Introduction to Pre-Workshop Document and BOG Structure

23-Jan-2018
Organizing Committee w/ Additional Contributors
Overview

Pre-Workshop Document for

Extreme Heterogeneity 2018

DOE ASCR Workshop on Extreme Heterogeneity
23-25 Jan 2018

DRAFT compiled on January 15, 2018 08:43 Eastern
Basic Research
In Extreme Heterogeneity

2 Basic Research in Extreme Heterogeneity of HPC Systems
  2.1 The Programming Environment (McCormick) .................................................. 8
  2.1.1 Programming Models and Languages ......................................................... 8
  2.1.2 Libraries and Runtime Systems ..................................................................... 10
  2.1.3 Compilers ....................................................................................................... 10
  2.1.4 Domain-Specific Environments ...................................................................... 12
  2.2 Operating System and Resource Management (Brightwell) ............................ 14
    2.2.1 Current Status and Recent Advances ......................................................... 14
    2.2.2 Scientific Challenges and Opportunities .................................................... 16
  2.3 Tools for Systems Management and Administration (Antypas) ................... 17
    2.3.1 Scheduling .................................................................................................. 18
    2.3.2 Diagnostic Tools and Provisioning .............................................................. 18
    2.3.3 Programming Environment and Workflows ............................................. 18
  2.4 Productivity and Performance Metrics and Tools (Humble) ......................... 19
    2.4.1 User Productivity ....................................................................................... 19
    2.4.2 Understandability (and Maintainability) .................................................... 19
    2.4.3 Reproducibility and Performance .............................................................. 20
  2.5 Software Development Methodologies (Li and McInnes) ............................... 22
    2.5.1 Status and Recent Advances ..................................................................... 23
    2.5.2 Scientific Challenges and Opportunities .................................................... 24
  2.6 Data Management and I/O (Ross) ................................................................. 24
    2.6.1 Status and Recent Advances ..................................................................... 24
    2.6.2 Challenges and Opportunities ................................................................... 25
  2.7 Data Analytics and In Situ Workflow Management (Yoo) .............................. 26
    2.7.1 Status and Recent Advances ..................................................................... 26
    2.7.2 Challenges and Opportunities ................................................................... 28
  2.8 Modeling and Simulation (Gokhale) ............................................................... 28
    2.8.1 Status and Recent Advances ..................................................................... 29
    2.8.2 Challenges and Opportunities ................................................................... 29
Breakout Group Organization

<table>
<thead>
<tr>
<th>BOG ID</th>
<th>B/O Topic</th>
<th>Tu PM</th>
<th>Wed PM-1</th>
<th>Wed PM-2</th>
<th>Th AM</th>
<th>Th PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Prog Env. Abstractions, Models, and Languages</td>
<td>Aiken/McCormick</td>
<td></td>
<td></td>
<td>McCormick</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Data Management and I/O</td>
<td>Ross/Byna</td>
<td></td>
<td></td>
<td>Ross</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Data Analytics and Workflows</td>
<td>Tom P./Yoo</td>
<td></td>
<td>Brightwell</td>
<td>Christine S./Bethel</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>OS/RM: global, composition, workflow</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Software Development Methodologies</td>
<td></td>
<td></td>
<td></td>
<td>Li/Bernhold</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Crosscut: Modeling and Simulation</td>
<td>Chien/Donofrio/Leidel</td>
<td></td>
<td></td>
<td>Wilke/Lan/Gokhale</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Prog Env. Compilers, Libraries, and Runtimes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>System Management, Admin, Job Scheduling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Crosscut: productivity, composition, interoperability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>OS/RM: local, prog env support</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Crosscut: Portability, code reuse, performance portability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Prog Env. Debugging and Correctness, autotuning, specialization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Crosscut: resilience, power</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The Challenges are Daunting

- **Exponentially Increasing Parallelism** (central challenge for ECP, but will be even worse)
  - Trend: *End of exponential clock frequency scaling (end of Dennard scaling)*
  - Consequence: *Exponentially increasing parallelism*

- **End of Lithography as Primary Driver for Technology Improvements**
  - Trend: *Tapering of lithography Scaling*
  - Consequence: *Many forms of heterogeneous acceleration (not just GPGPUs anymore)*

- **Data Movement Heterogeneity and Increasingly Hierarchical Machine Model**
  - Trend: *Moving data operands costs more than computation performed on them*
  - Consequence: *More heterogeneity in data movement and more pronounced NUMA effects*

- **Performance Heterogeneity**
  - Trend: *Heterogeneous execution rates from contention and aggressive power management*
  - Consequence: *Extreme variability and heterogeneity in execution rates*

- **Emerging Memory and Storage Technologies**
  - Trend: *Emerging memory technologies and stall in disk performance improvements*
  - Consequence: *Disruptive changes to our storage environment*

- **User Requirements**
  - Trend: *Diverse and Complex and heterogeneous scientific workflows*
  - Consequence: *Complex mapping of heterogeneous workflows on heterogeneous systems.*
Opportunities Exist

- If ASCR does not confront these challenges through new research
  - HPC is consigned to only modest improvements beyond exascale
  - Complexity will make code maintenance impractical or unsustainable in the long term
  - Overall: cost/complexity impedes long-term pursuit of scientific discovery using HPC

- Most options drive towards “extreme heterogeneity”
Opportunities Exist
but the paths are non-traditional -- You must think bigger!

