Performance Portability for Extreme Scale High Performance Computing

Jeffrey S. Vetter
and many collaborators

Presented to
Workshop on Clusters, Clouds, and Data for Scientific Computing (CCDSC 2016)

Châteauform'
La Maison des Contes
427 Chemin de Chanzé, France

5 Oct 2016
Accepted papers include:
- Quantum computing (Dwave)
- Neuromorphic computing
- Probabilistic
- Approximate computing, numerics
- Reconfigurable
- Photonics
- Software
- Performance modeling
Overview

• Recent trends in extreme-scale HPC paint an ambiguous future
  – Contemporary systems provide evidence that power constraints are driving architectures to change rapidly
  – Multiple architectural dimensions are being (dramatically) redesigned: Processors, node design, memory systems, I/O
  – Complexity is our main challenge

• Applications and software systems are all reaching a state of crisis
  – Applications will not be functionally or performance portable across architectures
  – Programming and operating systems need major redesign to address these architectural changes
  – Procurements, acceptance testing, and operations of today’s new platforms depend on performance prediction and benchmarking.

• We need performance portable programming models now more than ever!

• Programming systems must provide performance portability (in addition to functional portability)!!
  – New memory hierarchies with NVM everywhere
  – Heterogeneous systems
Trends toward Exascale
Exascale architecture targets circa 2009
2009 Exascale Challenges Workshop in San Diego

Attendees envisioned two possible architectural swim lanes:
1. Homogeneous many-core thin-node system
2. Heterogeneous (accelerator + CPU) fat-node system

<table>
<thead>
<tr>
<th>System attributes</th>
<th>2009</th>
<th>“Pre-Exascale”</th>
<th>“Exascale”</th>
</tr>
</thead>
<tbody>
<tr>
<td>System peak</td>
<td>2 PF</td>
<td>100-200 PF/s</td>
<td>1 Exaflop/s</td>
</tr>
<tr>
<td>Power</td>
<td>6 MW</td>
<td>15 MW</td>
<td>20 MW</td>
</tr>
<tr>
<td>System memory</td>
<td>0.3 PB</td>
<td>5 PB</td>
<td>32–64 PB</td>
</tr>
<tr>
<td>Storage</td>
<td>15 PB</td>
<td>150 PB</td>
<td>500 PB</td>
</tr>
<tr>
<td>Node performance</td>
<td>125 GF</td>
<td>0.5 TF</td>
<td>7 TF</td>
</tr>
<tr>
<td>Node memory BW</td>
<td>25 GB/s</td>
<td>0.1 TB/s</td>
<td>1 TB/s</td>
</tr>
<tr>
<td>Node concurrency</td>
<td>12</td>
<td>O(100)</td>
<td>O(1,000)</td>
</tr>
<tr>
<td>System size (nodes)</td>
<td>18,700</td>
<td>500,000</td>
<td>50,000</td>
</tr>
<tr>
<td>Node interconnect BW</td>
<td>1.5 GB/s</td>
<td>150 GB/s</td>
<td>1 TB/s</td>
</tr>
<tr>
<td>IO Bandwidth</td>
<td>0.2 TB/s</td>
<td>10 TB/s</td>
<td>30-60 TB/s</td>
</tr>
<tr>
<td>MTTI</td>
<td>day</td>
<td>O(1 day)</td>
<td>O(0.1 day)</td>
</tr>
</tbody>
</table>
# Contemporary ASCR Computing At a Glance

