Exceptional service in the national interest

Meeting the Future Needs of HPC with MPI

Ron Brightwell, R&D Manager

Scalable System Software Department

CCR Center for Computing Research

Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Goals For This Talk

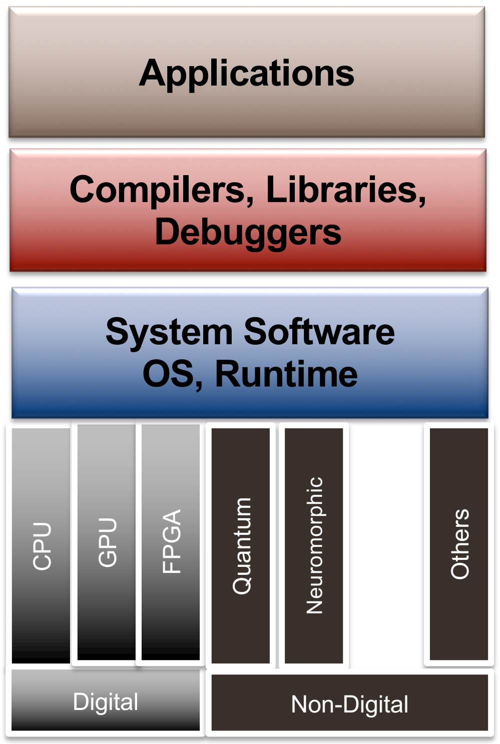
- Describe two recent DOE/ASCR workshops that explored research needs relevant to MPI
- Offer some thoughts on current and future challenges to and for MPI
- Be thought-provoking and maybe controversial

ASCR Extreme Heterogeneity Workshop

January 23-25, 2018 Virtual Meeting

Goal: Identify Priority Research Directions for Computer Science needed to make future supercomputers usable, useful and secure for science applications in the 2025-2040 timeframe

- POC: Lucy Nowell (Lucy.Nowell@science.doe.gov)
- Primary focus on the software stack and programming models/environments/tools
- 150+ participants: DOE Labs, academia, & industry
- White papers solicited (106 received!) to contribute to the FSD, identify potential participants, and help refine the agenda



Workshop Organizers & Program Committee

- Jeff Vetter (ORNL), Lead Organizer and Program Committee Chair
- Organizing Committee and Program Committee Members:

  Ron Brightwell (Sandia-NM), Pat McCormick (LANL), Rob Ross (ANL), John Shalf (LBNL)
- Program Committee Members

  Katy Antypas (LBNL, NERSC), David Donofrio (LBNL), Maya Gokhale (LLNL), Travis Humble (ORNL), Catherine Schuman (ORNL), Brian Van Essen (LLNL), Shinjae Yoo (BNL)

Workshop Plenary Talks

- **Welcome and ASCR Update**
  Barbara Helland, Director, ASCR
- **View from ASCR Research Division**
  Steve Lee, Acting Division Director
- **Invited Plenary Talk: IEEE Rebooting Computing**
  Tom Conte
- **Architectural Trends and System Design Issues**
  Bob Colwell
- **Introduction to Extreme Heterogeneity**
  Jeffrey Vetter, John Shalf, and Maya Gokhale
- **Report on the ASCAC Future of Computing study**
  Maya Gokhale
- **Panel on Issues Raised by Extreme Heterogeneity**
  Moderator Ron Brightwell
  Usability, Understandability and Programmability – Salman Habib
  Operating and Runtime Systems – Ron Brightwell
  Data Analytics – Wes Bethel
  EH Workflow Management – Ewa Deelman
- **Memory Systems and I/O**
  Bruce Jacob
- **ORNL/DoD Beyond CMOS Workshops**
  Neena Imam
- **Exascale Computing Project CS R&D for Heterogeneity**
  Mike Heroux

Breakout Groups

1A - Programming Environments, Models, & Languages
Alex Aiken & Pat McCormick
1B - Data Management and I/O
Rob Ross & Suren Byna
1C - Data Analytics and Workflows
Tom Peterka & SJ Yoo
2A - Operating Systems and Resource Management
Ron Brightwell
2B- Software Development Methodologies
Sherry Li
2C - Modeling & Simulation for Hardware Characterization
Andrew Chien & David Donofrio
3A - Programming Environments: Compilers, Libraries and Runtimes
Michelle Strout & Barbara Chapman
3B - Data Analytics and Workflows
Christine Sweeney
3C - System Management, Administration & Job Scheduling
Paul Peltz & Rebecca Hartman-Baker
3D - Crosscut: Productivity, Composability, & Interoperability
Bob Lucas
4A - Programming Environments: Abstractions, Models and Languages
Pat McCormick
4B - Crosscut: Modeling and Simulation
Jeremy Wilke & Zhiling Lan
4C - OS & Resource Management: Locality & Programming Environment Support
Mike Lang
4D -Crosscut: Portability, Code Reuse and Performance Portability
Anshu Dubey
5A -Data Management and I/O
Rob Ross
5B- Programming Environments: Debugging, Autotuning, & Specialization
Mary Hall & John Mellor-Crummey
5C -Crosscut: Resilience & Power Management
Franck Cappello & Kirk Cameron

Sandia National Laboratories

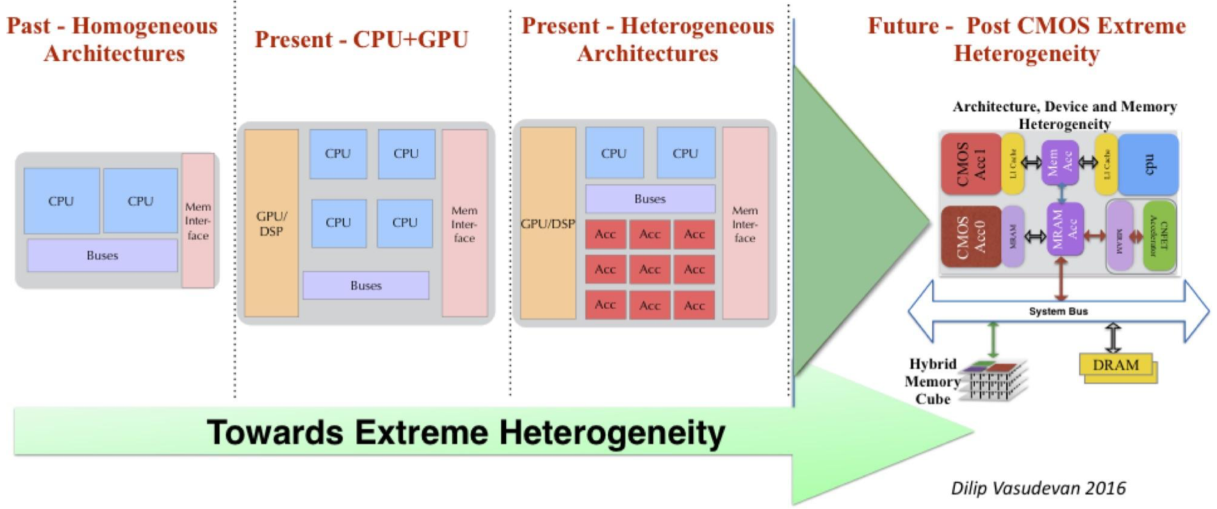
What Do We Mean by Extreme Heterogeneity?

Sandia National Laboratories

- **Exponentially Increasing Parallelism** (central challenge for ECP, but will be 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 performance and energy cost*
- **Performance Heterogeneity**
  - **Trend**: *Heterogeneous execution rates from contention and aggressive power management*
  - **Consequence**: *Extreme variability and heterogeneity in execution rates*
- **Diversity of Emerging Memory and Storage Technologies**
  - **Trend**: *Emerging memory technologies and stall in disk performance improvements*
  - **Consequence**: *Disruptive changes to our storage environment*
- **Increasingly Diverse User Requirements**
  - **Trend**: *Diverse and Complex and heterogeneous scientific workflows*
  - **Consequence**: *Complex mapping of heterogeneous workflows on heterogeneous systems.*

The Challenge of Heterogeneity

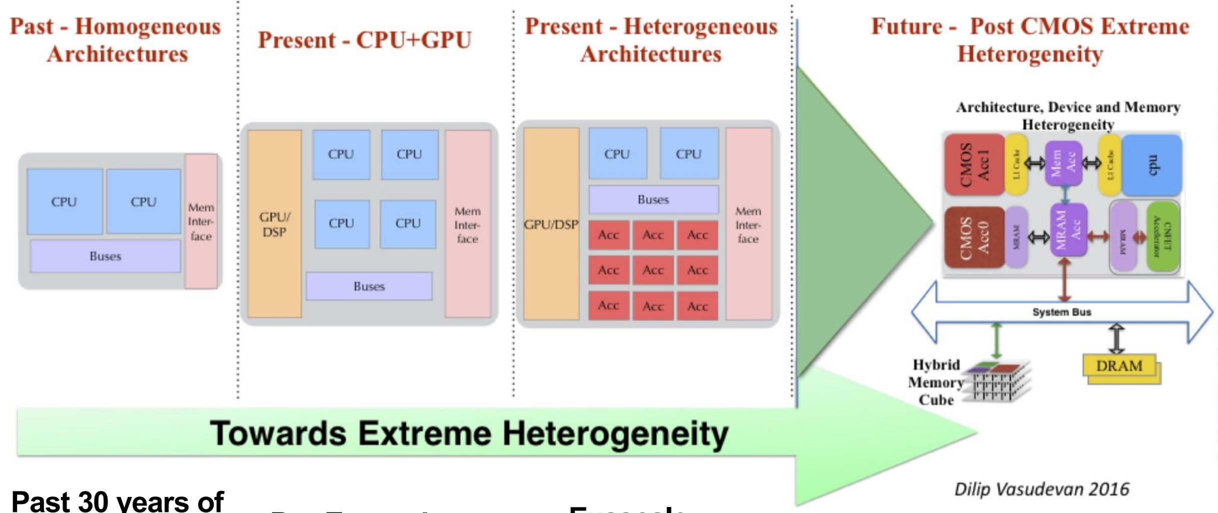
- ***"A challenge of heterogeneity is how to build large systems comprised of massive numbers of these already heterogeneous systems"*** Bob Colwell (former Intel chip architect and DARPA MTO Director)
- **If ASCR does not confront these challenges through new research**
    - HPC is consigned to only 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



Dilip Vasudevan 2016

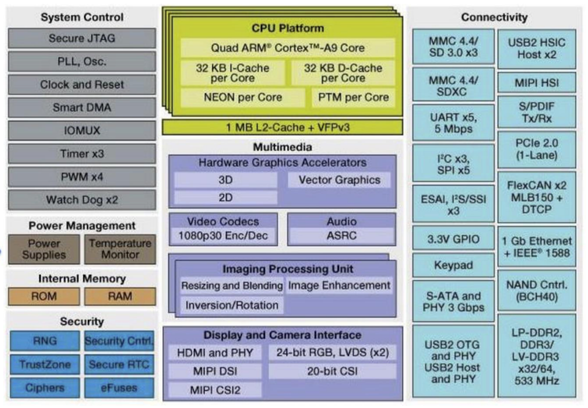
The Challenge of Heterogeneity

- ***"A challenge of heterogeneity is how to build large systems comprised of massive numbers of these already heterogeneous systems"*** Bob Colwell (former Intel chip architect and DARPA MTO Director)
- **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



Dilip Vasudevan 2016

<span style="color:red">This is already happening TODAY!</span>
Below is a SmartPhone SoC circa 2016
Dozens of kinds of integrated HW acceleration



Past 30 years of Parallel systems
(1,000,000,000x of scaling)

Pre-Exascale
(Titan/Summit)

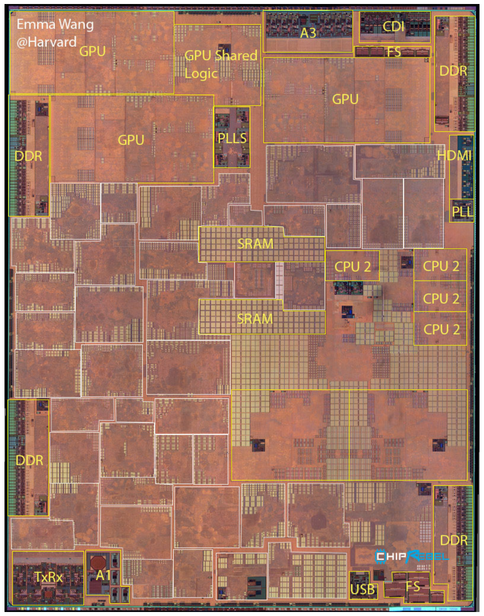
Exascale
(A21/Coral2)

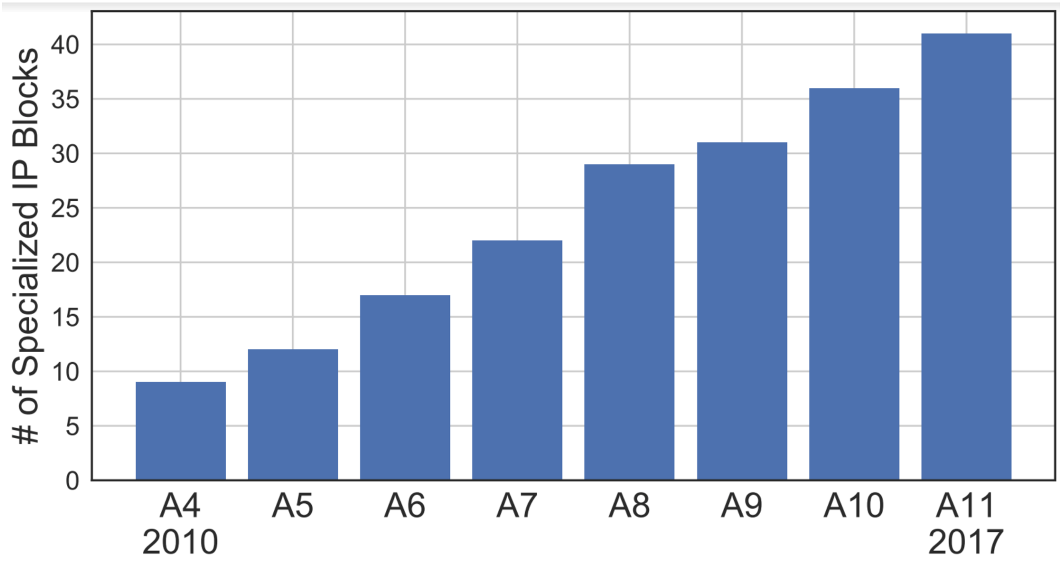
Post-Exascale
(???)

<span style="color:red">10+ years to make GPU accelerators usable for science</span>
Will it take us 100 years to get 10 more of them usable?
Or will HPC fall behind the rest of the computing industry?

Extreme Specialization Happening Now

(and it will happen to HPC too… will we be ready?)



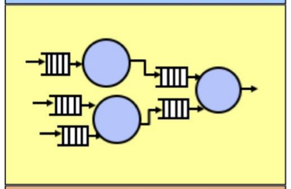


http://vlsiarch.eecs.harvard.edu/accelerators/die-photo-analysis

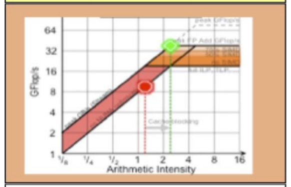
Extreme Heterogeneity Priority Research Directions (PRDs)

```
an_done) {
if(plant[0].bar_id == 1)
  temp_mode = mode;
  mode = watch_bars;
  file_output(
  an_done = on;
  }
```

- **Maintaining and improving programmer productivity**
  - Flexible, expressive, programming models and languages
  - Intelligent, domain-aware compilers and tools
  - Composition of disparate software components



- **Managing resources intelligently**
  - Automated methods using introspection and machine learning
  - Optimize for performance, energy efficiency, and availability



- **Modeling & predicting performance**
  - Evaluate impact of potential system designs and application mappings
  - Model-automated optimization of applications

- **Enabling reproducible science despite non-determinism & asynchrony**
  - Methods for validation on non-deterministic architectures
  - Detection and mitigation of pervasive faults and errors

Data Analytics
Data Modeling
Business Intelligence & Reporting
Extract, Transform, and Load
Data Migration
Data Architecture & Structure

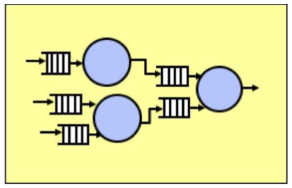
- **Facilitating execution of science workflows**
  - Mapping of science workflows to heterogeneous hardware & software services
  - Adapting workflows & services to meet facility-level objectives through learning approaches

Maintaining & Improving Programmer Productivity

```
an_done) {
if(plant[0].bar_id == 1)
    temp_mode = mode;
    mode = watch_bars;
    file_output(
    an_done = on;
    }
```

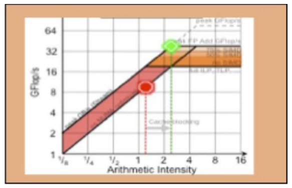
- **Programmability:** A diverse range of applications and memory/storage and hardware accelerator technologies will dramatically raise the complexity of programming challenges. Our programming mechanisms must change to maintain, let alone improve, the *current time-to-discovery*.
  - Requires more sophisticated implementations and range of capabilities (descriptive and prescriptive techniques) to fully utilize system capabilities.
  - Must be combined with higher levels of abstraction and domain-awareness to reduce complexity for user-facing interfaces, and increase flexibility and introspection for low-level developer-facing interfaces.
- **Intelligent toolchains:** Must assist and enable manual and/or automatic (ML/AI-based) selection of: resources for execution, data movement and transformations, and dynamic optimization for complex applications and workflows.
  - This requires more advanced compilers and supporting technology infrastructure.
  - Better tools for understanding performance profiling, debugging, and the associated state of the system and its components (introspection and real time machine learning mechanisms).
  - Runtime systems must support “smart” scheduling and affinity across in dynamic and diverse application and system architecture environments.
- **Composition**: Software reuse and interoperability will be key to controlling *costs and accelerating time-to-discovery.*
  - The software stack - from languages, tools, and OS/R components - must support effective and efficient composition and execution of different high-level software components and across increasingly heterogeneous hardware.
  - Efforts at the lower-levels of the stack will become key building blocks for coordination and execution of higher level features and domain-aware abstractions.

Managing System Resources Intelligently



- **OS/RTS Design:** Hardware resources will become more complex and diverse. The operating system (OS) and runtime system (RTS) must integrate special-purpose devices and accelerators. The OS cannot assume all resources on a node are identical and dedicated devices
  - OS/RTS must be efficient and sustainable for an increasingly diverse set of hardware components
  - Must provide capability for dynamic discovery of resources as power/energy constraints impose restrictions on availability
- **Decentralized resource management:** New scalable methods of coordinating resources must be developed that allow policy decisions and mechanisms to co-exist throughout the system. Hardware resources are becoming inherently adaptive, making it increasingly complex to understand and evaluate optimal execution and utilization
  - System software must be enhanced to coordinate resources across multiple levels and disparate devices in the system
  - Must leverage cohesive integration of performance introspection and programming system abstractions to provide more adaptive execution
- **Autonomous resource optimization**: Responsibility for efficient use of resources must shift from the user to the system software; must employ sophisticated and intelligent approaches optimize selection of resources to application needs
  - Need more automated methods using machine learning to optimize the performance, energy efficiency, and availability of resources for integrated application workflows
  - More sophisticated usage models beyond batch-scheduled, spaced-shared nodes adds significant complexity to the management of system resources

Modeling & Predicting Performance



Understand, predict, & steer application behavior in EH environments

- **Develop methodologies and tools for accurate and fast modeling of EH at scale**
  - Accurately and quickly explore design space of heterogeneous function units, memories, and interconnects
  - Design methodologies and tools to understand system design tradeoffs in using heterogeneous components, including power, heat, resilience, and overall impact on applications
  - Provide integrated architecture and application modeling
    - Capture realistic workloads
    - Model diverse compute, storage, and interconnect technologies
  - Investigate new methods for model building and runtime cost reduction
    - Stochastic methods, and ML
    - Reduced precision and other approximation techniques
  - **Impact will be transformative**
    - From hardware components to workflow design
    - Predictive and accurate evaluation enabling optimization of new EH systems
- **Cohesively integrate modeling & simulation infrastructure with programming models/environment**
  - Support performance portability across many target EH architectures by providing cost models to aid both application developers and compilers/runtime
  - Invent new techniques to combine non-von (neuromorphic, quantum) with “more Moore”
  - Leverage ML and synthesis of mathematical models to tune compiler/runtime options from high-dimensional search space
  - **Near term impact to steer development/delivery of EH platforms and applications**
  - **Long term impact to workflow and programming environments guided by modsim cost models**

Enabling Reproducible Science

- **Reproducibility: Determination of verifiable computational results faces significant challenges in an increasingly diverse and heterogeneous environment**
  - Different types of processors/accelerators will not produce identical results for the same computations. Past techniques such as bitwise reproducibility are no longer applicable. These issues will be amplified by dynamic application behaviors, asynchronous and non-deterministic execution, and differences in capabilities across systems.
  - Additional challenges arise from non-Von Neumann architectures (e.g. quantum and neuromorphic) and machine learning techniques that can introduce "approximate" solutions.
- **Faults and Errors: The detection and mitigation of pervasive faults and errors becomes more complex with different behaviors and design details of an increasing number of hardware components**
  - Verification at different granularities becomes significant when reproducibility may not be possible or too expensive.
  - Must find ways to build effective and efficient testing strategies in the face of a combinatorial explosion of different realizations and compositions of heterogeneous processors
  - Must develop a more robust understanding of how to evaluate the validity of code and results.

Facilitating Execution of Science Workflows

Data Modeling
Business Intelligence & Reporting
Extract, Transform, and Load
Data Migration
Data Architecture & Structure
Data Analytics

- **Workflow execution:** EH systems and applications bring significant challenges to usability through an unprecedented variety and number of data, resources, and services.
  - Discovery and mapping of science workflow requirements to appropriate data, hardware, and software services
  - New methods of composing scientific workflows (e.g., via workflow motifs)
  - Interfaces that facilitate HW and service composition – programming environments for data services
  - **Ties to Programmer Productivity**
- **Autonomous workflows:** Extracting the highest value from EH systems requires rapid, complex balancing across a wide variety of storage and networking technologies in response to a widely varying workload and significant rate of faults.
  - Profiling workflow telemetric data and learn what and how to instrument these systems
  - Online learning to adapt computation and data organization to available resources and emerging results
  - Transfer learning the knowledge of workflow optimization to other types of workflows and EH systems
  - **Ties to Managing System Resources**
- **(Also) Rapid adaptive analytics:** Data analytic workflows on EH systems must employ new, specialized algorithms and data organizations and facilitate use by domain scientists, not just data scientists.
  - Data and algorithm abstractions for domain scientists (e.g., probabilistic execution vs deterministic execution)
  - New data organizations and specialized robust algorithm developments for specific EH components
  - Important, but out of scope for this specific workshop

Resource Management Issues

- Rethinking virtual memory and translation to support more flexible future accelerator classes and multiple memory and storage types
- Co-design for accelerator offload methodologies
  - At what level do the ISA, accelerator, compiler, runtime, and OS interact?
  - Where are the control-plane boundaries?
- Virtualization and compartmentalization of accelerators
  - What are the potential future and hardware interfaces to accelerators and memory systems?
- What levels of system virtualization are needed for future systems to maximize their ability to maintain flexibility in heterogeneous accelerator composition, maximize portability, and minimize single accelerator lock in?
- Standardizing the exception model for accelerators

Hardware Perspective

- System-on-Chip (SoC)
  - Hardware specialization
  - OS/R needs to be aware of custom hardware capabilities
    - Potentially large collection of hardware capabilities where only a few may be used at a time
  - A single node will not be a single cache-coherent physical address space (true today)
- Photonic interconnects
  - Load/store across a larger domain
  - More intelligent memory controllers
    - Perhaps programmable by OS/R or application
    - Converged with network interface
  - Nodes will look more like racks, racks will look more like systems
- Special-purpose systems will become more general
  - OS will have to be engineered to adapt more easily to new hardware
- Trust model will have to evolve
  - Security model for users and applications likely needs to change
- OS will become much more distributed

Applications Perspective

- Increased complexity
  - Reduce complexity through abstractions, componentization, and composition
  - Decompose applications into tasks and services
  - OS/R will need to provide mechanisms for service discovery and composition
- Access to system services
  - Traps and blocking system calls are already insufficient
  - Convergence between OS and RTS
  - Expose hardware directly to application
- Tools are applications too
  - Tools typically depend more on system services
  - Less human interaction with tools
    - Consumer of diagnostic and debugging information may be the OS or RTS
- Rethink the connections between OS/R and programming environment
- Likely to be event-driven at some level

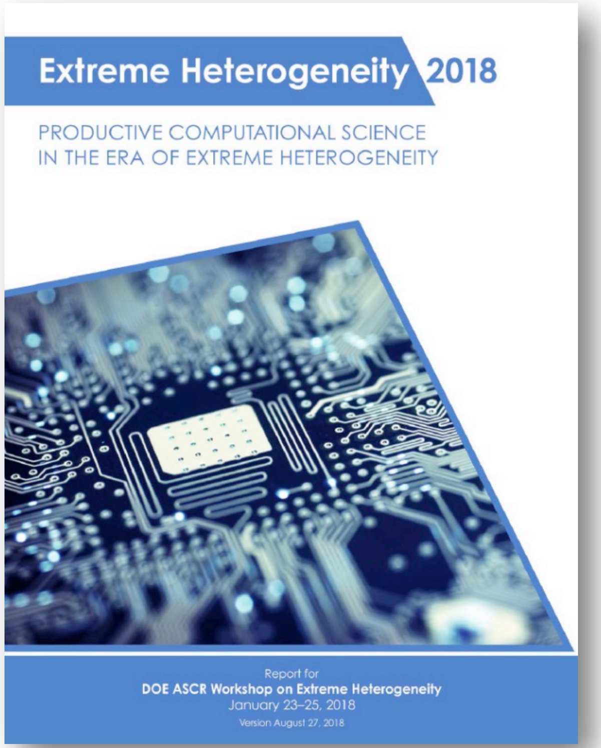
System Usage Model Perspective

- Need to move beyond batch-scheduled, space-shared, non-interactive jobs
  - Dedicated resources versus shared resources
  - More interactivity with users and application services
  - Need to develop a new cost charging model for facilities
- Implicit versus explicit allocation and management of resources
  - Already seeing limitations with explicitly allocating cores, nodes, memory (burst buffers) etc.
  - OS/R will likely need to determine resources implicitly and be elastic
  - Methods for handling resource failures
- Data-centric versus compute-centric view of system
  - Differentiating between HPC and Cloud/BigData approaches
- Support new methods of moving data on and off of the system

Shared Services Perspective

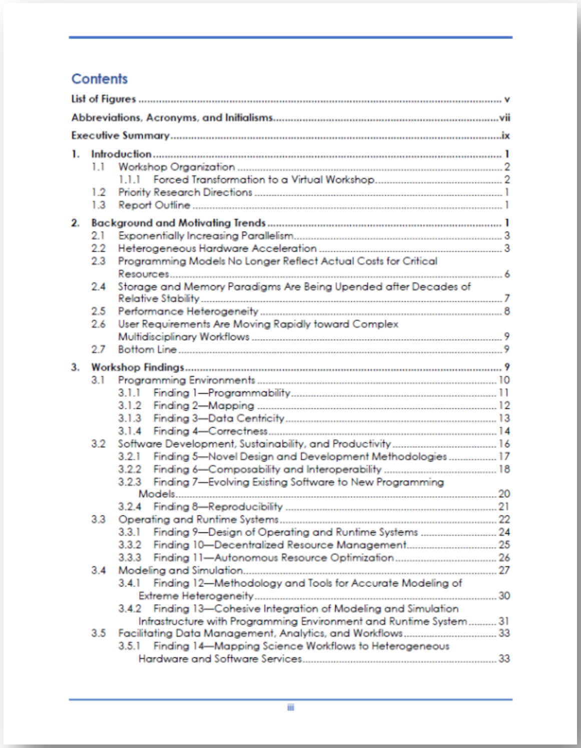
- RAS System (Reliability/Availability/Serviceability)
  - Instrumentation and analysis
  - System health monitoring
    - In-band and/or out-or-band
  - Global Information Bus
- External resources
  - External connectivity to network and storage
  - Streaming data from external instruments
  - New methods of data ingest/egest

DOE/ASCR Extreme Heterogeneity 2018 Report

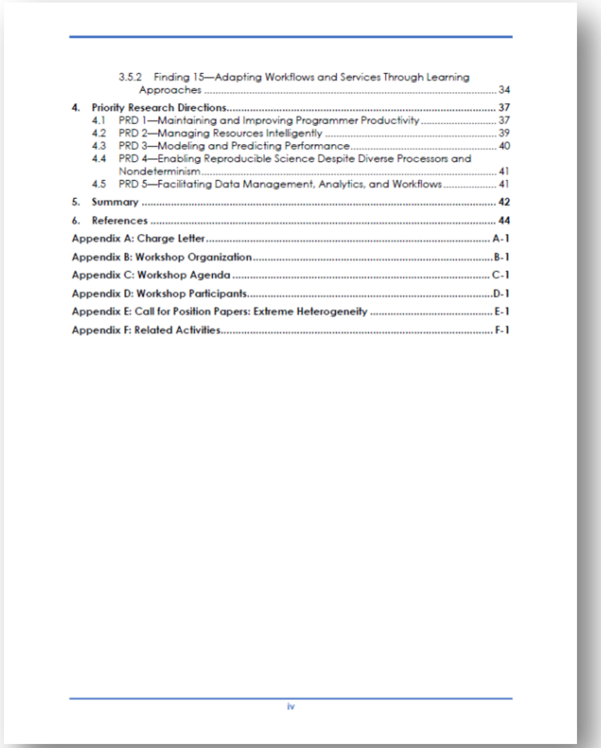

Extreme Heterogeneity 2018

PRODUCTIVE COMPUTATIONAL SCIENCE
IN THE ERA OF EXTREME HETEROGENEITY

Report for
DOE ASCR Workshop on Extreme Heterogeneity
January 23–25, 2018
Version August 27, 2018


Contents

iii



iv

Jeffrey S. Vetter; Katie Antypas; Ron Brightwell; David Donofrio; Maya Gokhale; Travis Humble; Pat McCormick; Rob Ross; Catherine Schuman; John Shalf; et al.

https://orau.gov/exheterogeneity2018/

ASCR In Situ Data Management Workshop

January 28-29, 2019

- Workshop Goal:
  - Identify Priority Research Directions for In Situ Data Management
- Workshop Purpose:
  - Scientific computing will increasingly incorporate a number of different tasks that need to be managed along with the main simulation tasks. For example, this year's SC18 agenda featured in situ infrastructures, in situ analytics, big data analytics, workflows, data intensive science, machine learning, deep learning, and graph analytics—all nontraditional applications unheard of in an HPC conference just a few years ago. