.4 GTEPS (10–100W)
Nonvolatility Issues
Unified VM+FS Subsystems (OS redesign)

- By default, data in process address space temporary, garbage-collected at exit(); 
  permanentify function to keep around

- Possible directions:
  - Persistent objects (e.g. Mneme, POMS) [failed only due to reliance on disk]
  - Named regions
- Journaled main memory w/ checkpointing
Capacity Issues

Rethink Protection & Translation

- TLB overhead is ~20%
  - So get rid of it already!
  - BUT: need protection, authentication

- Why not waste bits? Simplify both sharing and translation by eliminating much of VM

- OS/HW co-design needed: e.g., sharing via vaddr instead of paddr, language support? Might make MPI less painful?
Recap Regarding Software

**Bandwidth** gives you $2x$ right off the bat

NVMM: $5x$ performance hit vs DRAM for a $100–1000x$ increase in capacity

- $10–100$ TB main memory for $1$-U server
  (really large data sets become realistic)

**Nonvolatility** opens up many questions:

- Redesign VM+FS subsystems
- Journaled main memory (e.g. thru flash)
- Persistent objects (Mneme, POMS, etc)
Shameless
Plug

www.memsys.io

Organizers
Bruce Jacob, U. Maryland
Kathy Smiley, Memory Systems
Rajaj Agarwal, Intel
Amr el-Abas, Micron
Jiann Tang, Sandia National Labs
Bruce Chokare, U. Pittsburgh
Zhiyin Chao, Intel
Bravo Christenson, Intel
Qing Qian, U. Rochester
David Donofrio, Berkeley Lab
Wenda Elasser, AIML
Maya Gokhale, LLNL
Keshav Gu, Lehigh U.
Michael Ignatowski, AMD
Matthias Jung, U. Kaiserslautern
Karin Katz, Georgia Tech
Scott Lloyd, LLNL
Tally A. McKeel, Chalmers/Rambus
Ehsan Meidani, Georgia Tech
David Reinsink, Sandia National Labs
Anur Rodrigues, Sandia National Labs
Robert Voglet, Northrop Grumman
Vincent Weter, U. Maine
Christian Weiss, U. Kaiserslautern
Kriitn Kuethe, Intel
Sudhakar Yalamanchili, Georgia Tech
Kun Zhang, Chinese Asial of Sciences
Jiaran Zhao, UC Santa Cruz

Important Dates
Submission: 1 May*, 2017
Notification: 16 June, 2017

Washington DC
Sep/Oct 2018

Call For Papers

MEMSYS 2017
The International Symposium on Memory Systems October 2–5, Washington D.C.

Memory-device manufacturing, memory-architecture design, and the use of memory technologies by application software all profoundly impact today’s and tomorrow’s computing systems, in terms of their performance, function, reliability, predictability, power dissipation, and cost. Existing memory technologies are seen as limiting in terms of power, capacity, and performance; and design-related limitations to answer the requirements of applications. Our goal is to bring together researchers, architects, and designers in this exciting and rapidly evolving field to talk to each other on the latest state of the art, to exchange ideas, reframe challenges. Visit memsys.io for more information.

Call for Papers

Submission extension of one week

Camera Ready Notification: 1 August, 2017
Submission: 16 June, 2017

Important Dates

Call For Papers

Submission: 1 May*, 2017
Notification: 16 June, 2017

There will be an automatic
Expiration of extended abstracts

Research Papers

Abstracts

Position Papers

Areas of Interest

Processor cache design, prefetching, data prediction, etc.

Algorithmic & software memory management techniques

Interconnects to support large-scale data movement

Memory failure modes and mitigation strategies

Operating system design for hybrid/nonvolatile memories

Memory and system security issues

Memory-centric programming models, languages, optimization

Operating system design

Multithreading, multicore, and NUMA architectures

Emerging memory technologies, their controllers, and novel uses

Interference at the memory level across datacenter applications

Memory and system security issues

Issues in the design and operation of large memory machines

Memory and system security issues

Meet researchers, architects, and designers of memory for more information.

Submit papers and presentations

To reiterate, papers that focus on topics outside of traditional conference scopes, will be extended abstracts, position papers, and/or full papers. Pre-submission of extended abstracts, position papers, and/or full research papers is encouraged. The extended abstract submission is given a 20-minute presentation time slot. All accepted papers will be published in the ACM Digital Library.

Submission Formats

Conference paper layout, using ACM's proceedings template (if you choose to sign your non-ACM papers). The general structure of the paper should be 10+ page extended abstract, 5-6 page research paper, and 1-2 page position paper. All papers must be anonymous for the initial submission.

Submission F ormat

Camera Ready Notification: 1 August, 2017
Submission: 16 June, 2017

There will be an automatic
Expiration of extended abstracts

Research Papers

Abstracts

Position Papers

Areas of Interest

Processor cache design, prefetching, data prediction, etc.

Algorithmic & software memory management techniques

Interconnects to support large-scale data movement

Memory failure modes and mitigation strategies

Operating system design for hybrid/nonvolatile memories

Memory and system security issues

Memory-centric programming models, languages, optimization

Operating system design

Multithreading, multicore, and NUMA architectures

Emerging memory technologies, their controllers, and novel uses

Interference at the memory level across datacenter applications

Memory and system security issues

Issues in the design and operation of large memory machines

Memory and system security issues

Meet researchers, architects, and designers of memory for more information.

Submit papers and presentations

To reiterate, papers that focus on topics outside of traditional conference scopes, will be extended abstracts, position papers, and/or full papers. Pre-submission of extended abstracts, position papers, and/or full research papers is encouraged. The extended abstract submission is given a 20-minute presentation time slot. All accepted papers will be published in the ACM Digital Library.
Thank You!

Bruce Jacob
blj@umd.edu
www.ece.umd.edu/~blj