Blackcomb

Jeffrey Vetter

Presented to
SOS16, Santa Barbara

http://ft.ornl.gov  vetter@computer.org
Acknowledgements

**Contributors**

- NSF Keeneland Project: [http://keeneland.gatech.edu](http://keeneland.gatech.edu)
- DOE
  - DOE Vancouver Project: [https://ft.ornl.gov/trac/vancouver](https://ft.ornl.gov/trac/vancouver)
  - DOE Blackcomb Project: [https://ft.ornl.gov/trac/blackcomb](https://ft.ornl.gov/trac/blackcomb)
  - DOE ExMatEx Codesign Center: [http://codesign.lanl.gov](http://codesign.lanl.gov)
  - DOE Exascale Efforts: [http://science.energy.gov/ascr/research/computer-science/](http://science.energy.gov/ascr/research/computer-science/)
- International Exascale Software Project: [http://www.exascale.org/iesp/Main_Page](http://www.exascale.org/iesp/Main_Page)
- DARPA NVIDIA Echelon

**Sponsors**

- NVIDIA
  - CUDA Center of Excellence
- US National Science Foundation
- US Department of Energy Office of Science
- US DARPA
Highlights

- Memory capacity and bandwidth will be a major challenge for Exascale
- New NVRAM technologies may offer a solution
  - Many open questions, however
- Blackcomb: how does NVRAM fit into Exascale?
- Initial architecture and applications results are promising
HPC Landscape Today
RIKEN/Fujitsu K: #1 in November 2011

- 10.5 PF (93% of peak)  
  - 12.7 MW
- 705,024 cores
- 1.4 PB memory

Source: Riken, Fujitsu
ORNL’s “Titan” System

- Upgrade of existing Jaguar Cray XT5
- Cray Linux Environment operating system
- Gemini interconnect
  - 3-D Torus
  - Globally addressable memory
  - Advanced synchronization features
- AMD Opteron 6200 processor (Interlagos)
- New accelerated node design using NVIDIA multi-core accelerators
  - 2011: 960 NVIDIA M2090 “Fermi” GPUs
  - 2012: 10-20 PF NVIDIA “Kepler” GPUs
- 10-20 PFlops peak performance
  - Performance based on available funds
- 600 TB DDR3 memory (2x that of Jaguar)

<table>
<thead>
<tr>
<th>Titan Specs</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute Nodes</td>
<td>18,688</td>
</tr>
<tr>
<td>Login &amp; I/O Nodes</td>
<td>512</td>
</tr>
<tr>
<td>Memory per node</td>
<td>32 GB + 6 GB</td>
</tr>
<tr>
<td>NVIDIA “Fermi” (2011)</td>
<td>665 GFlops</td>
</tr>
<tr>
<td># of Fermi chips</td>
<td>960</td>
</tr>
<tr>
<td>NVIDIA “Kepler” (2012)</td>
<td>&gt;1 TFlops</td>
</tr>
<tr>
<td>Opteron</td>
<td>2.2 GHz</td>
</tr>
<tr>
<td>Opteron performance</td>
<td>141 GFlops</td>
</tr>
<tr>
<td>Total Opteron Flops</td>
<td>2.6 PFlops</td>
</tr>
<tr>
<td>Disk Bandwidth</td>
<td>~ 1 TB/s</td>
</tr>
</tbody>
</table>

Source: OLCF
**AMD’s Llano: A-Series APU**

- Combines
  - 4 x86 cores
  - Array of Radeon cores
  - Multimedia accelerators
  - Dual channel DDR3
- 32nm
- Up to 29 GB/s memory bandwidth
- Up to 500 Gflops SP
- 45W TDP

