Exploring Emerging Technologies in the HPC Co-Design Space

Jeffrey S. Vetter

Presented to
AsHES Workshop, IPDPS
Phoenix
19 May 2014
Our community expects major challenges in HPC as we move to extreme scale

- Power, Performance, Resilience, Productivity
- Major shifts in architectures, software, applications
  - Most uncertainty in two decades

Applications will have to change in response to design of processors, memory systems, interconnects, storage

- DOE has initiated Codesign Centers that bring together all stakeholders to develop integrated solutions

Technologies particularly pertinent to addressing some of these challenges

- Heterogeneous computing
- Nonvolatile memory

We need to reexamine software solutions to make this period of uncertainty palpable for computational science

- OpenARC
- Memory allocation strategies
HPC Landscape Today
# 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>5,000</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>
## Contemporary HPC Architectures

<table>
<thead>
<tr>
<th>Date</th>
<th>System</th>
<th>Location</th>
<th>Comp</th>
<th>Comm</th>
<th>Peak (PF)</th>
<th>Power (MW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>Jaguar; Cray XT5</td>
<td>ORNL</td>
<td>AMD 6c</td>
<td>Seastar2</td>
<td>2.3</td>
<td>7.0</td>
</tr>
<tr>
<td>2010</td>
<td>Tianhe-1A</td>
<td>NSC Tianjin</td>
<td>Intel + NVIDIA</td>
<td>Proprietary</td>
<td>4.7</td>
<td>4.0</td>
</tr>
<tr>
<td>2010</td>
<td>Nebulae</td>
<td>NSCS Shenzhen</td>
<td>Intel + NVIDIA</td>
<td>IB</td>
<td>2.9</td>
<td>2.6</td>
</tr>
<tr>
<td>2010</td>
<td>Tsubame 2</td>
<td>TiTech</td>
<td>Intel + NVIDIA</td>
<td>IB</td>
<td>2.4</td>
<td>1.4</td>
</tr>
<tr>
<td>2011</td>
<td>K Computer</td>
<td>RIKEN/Kobe</td>
<td>SPARC64 VIIIfx</td>
<td>Tofu</td>
<td>10.5</td>
<td>12.7</td>
</tr>
<tr>
<td>2012</td>
<td>Titan; Cray XK6</td>
<td>ORNL</td>
<td>AMD + NVIDIA</td>
<td>Gemini</td>
<td>27</td>
<td>9</td>
</tr>
<tr>
<td>2012</td>
<td>Mira; BlueGeneQ</td>
<td>ANL</td>
<td>SoC</td>
<td>Proprietary</td>
<td>10</td>
<td>3.9</td>
</tr>
<tr>
<td>2012</td>
<td>Sequoia; BlueGeneQ</td>
<td>LLNL</td>
<td>SoC</td>
<td>Proprietary</td>
<td>20</td>
<td>7.9</td>
</tr>
<tr>
<td>2012</td>
<td>Blue Waters; Cray</td>
<td>NCSA/UIUC</td>
<td>AMD + (partial) NVIDIA</td>
<td>Gemini</td>
<td>11.6</td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Stampede</td>
<td>TACC</td>
<td>Intel + MIC</td>
<td>IB</td>
<td>9.5</td>
<td>5</td>
</tr>
<tr>
<td>2013</td>
<td>Tianhe-2</td>
<td>NSCC-GZ (Guangzhou)</td>
<td>Intel + MIC</td>
<td>Proprietary</td>
<td>54</td>
<td>~20</td>
</tr>
</tbody>
</table>
Notional Future Architecture
Co-designing Future Extreme Scale Systems
Designing for the future

- Empirical measurement is necessary but we must investigate future applications on future architectures using future software stacks

Bill Harrod, 2012 August ASCAC Meeting
# 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
Holistic View of HPC – Going Forward

Large design space → uncertainty!

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
  - Algorithms

System Software
- Resource Allocation
- Scheduling
- Security
- Communication
- Synchronization
- Filesystems
- Instrumentation
- Virtualization
- Interconnects
- Infiniband
- IBM Torrent
- Cray Gemini, Aires
- BGL/P/Q
- 1/10/100 GigE

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

Large design space is challenging for apps, software, and architecture scientists.
Exascale Co-Design Center for Materials in Extreme Environments (ExMatEx)
Director: Timothy Germann (LANL)

Center for Exascale Simulation of Advanced Reactors (CESAR)
Director: Andrew Siegel (ANL)

Center for Exascale Simulation of Combustion in Turbulence (EXaCT)
Director: Jacqueline Chen (SNL)