IEEE Computer December 2015 special issue on Rebooting Computing


John Shalf
Section Template

● Define Topic and Scope (what is included, not included)
  ○ Other relevant topics, if nec
● FSD Status (very brief)
● Example Research Direction
● Name Relevant B/O Groups
Programming Environments

- Layers of the software stack responsible for enabling effective and efficient utilization of HPC resources
  - Design, implementation, evaluation and optimization of applications and supporting software infrastructures (e.g. data analysis, visualization, solvers, etc.)
- Goal: Bridge the gap between the nuances of the underlying layers (both software and hardware) and the programmer. But also make it possible to understand and reason about the impact those nuances have – e.g., debugging, performance analysis and optimization, etc.
- Many challenges: managing topology, locality, affinity and asynchronous execution within complex compositions of libraries/workflows in a performance portable fashion…
- The FSD is “alive” (i.e. still needs further contributions)
  - Thanks to: Pavan Balaji, Barbara Chapman, Mary Hall, Costin Iancu, Martin Kong, Sriram Krishnamoorthy
Programming Environments (continued)

Breakout groups:

- Abstractions, Models and Languages: Tuesday 5:25-7:00pm
- Compilers, Libraries and Runtimes: Wednesday 4:30-6:00pm
- Abstractions, Models and Languages: Thursday 11:40-1:40pm
- Debugging, Autotuning, and Specialization: Thursday 1:40-3:10pm

Cross-cuts:

- Productivity, Composability and Interoperability: Wednesday 4:30-6:00pm
- Portability, Code Reuse, & Performance Portability: Thursday 11:40-1:40pm
- Locality and Programming Environment Support: Thursday 11:40-1:40pm
Operating Systems and Resource Management (1/2)

● Scope includes local and global hardware-level resource management
● FSD captures status and recent advances
  ○ OS design - monolithic kernel, virtual memory, moving beyond accelerators as devices
  ○ Runtime systems - support for GPUs, interfaces like HSA
  ○ Resource isolation strategies - virtual machines, multi-kernels, containers
  ○ Increasing complexity of resource management - task overdecomposition, code coupling
  ○ Managing heterogeneous memory - portable interfaces for memory hierarchy
● FSD challenges and opportunities
  ○ Locality and access - automatic discovery of locality optimizations
  ○ Introspection - runtime system as target for resource usage info
  ○ Temporal non-determinism - variation in cores, memory latency/bw
  ○ OS complexity - formal methods to understand OS/HW state, reliability
  ○ Decentralized resource management - avoid serialization, logic in HW
  ○ Memory capabilities - convergence of memory and storage, logic in memory
  ○ Usage models - move beyond batch-scheduled, space-shared model
Operating Systems and Resource Management (2/2)

- Example research direction
  - Mapping the machine to the application
    - Dynamic discovery of resource and resource utilization
    - Make informed decisions on policies and mechanisms
    - Understand behavior in response to decisions
    - Respond to changing cost models during execution

- FSD contributors
  - Mike Lang, Chris Rossbach, Kamil Iskra

- Specific B/O groups
  - 4: OS/RM: global, composition, workflow - 2A Wednesday afternoon
  - 11: OS/RM: local, prog env support - 4C Thursday morning

- Not-so-relevant B/O group
  - 5: Software Development Methodologies - 2B Wednesday afternoon

- Relevant B/O groups
  - 1-3, 6-10, 12-14
System Productivity

- How do we currently evaluate productivity in the context of extreme heterogeneity and what challenges remain?
  - Many definitions for HPC productivity have been tied to growth in processor performance.
  - Extreme heterogeneity challenges the idea that only processors dominate performance.
- The FSD captures system productivity in terms of execution efficiency, user productivity, understandability, maintainability, reproducibility, workflows.
  - Example: Scientific applications increasingly use workflows for orchestrating simulations. This requires performance metrics and provenance characteristics to be extracted with greater granularity. How will we meet these demands for each definition of productivity?
- Relevant Breakout Groups
  - 2B: Software development methodologies, WED 2:40 PM ET
  - 3D: Crosscut: Productivity, composition, interoperability, WED 4:30 PM ET
  - 4D: Crosscut: Portability, code reuse, performance portability, THU 11:40 ET
Software Development Methodologies

● **Scope**: Methodologies, tools, and processes that promote the productivity of software developers and sustainability of software artifacts
  ○ Programmability, usability, composability, interoperability
  ○ Sustainability, maintainability, portability

● **FSP captures status and recent advances**
  ○ Best practices in scientific software devpt: WhatIs and HowTo docs, webinars, tutorials, etc.
  ○ Community policies for interoperable numerical software: xSDK community package policies

● **Challenges**
  ○ Node-level interoperability, programming models

● **Breakout Groups on Software Development Methodologies**
  ○ 2B: Software Development Methodologies (5). Wed. 2:40 - 4:15 pm
Data Management

Reflections of EH in data management and I/O aspects of future systems.

- Suren Byna and I are organizing the BOGs
- Breakout today (BOG-2, 5:25pm eastern)!
- Follow-up on Thursday (refining material from today)
DM: Status and Recent Advances (1/3)

- **Hardware technologies and system architecture**
  - Memory, solid-state, traditional disk, tape
  - Tiered memory and storage subsystem:
    - Memory, performance tier (burst buffers using nonvolatile storage -- on and off compute-platform)
    - Capacity tier (PFS or distributed object store using disks -- off compute-platform)
  - Specialized devices:
    - Attached to system network, accessible via RPCs
    - Samsung Key Value SSDs

- **Data management and I/O**
  - Data models -- Arrays, Key / Value stores
  - Writes - Data placement on storage -- mostly POSIX files, few using databases
  - Reads - Access to entire data variables, subsets of data variables
  - Metadata management -- file info, data descriptions, provenance

Thanks to Gary Grider for his help in putting the DM assessment together, to Suren Byna for converting into slides.
DM: Status and Recent Advances (2/3)

- **Applications**
  - Simulation codes
    - Bursty I/O, write-heavy, and large datasets
    - Checkpoints in the form of POSIX files
  - Data intensive apps, such as experimental and observational data
    - Large input datasets, vary widely on how much data is they (apps) generate
    - Some real-time/streaming
  - Emerging learning apps
    - Often operate on large and unstructured dataset input during learning phase, produce large volumes of data in an uncoordinated manner
DM: Challenges and Opportunities (3/3)

- **Reducing cost**
  - Current methods often use “one size fits all” approach (e.g., parallel file systems)
  - Emerging technologies have the potential to “right size” services
    - Metadata management to the needs of apps
    - Develop cost-effective specialize hardware for data management roles
    - Enable capturing and storing salient information (such as provenance)

- **Enabling new paradigms**
  - Current reliance on the POSIX model of data storage
  - Emerging: key value stores, object storage into the hierarchy, grouping data for effective management and access

- **Leveraging new technologies**
  - Use of intelligent storage devices could allow new functionalities
  - Opportunity for lightweight, user-space services in the storage hierarchy or directly attached to the system network
Data Analytics and In-situ Workflow Management

- DA&W scope include heterogeneous HW (FPGA, NC, QC), Software Stacks (HPC & big data), workloads, and HPC-Web-Cloud workflow management

- FSD Status
  - Heterogeneous HW for DA
  - Big data framework including Spark/Hadoop, TF/CNTK/Caffe+MPI
  - Status of HPC and distributed web & cloud workflow management

- Potential Research Direction
  - Specialized or disparate HW & Energy optimization in EH DA&W
  - Optimizing various levels of tasks in EH
  - Exploiting HPC and enterprise big data system software stack

- Relevant B/O Groups
  - BOG1: Data Analytics and Workflow -- Tue. 5:25 - 7:00pm lead by T. Peterka / S. Yoo
  - BOG3: Data Analytics and Workflow -- Wed. 4:30 - 6:00pm lead by C. Sweeney / W. Bethel
  - BOG2: Three Sessions in AI & ML for EH -- Wed 2:40 - 4:15pm
Modeling and simulation

• Quantitatively assess new architectural features
• Be capable of evaluating a wide range of granularities from microarchitecture to workflow level
• Be capable of integrating simulation components of novel heterogeneous accelerators such as quantum, neuromorphic, approximate, analog
• Properly assess characteristics of component interconnects
• Be scalable
• Characterize performance, power, thermal, reliability … attributes
Node level modsim (BO # 6, Wed. afternoon)

• Complex CPUs that incorporate wide variability in function units
  • Reduced precision, mixed precision, special purpose IP blocks
• Data visibility and sharing between closely coupled heterogeneous units
  • Accurately model complex protocols
  • Fast and scalable to enable comprehensive simulation (catch corner cases that show up non-deterministically and infrequently)
• Model many different memory types
  • Potential combinatorial explosion of combinations
  • Compute in memory or near memory
System level modsim (BO # 6, Thurs. morning)

• Model the interactions of interconnect networks
  • From experimental/observational data sources to high performance compute resources
• Accurately characterize overheads of task dispatch, synchronization
• Model overheads for data format conversions
• Model novel compute in wire capabilities
• Timing accurate transmission and routing in the presence of in-transit compute
• Workflow level modsim
Summary