Source: AMD
### Notional Exascale Architecture Targets

*(From Exascale Arch Report 2009)*

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>System peak</td>
<td>10 Tera</td>
<td>2 Peta</td>
<td>200 Petaflop/sec</td>
<td>1 Exaflop/sec</td>
</tr>
<tr>
<td>Power</td>
<td>~0.8 MW</td>
<td>6 MW</td>
<td>15 MW</td>
<td>20 MW</td>
</tr>
<tr>
<td>System memory</td>
<td>0.006 PB</td>
<td>0.3 PB</td>
<td>5 PB</td>
<td>32-64 PB</td>
</tr>
<tr>
<td>Node performance</td>
<td>0.024 TF</td>
<td>0.125 TF</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/sec</td>
<td>1 TB/sec</td>
<td>0.4 TB/sec</td>
</tr>
<tr>
<td>Node concurrency</td>
<td>16</td>
<td>12</td>
<td>O(100)</td>
<td>O(1,000)</td>
</tr>
<tr>
<td>System size (nodes)</td>
<td>416</td>
<td>18,700</td>
<td>50,000</td>
<td>1,000,000</td>
</tr>
<tr>
<td>Total Node Interconnect BW</td>
<td>1.5 GB/s</td>
<td>150 GB/sec</td>
<td>1 TB/sec</td>
<td>250 GB/sec</td>
</tr>
<tr>
<td>MTTI</td>
<td>day</td>
<td>O(1 day)</td>
<td>O(1 day)</td>
<td>O(1 day)</td>
</tr>
</tbody>
</table>

Challenges to Exascale

Performance Growth

1) **System power** is the primary constraint
2) **Memory** bandwidth and capacity are not keeping pace
3) **Concurrency** (1000x today)
4) **Processor** architecture is an open question
5) **Programming model** heroic compilers will not hide this
6) **Algorithms** need to minimize data movement, not flops
7) **I/O bandwidth** unlikely to keep pace with machine speed
8) **Reliability and resiliency** will be critical at this scale
9) **Bisection bandwidth** limited by cost and energy

Unlike the last 20 years most of these (1-7) are equally important across scales, e.g., 100 10-PF machines

Source: Hitchcock, Exascale Research Kickoff Meeting
Memory Bandwidth and Capacity
**Critical Concern : Memory Capacity**

<table>
<thead>
<tr>
<th></th>
<th>2010</th>
<th>2018</th>
<th>Factor Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>System peak</td>
<td>2 Pf/s</td>
<td>1 Ef/s</td>
<td>500</td>
</tr>
<tr>
<td>Power</td>
<td>6 MW</td>
<td>20 MW</td>
<td>3</td>
</tr>
<tr>
<td>System Memory</td>
<td>0.3 PB</td>
<td>10 PB</td>
<td>33</td>
</tr>
<tr>
<td>Node Performance</td>
<td>0.125 Tf/s</td>
<td>10 Tf/s</td>
<td>80</td>
</tr>
<tr>
<td>Node Memory BW</td>
<td>25 GB/s</td>
<td>400 GB/s</td>
<td>16</td>
</tr>
<tr>
<td>Node Concurrency</td>
<td>12 CPUs</td>
<td>1,000 CPUs</td>
<td>83</td>
</tr>
<tr>
<td>Interconnect BW</td>
<td>1.5 GB/s</td>
<td>50 GB/s</td>
<td>33</td>
</tr>
<tr>
<td>System Size (nodes)</td>
<td>20 K nodes</td>
<td>1 M nodes</td>
<td>50</td>
</tr>
<tr>
<td>Total Concurrency</td>
<td>225 K</td>
<td>1 B</td>
<td>4,444</td>
</tr>
<tr>
<td>Storage</td>
<td>15 PB</td>
<td>300 PB</td>
<td>20</td>
</tr>
<tr>
<td>Input/Output bandwidth</td>
<td>0.2 TB/s</td>
<td>20 TB/s</td>
<td>100</td>
</tr>
</tbody>
</table>

**Table 1:** Potential Exascale Computer Design for 2018 and its relationship to current HPC designs.

- Small memory capacity has profound impact on other features
- Feeding the core(s)
- Poor efficiencies
- Small messages, I/O
New Technologies May Offer a Solution

<table>
<thead>
<tr>
<th>Device Type</th>
<th>HDD</th>
<th>DRAM</th>
<th>NAND Flash</th>
<th>FRAM</th>
<th>MRAM</th>
<th>STTRAM</th>
<th>PCRAM</th>
<th>NRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maturity</td>
<td>Product</td>
<td>Product</td>
<td>Product</td>
<td>Product</td>
<td>Product</td>
<td>Prototype</td>
<td>Product</td>
<td>Prototype</td>
</tr>
<tr>
<td>Present Density</td>
<td>400Gb/in²</td>
<td>8Gb/chip</td>
<td>64Gb/chip</td>
<td>128Mb/chip</td>
<td>32Mb/chip</td>
<td>2Mb/chip</td>
<td>512Mb/chip</td>
<td>NA</td>
</tr>
<tr>
<td>Cell Size (SLC)</td>
<td>(2/3)F²</td>
<td>6F²</td>
<td>4F²</td>
<td>6F²</td>
<td>20F²</td>
<td>4F²</td>
<td>5F²</td>
<td>5F²</td>
</tr>
<tr>
<td>MLC Capability</td>
<td>No</td>
<td>No</td>
<td>4bits/cell</td>
<td>No</td>
<td>2bits/cell</td>
<td>4bits/cell</td>
<td>4bits/cell</td>
<td>No</td>
</tr>
<tr>
<td>Access Time (W/R)</td>
<td>9.5/8.5ms</td>
<td>10/10ns</td>
<td>200/25us</td>
<td>50/75ns</td>
<td>12/12ns</td>
<td>10/10ns</td>
<td>100/20ns</td>
<td>10/10ns [11]</td>
</tr>
<tr>
<td>Endurance/Retention</td>
<td>NA</td>
<td>10¹⁵/64ms</td>
<td>10⁸/10yr</td>
<td>10¹⁵/10yr</td>
<td>10¹⁵/10yr</td>
<td>10¹⁵/10yr</td>
<td>10¹⁵/10yr</td>
<td>10¹⁵/10yr</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Device Type</th>
<th>RRAM</th>
<th>CBRAM</th>
<th>SEM</th>
<th>Polymer</th>
<th>Molecular</th>
<th>Racetrack</th>
<th>Holographic</th>
<th>Probe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maturity</td>
<td>Research</td>
<td>Prototype</td>
<td>Prototype</td>
<td>Research</td>
<td>Research</td>
<td>Research</td>
<td>Product</td>
<td>Prototype</td>
</tr>
<tr>
<td>Present Density</td>
<td>64Kb/chip</td>
<td>2Mb/chip</td>
<td>128Mb/chip</td>
<td>128b/chip</td>
<td>160Kb/chip</td>
<td>NA</td>
<td>515Gb/in²</td>
<td>1Tb/in²</td>
</tr>
<tr>
<td>Cell Size</td>
<td>6F²</td>
<td>6F²</td>
<td>4F²</td>
<td>6F²</td>
<td>6F²</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>MLC Capability</td>
<td>2bits/cell</td>
<td>2bits/cell</td>
<td>No</td>
<td>2bits/cell</td>
<td>No</td>
<td>12bits/cell</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Access Time (W/R)</td>
<td>10/20ns</td>
<td>50/50ns</td>
<td>100/20ns</td>
<td>30/30ns</td>
<td>20/20ns</td>
<td>10/10ns</td>
<td>3.1/5.4ms</td>
<td>10/10us</td>
</tr>
<tr>
<td>Endurance/Retention</td>
<td>10⁹/10yr</td>
<td>10⁹/Months</td>
<td>10⁹/days</td>
<td>10⁴/Months</td>
<td>10⁵/Months</td>
<td>10¹⁵/10yr</td>
<td>10⁵/50yr</td>
<td>10⁵/NA</td>
</tr>
</tbody>
</table>

To More than Exascale HPC ...
Several Open Questions

- Which technologies will pan out?
  - Manufacturability, economics, system integration, timeline, device endurance

- How should NVRAM be integrated into an Exascale system?
  - Disk replacement
  - Hybrid DRAM-NVRAM main memory
  - Something else?

- How can applications make use of NVRAM
  - With no/minor changes to the application?
  - With major changes to the application?

- How can the system software and programming environment support this capability and device characteristics?
Blackcomb Overview
Blackcomb Project Overview

- Applications
  - ORNL

- Software and Programming Model
  - HP, ORNL

- System Architecture
  - HP, Michigan, ORNL

- Memory Architecture
  - PSU, HP, Michigan

- Device Technology
  - PSU

- Jeffrey Vetter, ORNL
- Robert Schreiber, HP Labs
- Trevor Mudge, Michigan
- Yuan Xie, PSU
Device-Architecture

Centip3De - 3D NTC Architecture

Holistic View of HPC

Performance, Resilience, Power, Programmability

Applications
- Materials
- Climate
- Fusion
- National Security
- Combustion
- Nuclear Energy
- Cybersecurity
- Biology
- High Energy Physics
- Energy Storage
- Photovoltaics
- National Competitiveness
- Usage Scenarios
  - Ensembles
  - UQ
  - Visualization
  - Analytics

Programming Environment
- Domain specific
  - Libraries
  - Frameworks
  - Templates
  - Domain specific languages
  - Patterns
  - Autotuners

- Platform specific
  - Languages
  - Compilers
  - Interpreters/Scripting
  - Performance and Correctness Tools
  - Source code control

System Software
- Resource Allocation
- Scheduling
- Security
- Communication
- Synchronization
- Filesystems
- Instrumentation
- Virtualization

Architectures
- Processors
  - Multicore
  - Graphics Processors
- Vector processors
- FPGA
- DSP
- Memory and Storage
  - Shared (cc, scratchpad)
  - Distributed
  - RAM
  - Storage Class Memory
  - Disk
  - Archival
- Interconnects
  - Infiniband
  - IBM Torrent
  - Cray Gemini, Aires
- BGL/P/Q
- 1/10/100 GigE
## Holistic View of HPC

### Applications
- Materials
- Climate
- Fusion
- National Security
- Combustion
- Nuclear Energy
- Cybersecurity
- Biology
- High Energy Physics
- Energy Storage
- Photovoltaics
- National Competitiveness
- Usage Scenarios
  - Ensembles
  - UQ
  - Visualization
  - Analytics

### Programming Environment
- Domain specific
  - Libraries
  - Frameworks
  - Templates
  - Domain specific languages
  - Patterns
  - Autotuners
- Platform specific
  - Languages
  - Compilers
  - Interpreters/Scripting
  - Performance and Correctness Tools
  - Source code control

### System Software
- Resource Allocation
- Scheduling
- Security
- Communication
- Synchronization
- Filesystems
- Instrumentation
- Virtualization

### Architectures
- Processors
- Multicore
- Graphics Processors
- Vector processors
- FPGA
- DSP
- Memory and Storage
  - Shared (cc, scratchpad)
  - Distributed
  - RAM
  - Storage Class Memory
  - Disk
  - Archival
- Interconnects
  - Infiniband
  - IBM Torrent
  - Cray Gemini, Aires
  - BGL/P/Q
  - 1/10/100 GigE

---

### Performance, Resilience, Power, Programmability

- Ensembles
- UQ
- Visualization
- Analytics
- Libraries
- Frameworks
- Templates
- Domain specific languages
- Patterns
- Autotuners
- Languages
- Compilers
- Interpreters/Scripting
- Performance and Correctness Tools
- Source code control
How can applications make use of NVRAM?
How could/should NVRAM be integrated into an Exascale system?

- Traditional I/O
  - Local disk replacement
  - I/O Buffer
  - Checkpoint Buffer
  - In-situ analysis

- Most of the capability and characteristics of NVRAM are hidden by system abstractions

- Integrate with memory hierarchy

- Where?
  - Cache
  - DRAM
  - New level of memory backing DRAM?
Setup

- Assume that main memory has two partitions: DRAM and NVRAM
- Byte addressable
- Similar memory infrastructure otherwise
- In general, NVRAM devices have
  - Near zero standby power
  - Higher latencies and energy for writes

Natural separation of applications objects for a hybrid DRAM-NVDRAM configuration?
Can we identify this separation with some basic metrics?

NVRAM Partition

DRAM
Our Metrics and Rationale

- **Empirical**
  - Object size – size matters for power
  - Reference rate – will it be cached
  - Read/write ratio – writes are expensive

- **Simulated**
  - Performance – needs to be competitive
  - Power – reduce it (prime directive)
Methodology: NV-Scavenger

- Empirically measure and simulate memory behavior of real applications
- Gather statistics per app object for locality, read/write ratios, etc
Focus on DOE workload

- Co-design center and other DOE applications
- Unmodified applications
- Runs in natural environment
- Initialization and finalization elided
Measurement Results

Figure 3: Read/write ratios, memory reference rates and memory object sizes for memory objects in Nek5000

Figure 6: Read/write ratios, memory reference rates and memory object sizes for memory objects in S3D
Observations: Numerous characteristics of applications are a match for byte-addressable NVRAM

- Many lookup, index, and permutation tables
- Inverted and ‘element-lagged’ mass matrices
- Geometry arrays for grids
- Thermal conductivity for soils
- Strain and conductivity rates
- Boundary condition data
- Constants for transforms, interpolation
Based on this evidence ...

- It appears that most scientific applications have a set of objects that could easily be mapped into NVRAM
  - Without modification
  - With no/little performance impact
  - With lower overall power

- Application co-design may increase this ratio significantly
Current Status
Current Applications and Software Tasks

- Continue working on software support for exploiting byte-addressable NVRAM
  - Programming models
  - Compiler support

- Understand future memory hierarchies and NVRAM insertion points

- Understand how apps might change in response to algorithms and other technology constraints
Memory integration forces important decisions: Evaluation of Llano v. other options

Note: Llano is a consumer, not server, part.
Tightly Coupled General In Situ Processing

- Simulation uses data adapter layer to make data suitable for general purpose visualization library
- Rich feature set can be called by the simulation
- Operate directly on the simulation’s data arrays when possible
- Write once, use many times

B. Whitlock, J. Favre, and J.S. Meredith, “Parallel In Situ Coupling of a Simulation with a Fully Featured Visualization System,” in Eurographics Symposium on Parallel Graphics and Visualization (EGPGV) in association with Eurographics, 2011
Bonus Slides
FAQ
Blackcomb: Hardware-Software Co-design for Non-Volatile Memory in Exascale Systems

A comparison of various memory technologies

<table>
<thead>
<tr>
<th></th>
<th>SRAM</th>
<th>DRAM</th>
<th>NAND Flash</th>
<th>PC-RAM</th>
<th>STT-RAM</th>
<th>R-RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Retention</td>
<td>N</td>
<td>H</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Memory Cell Factor ($F$)</td>
<td>50-120</td>
<td>6-10</td>
<td>2-3</td>
<td>6-12</td>
<td>4-20</td>
<td>&lt;1</td>
</tr>
<tr>
<td>Read Time (ns)</td>
<td>1</td>
<td>30</td>
<td>50</td>
<td>20-50</td>
<td>2-20</td>
<td>&lt;50</td>
</tr>
<tr>
<td>Write/Erase Time (ns)</td>
<td>1</td>
<td>50</td>
<td>10-100</td>
<td>50-120</td>
<td>2-20</td>
<td>&lt;100</td>
</tr>
<tr>
<td>Number of Rewrites</td>
<td>$10^{16}$</td>
<td>$10^{16}$</td>
<td>$10^7$</td>
<td>$10^{10}$</td>
<td>$10^{15}$</td>
<td>$10^{15}$</td>
</tr>
<tr>
<td>Power Read/Write</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Power (Other than R/W)</td>
<td>Leakage Current</td>
<td>Refresh Power</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

**Novel Ideas**

- **New resilience-aware designs for non-volatile memory applications**
  - Mechanical-disk-based data-stores are completely replaced with energy-efficient non-volatile memories (NVM).
  - Most levels of the hierarchy, including DRAM and last levels of SRAM cache, are completely eliminated.

- **New energy-aware systems/applications for non-volatile memories (nanostores)**
  - Compute capacity, comprised of balanced low-power simple cores, is co-located with the data store.

**Impact and Champions**

- Reliance on NVM addresses device scalability, energy efficiency and reliability concerns associated with DRAM
  - More memory – NVM scalability and density permits significantly more memory/core than projected by current Exascale estimates.
  - Less power – NVMs require zero stand-by power.
  - More reliable – alleviates increasing DRAM soft-error rate problem.

- **Node architecture with persistent storage near processing elements enables new computation paradigms**
  - Low-cost checkpoints, easing checkpoint frequency concerns.
  - Inter-process data sharing, easing in-situ analysis (UQ, Visualization)

**Milestones**

- Identify and evaluate the most promising non-volatile memory (NVM) device technologies.
- Explore assembly of NVM technologies into a storage and memory stack
- Build the abstractions and interfaces that allow software to exploit NVM to its best advantage
- Propose an exascale HPC system architecture that builds on our new memory architecture
- Characterize key DOE applications and investigate how they can benefit from these new technologies

Jeffrey Vetter, ORNL
Robert Schreiber, HP Labs
Trevor Mudge, U Michigan
Yuan Xie, PSU

4/20/2011
Opportunities go far beyond a plugin replacement for disk drives...

- New distributed computer architectures that address exascale resilience, energy, and performance requirements
  - replace mechanical-disk-based data-stores with energy-efficient non-volatile memories
  - explore opportunities for NVM memory, from plug-compatible replacement (like the NV DIMM, below) to radical, new data-centric compute hierarchy (nanostores)
  - place low power compute cores close to the data store
  - reduce number of levels in the memory hierarchy
- Adapt existing software systems to exploit this new capabilities
HPC Landscape Today
SPARC64™ VIIIfx Chip Overview

- **Architecture Features**
  - 8 cores
  - Shared 5 MB L2$.
  - Embedded Memory Controller
  - 2 GHz
- **Fujitsu 45nm CMOS**
  - 22.7mm x 22.6mm
  - 760M transistors
  - 1271 signal pins
- **Performance (peak)**
  - 128GFlops
  - 64GB/s memory throughput
- **Power**
  - 58W (TYP, 30°C)
  - Water Cooling – Low leakage power and High reliability
#2: Tianhe-1A uses 7,000 NVIDIA GPUs

- **Tianhe-1A uses**
  - 7,168 NVIDIA Tesla M2050 GPUs
  - 14,336 Intel Westmeres

- **Performance**
  - 4.7 PF peak
  - 2.5 PF sustained on HPL

- **4.04 MW**
  - If Tesla GPU’s were not used in the system, the whole machine could have needed 12 megawatts of energy to run with the same performance, which is equivalent to 5000 homes

- **Custom fat-tree interconnect**
  - 2x bandwidth of Infiniband QDR
Trend #1: Facilities and Power
Trend #2: Dark Silicon Will Make Heterogeneity and Specialization More Relevant

<table>
<thead>
<tr>
<th>Node</th>
<th>45nm</th>
<th>22nm</th>
<th>11nm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>2008</td>
<td>2014</td>
<td>2020</td>
</tr>
<tr>
<td>Area(^{-1})</td>
<td>1</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>Peak freq</td>
<td>1</td>
<td>1.6</td>
<td>2.4</td>
</tr>
<tr>
<td>Power</td>
<td>1</td>
<td>1</td>
<td>0.6</td>
</tr>
</tbody>
</table>

\[(4 \times 1)^{-1} = 25\%\]
\[(16 \times 0.6)^{-1} = 10\%\]

Exploitable Si (in 45nm power budget)

Source: ITRS 2008

Source: ARM
Looking Forward to Exascale
DARPA Echelon team: NVIDIA, ORNL, Micron, Cray, Georgia Tech, Stanford, UC-Berkeley, U Penn, Utah, Tennessee, Lockheed Martin
Application Memory Access Patterns

- What features do we need?
  - Scatter/Gather
  - Streaming load/store
  - Atomics, Transactions
NVRAM-Friendly Access Patterns

- Not frequently written data
- Have good memory locality
  - Memory references hit caches
- “Streaming processing” pattern
- **Target:** capture NVRAM-friendly access patterns

- **Read-Write-Total (RWT)**
  - Read/write ratio matters
  - Absolute number of memory references matters too

- $MR_R$: the number of read references to a memory region
- $MR_W$: the number of write references to a memory region
- $TR/W$: the total number of read and write references to all memory regions

\[
RWT = \left( \frac{MR_R + MR_W}{TR/W} \right) \log \left( \frac{MR_R}{MR_W} \right)
\]

- **Make RWT independent of app**
- **Read major, or write major?**
- **Absolute number matters too!**
- **Ratio matters!**
RWT: an architecture-independent metric

Last level cache miss + cache eviction ratio (CMCER)

- An architecture-dependent metric
- Reflect the number of accesses to a memory region
- Identify which memory region accesses the main memory most

\[ MR_{CM} : \text{the number of access to a memory region due to the last level cache miss} \]

\[ MR_{CE} : \text{the number of access to a memory region due to the cache eviction} \]

\[ T_{R/W} : \text{the total number of read and write references to all memory regions} \]

\[ \text{CMCER} = \frac{MR_{CM} + MR_{CE}}{T_{R/W}} \]
- Statistical data for all level cache misses
  - Bypass the cache hierarchy to save energy
  - Can we remove the cache for NVRAM? (explore new architecture designs)

- Memory usage across computation time steps

![Diagram of memory regions and computation steps]
APP1

- A computational fluid dynamic solver
  - cover a broad range of applications
  - We instrument the eddy problem, a 2D problem

- RWT

Cumulative distribution of RWT value for APP1

Read Major (i.e., $MR_R > MR_W$)

Write only

Read only

59MB Read-only Data
APP2

- A turbulent reacting flow solver
- A high-order accurate, non-dissipative numerical scheme solved on a three-dimensional structured Cartesian mesh
- RWT

Cumulative distribution of RWT value for APP1

Memory size will be further increased if we increase the number of grid points.
- **CMCER**

Most memory references hit caches

NVRAM friendly
- Statistical data for all level cache miss rates

![Graph showing cache miss rates across memory regions.](Image)

- L2 cache miss rate
- L1 cache miss rate

A candidate to bypass caches
Most references are satisfied by the caches
In-situ Analysis

**in-situ**

[in sahy-too, -tyoo, see-; *Lat.* in sit-oo] Show IPA

- **noun**
  1. situated in the original, natural, or existing place or position: *The archaeologists were able to date the vase because it was found in situ.*
  2. Medicine/Medical.
     a. in place or position; undisturbed.
     b. in a localized state or condition: *carcinoma in situ.*
Visualization and Analysis Depends on I/O

- Scaling studies with VisIt report 90% time in I/O
- Worse as concurrency increases
In Situ Analysis Avoids I/O

- Many types of analysis are amenable to calculation as the data is generated, i.e. *in situ*

- Three general types of in situ processing:

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Description</th>
<th>Caveats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tightly-coupled</td>
<td>Visualization/analysis have direct access to memory of simulation code</td>
<td>• Very memory constrained&lt;br&gt;• Potential performance, stability costs</td>
</tr>
<tr>
<td>Loosely-coupled</td>
<td>Visualization/analysis run on concurrent resources and access data over network</td>
<td>• Data movement costs&lt;br&gt;• Requires separate resources</td>
</tr>
<tr>
<td>Hybrid</td>
<td>Data is reduced in a tightly coupled setting and sent to a concurrent resource</td>
<td>• Most complex&lt;br&gt;• Shares caveats of the other strategies</td>
</tr>
</tbody>
</table>
Loosely Coupled In Situ Processing

- I/O routines stage data into secondary memory buffers, possibly on other compute nodes
- Visualization applications access the buffers and obtain data
- Separates visualization processing from simulation processing
- Copies and moves data
Tightly Coupled Custom In Situ Processing

- Custom visualization routines are developed specifically for the simulation and are called as subroutines
  - Optimized for data layout
  - Create best visual representation

- Tendency to concentrate on very specific visualization scenarios

- Write once, use once