<table>
<thead>
<tr>
<th>System attributes</th>
<th>NERSC Now</th>
<th>OLCF Now</th>
<th>ALCF Now</th>
<th>NERSC Upgrade</th>
<th>OLCF Upgrade</th>
<th>ALCF Upgrades</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name Planned Installation</td>
<td>Edison</td>
<td>TITAN</td>
<td>MIRA</td>
<td>Cori 2016</td>
<td>Summit 2017-2018</td>
<td>Theta 2016</td>
</tr>
<tr>
<td>System peak (PF)</td>
<td>2.6</td>
<td>27</td>
<td>10</td>
<td>&gt; 30</td>
<td>200</td>
<td>&gt; 8.5</td>
</tr>
<tr>
<td>Peak Power (MW)</td>
<td>2</td>
<td>9</td>
<td>4.8</td>
<td>&lt; 3.7</td>
<td>13.3</td>
<td>1.7</td>
</tr>
<tr>
<td>Total system memory</td>
<td>357 TB</td>
<td>710TB</td>
<td>768TB</td>
<td>~1 PB DDR4 + High Bandwidth Memory (HBM) + 1.5PB persistent memory</td>
<td>&gt;2.4 PB DDR4 + HBM + 3.7 PB persistent memory</td>
<td>&gt;480 TB DDR4 + High Bandwidth Memory (HBM)</td>
</tr>
<tr>
<td>Node performance (TF)</td>
<td>0.460</td>
<td>1.452</td>
<td>0.204</td>
<td>&gt; 3</td>
<td>&gt; 40</td>
<td>&gt; 3</td>
</tr>
<tr>
<td>Node processors</td>
<td>Intel Ivy Bridge</td>
<td>AMD Opteron Nvidia Kepler</td>
<td>64-bit PowerPC A2</td>
<td>Intel Knights Landing many core CPUs &amp; Intel Haswell CPU in data partition</td>
<td>Multiple IBM Power9 CPUs &amp; multiple Nvidia Volta GPUs</td>
<td>Intel Knights Landing Xeon Phi many core CPUs</td>
</tr>
<tr>
<td>System size (nodes)</td>
<td>5,600 nodes</td>
<td>18,688 nodes</td>
<td>49,152</td>
<td>9,300 nodes</td>
<td>~4,600 nodes</td>
<td>2,500 nodes</td>
</tr>
<tr>
<td>System Interconnect</td>
<td>Aries</td>
<td>Gemini</td>
<td>5D Torus</td>
<td>Aries</td>
<td>Dual Rail EDR-IB</td>
<td>Aries</td>
</tr>
<tr>
<td>File System</td>
<td>7.6 PB 168 GB/s, Lustre®</td>
<td>32 PB 1 TB/s, Lustre®</td>
<td>26 PB 300 GB/s GPFS™</td>
<td>28 PB 744 GB/s Lustre®</td>
<td>120 PB 1 TB/s GPFS™</td>
<td>10PB, 210 GB/s Lustre initial</td>
</tr>
</tbody>
</table>

Binkley, ASCAC, April 2016

---

Complexity $\alpha T$
Complexity is the next major challenge!

- Time of rapid change in computer architectures
  - Heterogeneous cores
  - Deep, multimode memory systems
  - I/O architectures
  - Reliability
  - Changing system balance

- Uncertainty, Ambiguity among current and future architectures
  - Managing complexity is our main challenge!
    - Complex systems $\rightarrow$ Fewer apps $\rightarrow$ Smaller HPC

- Critical questions
  - How do we design future systems so that they are faster than current systems on mission applications?
    - Entirely possible that the new system will be slower than the old system!
  - How do we design applications for some level of performance portability?
Performance Portability: what is it?

- Effectively from application perspective, “write once, run anywhere efficiently”

- Performance portability is not a new topic
  - Kuck, 1996

- For two decades, expectations were set by ‘(Curse of) Moore’s Law’ with exception for MPI for scaling parallelism
  - Recompile and relink

- More important then ever
  - Becoming difficult to hide complexity for even functional portability

- Efficiently use resource of interest

Motivating Heterogeneous Systems
Specialization is here to stay: Core, Processor Architectures

- **Cores**
  - CPU
  - GPUs (discrete, integrated)
  - FPGAs
  - Special purpose engines
    - RNGs
    - AES, video engines
    - Transactional memory
    - Virtualization support
- **SIMD/short vector**
- **SMT, threading models**
- **DVFS (incl Turboboost)**
- etc

---

D.E. Shaw, M.M. Deneroff, R.O. Dror et al., "Anton, a special-purpose machine for molecular dynamics"
Nvidia and IBM create GPU interconnect for faster supercomputing

“InfiniBand” shrugs up to 80GB of data per second between CPUs and GPUs.

by Joe Brandon - Mar 20, 2014, 4:47am CST

It Begins: AMD Announces Its First ARM Based Server SoC, 64-bit/8-core Opteron A100

by Round Lal Shing - January 19, 2014, 12:15AM EST

“Seattle” 64-bit ARM Server Processor

First 64-bit ARM Server CPU to Sample in March

by Richard Kao - June 6, 2014, 1:38am CST

Nvidia Jetson TK1 mini supercomputer is up for pre-order

Will ship on 11 May

by Lee Bell - June 6, 2014, 11:20am CST

Nvidia’s Jetson TK1 mini supercomputer development kit is now up for pre-order. priced at $992.

by Richard Kao - June 6, 2014, 11:46am CST

Intel’s 14nm Broadwell GPU takes shape, indicates major improvements over Haswell

by Sebastian Ardanza - November 5, 2013 at 10:17 pm

Avago Agrees to Buy Broadcom for $37 Billion

by Michael A. de la Osa — and Chad Moon — November 30, 2015

Microsoft Extends FPGA Reach from Bing to Deep Learning

August 27, 2015 Timothy Prickett Morgan

After three years of research into how it might accelerate its Bing search engine using field programmable gate arrays (FPGAs), Microsoft came up with a scheme that would let it lash Stratix V devices from Altera to the two socket server nodes in the minimalist Open Cloud Servers that it has designed expressly for its hyperscale datacenters. These CPU-FPGA hybrids were rolled out into production earlier this year to accelerate Bing page rank functions, and Microsoft started hunting around for other workloads with which to juice with FPGAs, and...

Intel Mates FPGA With Future Xeon Server Chip

June 15, 2014 by Timothy Prickett Morgan

Intel is taking AMD programmable gate arrays seriously as a means of accelerating applications and has crafted a hybrid chip that marriage an FPGA to a Xeon E5 processor and puts them in the same processor socket.

In the news
## Current ASCR Computing At a Glance

<table>
<thead>
<tr>
<th>System attributes</th>
<th>NERSC Now</th>
<th>OLCF Now</th>
<th>ALCF Now</th>
<th>NERSC Upgrade</th>
<th>OLCF Upgrade</th>
<th>ALCF Upgrades</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planned Installation</td>
<td>Edison</td>
<td>TITAN</td>
<td>MIRa</td>
<td>Cori 2016</td>
<td>Summit 2017-2018</td>
<td>Theta 2016</td>
</tr>
<tr>
<td>System peak (PF)</td>
<td>2.6</td>
<td>27</td>
<td>10</td>
<td>&gt; 30</td>
<td>150</td>
<td>&gt;8.5</td>
</tr>
<tr>
<td>Peak Power (MW)</td>
<td>2</td>
<td>9</td>
<td>4.8</td>
<td>&lt; 3.7</td>
<td>10</td>
<td>1.7</td>
</tr>
<tr>
<td>Total system memory</td>
<td>357 TB</td>
<td>710TB</td>
<td>768TB</td>
<td>~1 PB DDR4 + High Bandwidth Memory (HBM) + 1.5PB persistent memory</td>
<td>&gt; 1.74 PB DDR4 + HBM + 2.8 PB persistent memory</td>
<td>&gt;480 TB DDR4 + High Bandwidth Memory (HBM)</td>
</tr>
<tr>
<td>Node performance (TF)</td>
<td>0.460</td>
<td>1.452</td>
<td>0.204</td>
<td>&gt; 3</td>
<td>&gt; 40</td>
<td>&gt; 3</td>
</tr>
<tr>
<td>Node processors</td>
<td>Intel Ivy Bridge</td>
<td>AMD Opteron Nvida Kepler</td>
<td>64-bit PowerPC A2</td>
<td>Intel Knights Landing many core CPUs &amp; Intel Haswell CPUs in data partition</td>
<td>Multiple IBM Power9 CPUs &amp; multiple Nvidia Voltaas GPUS</td>
<td>Intel Knights Landing Xeon Phi many core CPUs</td>
</tr>
<tr>
<td>System size (nodes)</td>
<td>5,600 nodes</td>
<td>18,688 nodes</td>
<td>49,152</td>
<td>9,300 nodes, 1,900 nodes in data partition</td>
<td>~3,500 nodes</td>
<td>&gt;2,500 nodes</td>
</tr>
<tr>
<td>System Interconnect</td>
<td>Aries</td>
<td>Gemini</td>
<td>5D Torus</td>
<td>Aries</td>
<td>Dual Rail EDR-IB</td>
<td>Aries</td>
</tr>
<tr>
<td>File System</td>
<td>7.6 PB 168 GB/s, Lustre®</td>
<td>32 PB 1 TB/s, Lustre®</td>
<td>26 PB 300 GB/s GPFS™</td>
<td>28 PB 744 GB/s Lustre®</td>
<td>120 PB 1 TB/s GPFS™</td>
<td>10PB, 210 GB/s Lustre initial</td>
</tr>
</tbody>
</table>

Binkley, ASCAC, April 2016
Programming Heterogeneous Systems
Yields Complex Programming Models

- This approach is not scalable, affordable, robust, elegant, etc.
- Not performance portable
OpenARC: Open Accelerator Research Compiler

- Open-Sourced, High-Level Intermediate Representation (HIR)-Based, Extensible Compiler Framework.
  - Perform source-to-source translation from OpenACC C to target accelerator models.
    - Support full features of OpenACC V1.0 (+ array reductions and function calls)
    - Support both CUDA and OpenCL as target accelerator models
  - Provide common runtime APIs for various back-ends
  - Can be used as a research framework for various study on directive-based accelerator computing.
    - Built on top of Cetus compiler framework, equipped with various advanced analysis/transformation passes and built-in tuning tools.
    - OpenARC’s IR provides an AST-like syntactic view of the source program, easy to understand, access, and transform the input program.

Understanding Performance Portability of High-level Programming Models for Heterogeneous Systems

• Problem
  - Directive-based, high-level accelerator programming models such as OpenACC provide code portability.
    • How does it fare on performance portability?
    • And what architectural features/compiler optimizations affect the performance portability? And how much?

• Solution
  - Proposed a high-level, architecture-independent intermediate language (HeteroIR) to map high-level programming models (e.g., OpenACC) to diverse heterogeneous devices while maintaining portability.
  - Using HeteroIR, port and measure the performance portability of various OpenACC applications on diverse architectures.

• Results
  - Using HeteroIR, OpenARC ported 12 OpenACC applications to diverse architectures (NVIDIA CUDA, AMD GCN, and Intel MIC), and measured the performance portability achieved across all applications.
  - HeteroIR abstracts out the common architecture functionalities, which makes it easy for OpenARC (and other compilers) to support diverse heterogeneous architectures.
  - HeteroIR, combined with rich OpenARC directives and built-in tuning tools, allows OpenARC to be used for various tuning studies on diverse architectures.

Intelligent selection of optimizations based on target architecture

Figure 5: Memory Coalescing Benefits on Different Architectures: MIC is impacted the least by the non-coalesced accesses

Figure 7: Impact of Tiling Transformation: MATMUL shows higher benefits than JACOBI owing to more contiguous accesses

Figure 9: Effects of Loop Unrolling - MIC shows benefits on unrolling

Fig. 11: Comparison of hand-written CUDA/OpenCL programs against auto-tuned OpenARC code versions: Tuned OpenACC programs perform reasonably well against hand-written codes
OpenACC to FPGA: A Framework for Directive-Based High-Performance Reconfigurable Computing

- OpenACC-to-FPGA translation framework
  - source-to-source translation of the input OpenACC program into an output OpenCL code,
  - further compiled to an FPGA program by the underlying backend Altera OpenCL compiler.
  - Prototyped new OpenACC directives to support pipelining of kernels

- Recent Results
  - Proposed several FPGA-specific OpenACC compiler optimizations and pragma extensions to achieve higher throughput.
  - Evaluated the framework using eight OpenACC benchmarks, and measured performance variations on diverse architectures (Altera FPGA, NVIDIA/AMD GPUs, and Intel Xeon Phi).

Emerging Non-volatile Memory Systems
Exascale architecture targets circa 2009

2009 Exascale Challenges Workshop in San Diego

Attendees envisioned two possible architectural swim lanes:
1. Homogeneous many-core thin-node system
2. Heterogeneous (accelerator + CPU) fat-node system

<table>
<thead>
<tr>
<th>System attributes</th>
<th>2009</th>
<th>“Pre-Exascale”</th>
<th>“Exascale”</th>
</tr>
</thead>
<tbody>
<tr>
<td>System peak</td>
<td>2 PF</td>
<td>100-200 PF/s</td>
<td>1 Exaflop/s</td>
</tr>
<tr>
<td>Power</td>
<td>6 MW</td>
<td>15 MW</td>
<td>20 MW</td>
</tr>
<tr>
<td>System memory</td>
<td>0.3 PB</td>
<td>5 PB</td>
<td>32–64 PB</td>
</tr>
<tr>
<td>Storage</td>
<td>15 PB</td>
<td>150 PB</td>
<td>500 PB</td>
</tr>
<tr>
<td>Node performance</td>
<td>125 GF</td>
<td>0.5 TF</td>
<td>7 TF</td>
</tr>
<tr>
<td>Node memory BW</td>
<td>25 GB/s</td>
<td>0.1 TB/s</td>
<td>1 TB/s</td>
</tr>
<tr>
<td>Node concurrency</td>
<td>12</td>
<td>O(100)</td>
<td>O(1,000)</td>
</tr>
<tr>
<td>System size (nodes)</td>
<td>18,700</td>
<td>500,000</td>
<td>50,000</td>
</tr>
<tr>
<td>Node interconnect BW</td>
<td>1.5 GB/s</td>
<td>150 GB/s</td>
<td>1 TB/s</td>
</tr>
<tr>
<td>IO Bandwidth</td>
<td>0.2 TB/s</td>
<td>10 TB/s</td>
<td>30-60 TB/s</td>
</tr>
<tr>
<td>MTTI</td>
<td>day</td>
<td>O(1 day)</td>
<td>O(0.1 day)</td>
</tr>
</tbody>
</table>
Memory Systems are Diversifying

- HMC, HBM/2/3, LPDDR4, GDDR5X, WIDEIO2, etc
- Configuration diversity
  - Fused, shared memory
  - Scratchpads
  - Write through, write back, etc
  - Consistency and coherence protocols
  - Virtual v. Physical, paging strategies
- 2.5D, 3D Stacking
- New devices (ReRAM, PCRAM, STT-MRAM, Xpoint)

---


NVRAM Technology Continues to Improve – Driven by Market Forces

Kevin Gibb, Product Line Manager, Technisights
4/20/2018 04:10 PM EDT
7 comments

The highly anticipated Samsung’s 48-layer V-NAND flash memory is out in the market, and we at Technisights have the first look.

Samsung announced its 256 Gb 3-bit multi-level cell (MLC) 92AF3V8BDM 3D V-NAND Flash memory earlier this week. April 9, saying that it would be used in both desktop and mobile products. (Figure 1)

SAN JOSE, Calif.—Facebook said it hopes to use its 3D XPoint memories in its data centers. Meanwhile, AMD is working to deliver high-performance computer racks to gain form factors for different cloud applications.

The two moves were likely the highest impact announcements at the annual event of the Facebook-led Open Compute Project (OCP) here. Among other news, Intel showed a new SoC with dual 10G Ethernet controllers and a prototype merging Xeon with an Avant FPGA in a single package.

R. Colin Johnson
7/18/2018 07:10 PM EDT
14 comments

BeSang wants to lower barrier to 3D NAND flash

Samsung Debuts 3D XPoint Killer

Samsung announced plans for what it calls 2-NAND chips that will power SSDs with similar performance but lower costs and rival the 3D XPoint drive. However, it was cautious about the details of the technology that will appear in products sometime next year.

By contrast, a Micron engineer leading its XPoint SSD program was surprisingly candid in an interview with EE Times. She described current prototype using early XPoint chips and an FMP-based controller for the SSDs expected to ship in about a year.

Samsung’s 2-NAND will deliver 10x faster reads than multi-level cell flash and writes that are twice as fast, the company said. At the drive level, they will support both reads and writes at about 20 microseconds, suggesting some of write performance comes from an enhanced controller.

Intel And Micron Jointly Announce Game-Changing 3D XPoint Memory Technology

http://www.eetasia.com/STATIC/ARTICLE_IMAGES/201212/EEOL_2012DEC28_STOR_MFG_NT_01.jpg
## Current ASCR Computing At a Glance

<table>
<thead>
<tr>
<th>System attributes</th>
<th>NERSC Now</th>
<th>OLCF Now</th>
<th>ALCF Now</th>
<th>NERSC Upgrade</th>
<th>OLCF Upgrade</th>
<th>ALCF Upgrades</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planned Installation</td>
<td>Edison</td>
<td>TITAN</td>
<td>MIRA</td>
<td>Cori 2016</td>
<td>Summit 2017-2018</td>
<td>Theta 2016</td>
</tr>
<tr>
<td>System peak (PF)</td>
<td>2.6</td>
<td>27</td>
<td>10</td>
<td>&gt; 30</td>
<td>150</td>
<td>&gt;8.5</td>
</tr>
<tr>
<td>Peak Power (MW)</td>
<td>2</td>
<td>9</td>
<td>4.8</td>
<td>&lt; 3.7</td>
<td>10</td>
<td>1.7</td>
</tr>
<tr>
<td>Total system memory</td>
<td>357 TB</td>
<td>710TB</td>
<td>768TB</td>
<td>~1 PB DDR4 + High Bandwidth Memory (HBM) + 1.5PB persistent memory</td>
<td>&gt; 1.74 PB DDR4 + HBM + 2.8 PB persistent memory</td>
<td>&gt;480 TB DDR4 + High Bandwidth Memory (HBM)</td>
</tr>
<tr>
<td>Node performance (TF)</td>
<td>0.460</td>
<td>1.452</td>
<td>0.204</td>
<td>&gt; 3</td>
<td>&gt; 40</td>
<td>&gt; 3</td>
</tr>
<tr>
<td>Node processors</td>
<td>Intel Ivy Bridge</td>
<td>AMD Opteron Nvidia Kepler</td>
<td>64-bit PowerPC A2</td>
<td>Intel Knights Landing many core CPUs</td>
<td>Intel Haswell CPU in data partition</td>
<td>Multiple IBM Power9 CPUs &amp; multiple Nvidia Voltas GPUs</td>
</tr>
<tr>
<td>System size (nodes)</td>
<td>5,600 nodes</td>
<td>18,688 nodes</td>
<td>49,152</td>
<td>9,300 nodes</td>
<td>1,900 nodes in data partition</td>
<td>~3,500 nodes</td>
</tr>
<tr>
<td>System Interconnect</td>
<td>Aries</td>
<td>Gemini</td>
<td>5D Torus</td>
<td>Aries</td>
<td>Dual Rail EDR-IB</td>
<td>Aries</td>
</tr>
<tr>
<td>File System</td>
<td>7.6 PB 168 GB/s, Lustre®</td>
<td>32 PB 1 TB/s, Lustre®</td>
<td>26 PB 300 GB/s GPFS™</td>
<td>28 PB 744 GB/s Lustre®</td>
<td>120 PB 1 TB/s GPFS™</td>
<td>10PB, 210 GB/s Lustre initial</td>
</tr>
</tbody>
</table>

Binkley, ASCAC, April 2016
## Comparison of Emerging Memory Technologies

<table>
<thead>
<tr>
<th></th>
<th>SRAM</th>
<th>DRAM</th>
<th>eDRAM</th>
<th>2D NAND Flash</th>
<th>3D NAND Flash</th>
<th>PCRAM</th>
<th>STTRAM</th>
<th>2D ReRAM</th>
<th>3D ReRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Retention</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Cell Size (F²)</td>
<td>50-200</td>
<td>4-6</td>
<td>19-26</td>
<td>2-5</td>
<td>&lt;1</td>
<td>4-10</td>
<td>8-40</td>
<td>4</td>
<td>&lt;1</td>
</tr>
<tr>
<td>Minimum F demonstrated (nm)</td>
<td>14</td>
<td>25</td>
<td>22</td>
<td>16</td>
<td>64</td>
<td>20</td>
<td>28</td>
<td>27</td>
<td>24</td>
</tr>
<tr>
<td>Read Time (ns)</td>
<td>&lt;1</td>
<td>30</td>
<td>5</td>
<td>10⁴</td>
<td>10⁴</td>
<td>10-50</td>
<td>3-10</td>
<td>10-50</td>
<td>10-50</td>
</tr>
<tr>
<td>Write Time (ns)</td>
<td>&lt;1</td>
<td>50</td>
<td>5</td>
<td>10³</td>
<td>10⁵</td>
<td>100-300</td>
<td>3-10</td>
<td>10-50</td>
<td>10-50</td>
</tr>
<tr>
<td>Number of Rewrites</td>
<td>10¹⁶</td>
<td>10¹⁶</td>
<td>10¹⁶</td>
<td>10²-10⁵</td>
<td>10²-10⁵</td>
<td>10⁸-10¹⁰</td>
<td>10¹⁵</td>
<td>10⁸-10¹²</td>
<td>10⁸-10¹²</td>
</tr>
<tr>
<td>Read Power</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Write Power</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Power (other than R/W)</td>
<td>Leakage</td>
<td>Refresh</td>
<td>Refresh</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>Sneak</td>
<td>Sneak</td>
</tr>
<tr>
<td>Maturity</td>
<td>Deployed</td>
<td>Experimental</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Intel/Micron Xpoint? Samsung Z-NAND?  

[http://ft.ornl.gov/trac/blackcomb](http://ft.ornl.gov/trac/blackcomb)
Migration up the hierarchy

- Caches
- Main Memory
- I/O Device
- HDD
Programming NVM Systems
NVM Programming Systems: Goals

• Architectures will vary dramatically
  – How should we design the node?
  – Portable across various NVM architectures
  – MPI and OpenMP do not solve this problem.

• Two modes of operation
  – Drop in replacement for DRAM
  – Exploit persistence

• Active area of research

• Performance for HPC scenarios
  – Allow user or compiler/runtime/OS to exploit NVM
  – Asymmetric R/W
  – Remote/Local

• Assume lower power costs under normal usage

• Security

• Correctness and durability
  – A crash or erroneous program could corrupt the NVM data structures
  – Programming system needs to provide support for this model
  – Enhanced ECC for NVM devices

• ACID
  – Atomicity: A transaction is “all or nothing”
  – Consistency: Takes data from one consistent state to another
  – Isolation: Concurrent transactions appears to be one after another
  – Durability: Changes to data will remain across system boots

http://j.mp/nvm-sw-survey

A Survey of Software Techniques for Using Non-Volatile Memories for Storage and Main Memory Systems

Sparsh Mittal, Member, IEEE, and Jeffrey S. Vetter, Senior Member, IEEE

Abstract—Non-volatile memory (NVM) devices, such as Flash, phase change RAM, spin transfer torque RAM, and resistive RAM, offer several advantages and challenges when compared to conventional memory technologies, such as DRAM and magnetic hard disk drives (HDDs). In this paper, we present a survey of software techniques that have been proposed to exploit the advantages and mitigate the disadvantages of NVMs when used for designing memory systems, and, in particular, secondary storage (e.g., solid-state drives and hard disk drives). We classify these software techniques along several dimensions to highlight their similarities and differences. Given the newness of NVMs and growing in popularity, we believe that this survey will indicate further research in the field of software technology for NVMs.

Index Terms—Non-volatile memory (NVM), phase change RAM (PCRAM), spin transfer torque RAM (STT-RAM), resistive RAM (ReRAM), storage class memory (SCM), Solid State Drive (SSD).
NVL-C: Portable Programming for NVMM

- Minimal, familiar, programming interface:
  - Minimal C language extensions.
  - App can still use DRAM.
- Pointer safety:
  - Persistence creates new categories of pointer bugs.
  - Best to enforce pointer safety constraints at compile time rather than run time.
- Transactions:
  - Prevent corruption of persistent memory in case of application or system failure.
- Language extensions enable:
  - Compile-time safety constraints.
  - NVM-related compiler analyses and optimizations.
- LLVM-based:
  - Core of compiler can be reused for other front ends and languages.
  - Can take advantage of LLVM ecosystem.

```c
#include <nvl.h>
struct list {
    int value;
    nvl struct list *next;
};
void remove(int k) {
    nvl_heap_t *heap
        = nvl_open("foo.nvl");
    nvl struct list *a
        = nvl_get_root(heap, struct list);
    #pragma nvl atomic
    while (a->next != NULL) {
        if (a->next->value == k)
            a->next = a->next->next;
        else
            a = a->next;
    }
    nvl_close(heap);
}
```

Evaluation: LULESH

- backup is important for performance
- clobber cannot be applied because old data is needed

- ExM = use SSD as extended DRAM
- T1 = BSR + transactions
- T2 = T1 + backup clauses
- T3 = T1 + clobber clauses
- BlockNVM = msync included
- ByteNVM = msync suppressed
Summary

• Recent trends in extreme-scale HPC paint an uncertain future
  – Contemporary systems provide evidence that power constraints are driving architectures to change rapidly
  – Multiple architectural dimensions are being (dramatically) redesigned: Processors, node design, memory systems, I/O
  – Complexity is our main challenge

• Applications and software systems are all reaching a state of crisis
  – Applications will not be functionally or performance portable across architectures

• Programming systems must provide performance portability (beyond functional portability)!!
  – Heterogeneous systems
  – New memory hierarchies
Acknowledgements

• Contributors and Sponsors
  – US Department of Energy Office of Science
    • DOE Vancouver Project: https://ft.ornl.gov/trac/vancouver
    • DOE Blackcomb Project: https://ft.ornl.gov/trac/blackcomb
    • DOE ExMatEx Codesign Center: http://codesign.lanl.gov
    • DOE Cesar Codesign Center: http://cesar.mcs.anl.gov/
    • DOE Exascale Efforts: http://science.energy.gov/ascr/research/computer-science/
  – US National Science Foundation Keeneland Project: http://keeneland.gatech.edu
  – US DARPA
  – NVIDIA CUDA Center of Excellence
Bonus Material