Each project is $4M/yr for 5 years, subject to satisfactory progress as gauged by frequent reviews.
Workflow within the Exascale Ecosystem

“(Application driven) co-design is the process where scientific problem requirements influence computer architecture design, and technology constraints inform formulation and design of algorithms and software.” – Bill Harrod (DOE)

Application Design

System Design

- Vendor Analysis
  - Sim Exp
  - Proto HW
  - Prog Models
  - HW Simulator
  - Tools

- Hardware Co-Design

- HW Design

- Domain/Alg Analysis
  - Application Co-Design
  - Proxy Apps

- Open Analysis
  - Models
  - Simulators
  - Emulators

- Computer Science Co-Design

- SW Solutions

- System Software

- Stack Analysis
  - Prog models
  - Tools
  - Compilers
  - Runtime
  - OS, I/O, ...

Slide courtesy of ExMatEx Co-design team.
Emerging Architectures
Earlier Experimental Computing Systems

- The past decade has started the trend away from traditional ‘simple’ architectures
- Mainly driven by facilities costs and successful (sometimes heroic) application examples
- Examples
  - Cell, GPUs, FPGAs, SoCs, etc
- Many open questions
  - Understand technology challenges
  - Evaluate and prepare applications
  - Recognize, prepare, enhance programming models
Emerging Computing Architectures – Future

- Heterogeneous processing
  - Latency tolerant cores
  - Throughput cores
  - Special purpose hardware (e.g., AES, MPEG, RND)
  - Fused, configurable memory

- Memory
  - 2.5D and 3D Stacking
  - HMC, HBM, WIDEIO2, LPDDR4, etc
  - New devices (PCRAM, ReRAM)

- Interconnects
  - Collective offload
  - Scalable topologies

- Storage
  - Active storage
  - Non-traditional storage architectures (key-value stores)

- Improving performance and programmability in face of increasing complexity
  - Power, resilience

HPC (mobile, enterprise, embedded) computer design is more fluid now than in the past two decades.
Emerging Computing Architectures – Future

• Heterogeneous processing
  – Latency tolerant cores
  – Throughput cores
  – Special purpose hardware (e.g., AES, MPEG, RND)
  – Fused, configurable memory

• Memory
  – 2.5D and 3D Stacking
  – HMC, HBM, WIDEIO2, LPDDR4, etc
  – New devices (PCRAM, ReRAM)

• Interconnects
  – Collective offload
  – Scalable topologies

• Storage
  – Active storage
  – Non-traditional storage architectures (key-value stores)

• Improving performance and programmability in face of increasing complexity
  – Power, resilience

HPC (mobile, enterprise, embedded) computer design is more fluid now than in the past two decades.
Heterogeneous Computing

You could not step twice into the same river. -- Heraclitus
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</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>

Exploitable Si (in 45nm power budget)

Source: ITRS 2008

Source: ARM
TH-2 System

• 54 Pflop/s Peak!

• Compute Nodes have 3.432 Tflop/s per node
  – 16,000 nodes
  – 32000 Intel Xeon cpus
  – 48000 Intel Xeon phis (57c/phi)

• Operations Nodes
  – 4096 FT CPUs as operations nodes

• Proprietary interconnect TH2 express

• 1PB memory (host memory only)

• Global shared parallel storage is 12.4 PB

• Cabinets: 125+13+24 = 162 compute/communication/storage cabinets
  – ~750 m2

• NUDT and Inspur

TH-2 (w/ Dr. Yutong Lu)
DOE’s “Titan” Hybrid System: Cray XK7 with AMD Opteron and NVIDIA Tesla processors

SYSTEM SPECIFICATIONS:
- Peak performance of 27.1 PF
  - 24.5 GPU + 2.6 CPU
- 18,688 Compute Nodes each with:
  - 16-Core AMD Opteron CPU
  - NVIDIA Tesla “K20x” GPU
  - 32 + 6 GB memory
- 512 Service and I/O nodes
- 200 Cabinets
- 710 TB total system memory
- Cray Gemini 3D Torus Interconnect
- 8.9 MW peak power

4,352 ft²
And many others

- **BlueGene/Q**
  - QPX vectorization
  - SMT
  - 16 cores per chip
  - L2 with memory speculation and atomic updates
  - List and stream prefetch

- **K - Vector system**
  - SPARC64 VIIIfx
  - Tofu interconnect

- **Standard clusters**
  - Tightly integrated GPUs
  - Wide AVX – 256b
  - Voltage and frequency islands
  - Transactional memory
  - PCIe G3
Integration is continuing ...
**Fused memory hierarchy: AMD Llano**

Figure 3: SGEMM Performance (one, two, and four CPU threads for Sandy Bridge and the OpenCL-based AMD APPML for Llano’s fGPU)

Programming Heterogeneous Systems Productively
Applications must use a mix of programming models for these architectures.
Crossing the Chasm, Geoffrey A. Moore

Technology Adoption Lifecycle

How to make technology more accessible?
Realizing performance portability across contemporary heterogeneous architectures

• Can we develop a ‘write once, run anywhere efficiently’ application with advanced compilers, runtime systems, and autotuners?

Table 1: Comparison of Heterogeneous Architectures

<table>
<thead>
<tr>
<th>Property</th>
<th>CUDA</th>
<th>GCN</th>
<th>MIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming models</td>
<td>CUDA, OpenCL</td>
<td>OpenCL, C++ AMP</td>
<td>OpenCL, Cilk, TBB, LEO, OpenMP</td>
</tr>
<tr>
<td>Thread Scheduling</td>
<td>Hardware</td>
<td>Hardware</td>
<td>Software</td>
</tr>
<tr>
<td>User Managed Cache</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Global Synchronization</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>L2 Cache Type</td>
<td>Shared</td>
<td>Private per core</td>
<td>Private per core</td>
</tr>
<tr>
<td>L2 Total Size</td>
<td>upto 1.5MB</td>
<td>upto 0.5MB</td>
<td>25MB</td>
</tr>
<tr>
<td>L2 Line-size</td>
<td>128</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>L1 Data Cache</td>
<td>Read-only + Read-write</td>
<td>Read-only</td>
<td>Read-write</td>
</tr>
<tr>
<td>Native Mode</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
“Write one program and run efficiently anywhere”

- 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
      - Supports OpenMP3
    - 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.
    - Building common high level IR that includes constructs for parallelism, data movement, etc

OpenARC System Architecture

- **Input C OpenACC Program**
  - C Parser
  - OpenACC Parser
  - OpenACC Preprocessor
  - General Optimizer

- **OpenARC Compiler**
  - Host CPU Code
  - A2G Translator
  - GPU-specific Optimizer

- **Backend Compiler**
  - Device Kernel Code

- **OpenARC Runtime**
  - CUDA Driver API
  - OpenCL Runtime API
  - Other Device-specific Runtime APIs

- **Output Executable**
Performance Portability is critical and challenging

- One ‘best configuration’ on other architectures
- Major differences
  - Parallelism arrangement
  - Device-specific memory
  - Other arch optimizations

<table>
<thead>
<tr>
<th>Best Program version of</th>
<th>Executed on</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CUDA</td>
</tr>
<tr>
<td>CUDA</td>
<td>100</td>
</tr>
<tr>
<td>GCN</td>
<td>91</td>
</tr>
<tr>
<td>MIC</td>
<td>58</td>
</tr>
</tbody>
</table>
Automating selection of optimizations based on machine model

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
Optimization and Interactive Program Verification with OpenARC

- **Problem**
  - "Too much abstraction" in directive-based GPU programming!
    - Debuggability
      - Difficult to diagnose logic errors and performance problems at the directive level
    - Performance Optimization
      - Difficult to find where and how to optimize

- **Solution**
  - Directive-based, interactive GPU program verification and optimization
    - **OpenARC compiler**:
      - Generates runtime codes necessary for GPU-kernel verification and memory-transfer verification and optimization.
  - **Runtime**
    - Locate trouble-making kernels by comparing execution results at kernel granularity.
    - Trace the runtime status of CPU-GPU coherence to detect incorrect/missing/redundant memory transfers.
  - **Users**
    - Iteratively fix/optimize incorrect kernels/memory transfers based on the runtime feedback and apply to input program.

Example Optimization: Identify and Optimize Data Transfers

- By adding additional instrumentation, OpenARC can help identify redundant and incorrect data transfers
- User can optimize by adding pragmas

Figure 1: Total data transferred by unopt memory management scheme and its execution normalized to fully optimal scheme

Listing 4: Sample debugging messages for JACOBI kernel in Listing 3. update0 refers to `memcpyout(b)` in line 8 in Listing 3.
Future Directions in Heterogeneous Computing

• Over the next decade: Heterogeneous computing will continue to increase in importance
  – Embedding and mobile community have already experienced this trend

• Manycore
  – Integrated GPUs, special purpose HW

• Hardware features
  – Transactional memory
  – Random Number Generators
    • MC caveat
  – Scatter/Gather
  – Wider SIMD/AVX
  – AES, Compression, etc

• Synergies with BIGDATA, mobile markets, graphics

• Top 10 list of features to include from application perspective. Now is the time!

• The future is about new productive programming models

• Inform applications teams to new features and gather their requirements
Memory Systems

The Persistence of Memory

# 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></td>
</tr>
</tbody>
</table>

http://science.energy.gov/ascr/news-and-resources/workshops-and-conferences/grand-challenges/
Notional Future Node Architecture

- NVM to increase memory capacity
- Mix of cores to provide different capabilities
- Integrated network interface
- Very high bandwidth, low latency to on-package locales
Blackcomb: Comparison of emerging memory technologies

http://ft.ornl.gov/trac/blackcomb

<table>
<thead>
<tr>
<th></th>
<th>SRAM</th>
<th>DRAM</th>
<th>eDRAM</th>
<th>NAND Flash</th>
<th>PCRAM</th>
<th>STT RRAM</th>
<th>ReRAM (1T1R)</th>
<th>ReRAM (Xpoint)</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>
</tr>
<tr>
<td>Cell Size (F²)</td>
<td>50-200</td>
<td>4-6</td>
<td>19-26</td>
<td>2-5</td>
<td>4-10</td>
<td>8-40</td>
<td>6-20</td>
<td>1-4</td>
</tr>
<tr>
<td>Read Time (ns)</td>
<td>&lt; 1</td>
<td>30</td>
<td>5</td>
<td>10⁴</td>
<td>10-50</td>
<td>10</td>
<td>5-10</td>
<td>50</td>
</tr>
<tr>
<td>Write Time (ns)</td>
<td>&lt; 1</td>
<td>50</td>
<td>5</td>
<td>10⁵</td>
<td>100-300</td>
<td>5-20</td>
<td>5-10</td>
<td>10-100</td>
</tr>
<tr>
<td>Number of Rewrites</td>
<td>10¹⁶</td>
<td>10¹⁶</td>
<td>10¹⁶</td>
<td>10⁴-10⁵</td>
<td>10⁹-10¹²</td>
<td>10¹⁵</td>
<td>10⁸-10¹²</td>
<td>10⁶-10¹⁰</td>
</tr>
<tr>
<td>Read Power</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low</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>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>
</tr>
</tbody>
</table>
NVRA M Technology Continues to Improve – Driven by Market Forces

News & Analysis

3D NAND Transition: 15nm Process Technology Takes Shape

Gary Hillson
5/13/2014 08:15 AM EDT
5 comments

TORONTO — With 3D NAND unlikely to make much of a dent in 2015, SanDisk and its flash foundry recently announced 15nm process technology.

SanDisk's 12-nm technology will be applied to 1-bit-per-cell and 3-bit-per-cell NAND flash memory architectures. It will initially ramp to support second generation 19nm process technology, followed by third generation 11nm process technology.

Toshiba's new process replaces its 19nm process, which is aimed at providing a transitional step to 3D-NAND. Toshiba, senior VP of Toshiba America Electronic Components, said Toshiba's 15nm process, in conjunction with improved peripheral circuitry, will achieve the same write speed as a 10nm process technology.

Nelson said there is room to advance floating-gate memory for a wide range of consumer and commercial applications including embedded NAND storage and solid-state drives.

The V-NAND component has the same memory capacity as a 128GB chip, but with a floating-gate stack that provides a higher bit density.
Early Uses of NVRAM: Burst Buffers

Tradeoffs in Exascale Memory Architectures

- Understanding the tradeoffs
  - ECC type, row buffers, DRAM physical page size, bitline length, etc

Programming Interfaces Example: NV-HEAPS

Figure 1. The NV-heap system stack. This organization allows read and write operations to bypass the operating system entirely.

```java
class NVList : public NVOBJECT {
    DECLARE_POINTER_TYPES(NVList);
    public:
        DECLARE_MEMBER(int, value);
        DECLARE_PTR_MEMBER(NVList::NVPtr, next);
    }

    void remove(int k) {
        NVHeap * nv = NVHOpen("foo.nvheap");
        NVList::NVPtr a =
            nv->GetRoot<NVList::NVPtr>();
        AtomicBegin {
            while(a->get_next() != NULL) {
                if (a->get_next()->get_value() == k) {
                    a->set_next(a->get_next()->get_next());
                }
                a = a->get_next();
            }
        } AtomicEnd;
    }
```

Figure 2. NV-heap example. A simple NV-heap function that atomically removes all links with value k from a non-volatile linked list.

New hybrid memory architectures: What is the ideal organizations for our applications?

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 good 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
- ...
Redesigning algorithms for multi-mode memory systems
Rethinking Algorithm-Based Fault Tolerance

- Algorithm-based fault tolerance (ABFT) has many attractive characteristics
  - Can reduce or even eliminate the expensive periodic checkpoint/rollback
  - Can bring negligible performance loss when deployed in large scale
  - No modifications from architecture and system software

- However
  - ABFT is completely opaque to any underlying hardware resilience mechanisms
  - These hardware resilience mechanisms are also unaware of ABFT
  - Some data structures are over-protected by ABFT and hardware

We consider ABFT using a holistic view from both software and hardware

• We investigate how to integrate ABFT and hardware-based ECC for main memory

• ECC brings energy, performance and storage overhead

• The current ECC mechanisms cannot work
  – There is a significant semantic gap for error detection and location between ECC protection and ABFT

• We propose an explicitly-managed ECC by ABFT
  – A cooperative software-hardware approach
  – We propose customization of memory resilience mechanisms based on algorithm requirements.
System Designs

• Architecture
  – Enable co-existence of multiple ECC
  – Introduce a set of ECC registers into the memory controller (MC)
  – MC is in charge of detecting, locating, and reporting errors

• Software
  – The users control which data structures should be protected by which relaxed ECC scheme by ECC control APIs.
  – ABFT can simplify its verification phase, because hardware and OS can explicitly locate corrupted data
Evaluation

- We use four ABFT (FT-DGEMM, FT-Cholesky, FT-CG and FT-HPL)

- We save up to 25% for system energy (and up to 40% for dynamic memory energy) with up to 18% performance improvement
Future Directions in Next Generation Memory

- Next decade will be exciting for memory technology
- New devices
  - Flash, ReRam, STTRAM will challenge DRAM
  - Commercial markets already driving transition
- New configurations
  - 2.5D, 3D stacking removes recent JEDEC constraints
  - Storage paradigms (e.g., key-value)
  - Opportunities to rethink memory organization
- Logic/memory integration
  - Move compute to data
  - Programming models

<table>
<thead>
<tr>
<th></th>
<th>SRAM</th>
<th>DRAM</th>
<th>eDRAM</th>
<th>NAND Flash</th>
<th>PCRAM</th>
<th>STTRAM</th>
<th>ReRAM (ITIR)</th>
<th>ReRAM (Xpoint)</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>
</tr>
<tr>
<td>Cell Size (F7)</td>
<td>50-200</td>
<td>4-6</td>
<td>19-26</td>
<td>2-5</td>
<td>4-10</td>
<td>8-40</td>
<td>6-20</td>
<td>1-4</td>
</tr>
<tr>
<td>Read Time (ns)</td>
<td>$&lt;1$</td>
<td>30</td>
<td>5</td>
<td>$10^4$</td>
<td>10-50</td>
<td>10</td>
<td>5-10</td>
<td>50</td>
</tr>
<tr>
<td>Write Time (ns)</td>
<td>$&lt;1$</td>
<td>50</td>
<td>5</td>
<td>$10^4$</td>
<td>100-300</td>
<td>5-20</td>
<td>5-10</td>
<td>10-100</td>
</tr>
<tr>
<td>Number of Rewrites</td>
<td>$10^{15}$</td>
<td>$10^{15}$</td>
<td>$10^{11}$</td>
<td>$10^9$-$10^{10}$</td>
<td>$10^5$-$10^6$</td>
<td>$10^3$-$10^4$</td>
<td>$10^2$-$10^3$</td>
<td>$10^2$-$10^3$</td>
</tr>
<tr>
<td>Read Power</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Low</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>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>
</tr>
</tbody>
</table>

- Refactor our applications to make use of this new technology
- Add HPC programming support for these new technologies
- Explore opportunities for improved resilience, power, performance
Summary

- Our community expects major challenges in HPC as we move to extreme scale
  - Power, Performance, Resilience, Productivity
  - Major shifts and uncertainty in architectures, software, applications

- Applications will have to change in response to design of processors, memory systems, interconnects, storage
  - DOE has initiated Codesign Centers that bring together all stakeholders to develop integrated solutions

- Technologies particularly pertinent to addressing some of these challenges
  - Heterogeneous computing
  - Nonvolatile memory

- We need to reexamine software solutions to make this period of uncertainty palpable for computational science
  - OpenARC
  - Memory use and allocation strategies

- New book surveys the international landscape of HPC
- 24 chapters with many of today’s top systems/facilities: Titan, Tsubame2, BlueWaters, Tianhe-1A

http://j.mp/YhLiQP
Q & A

More info: vetter@computer.org
Recent Publications from FTG (2012-3)


Recent Publications from FTG (2012-3)


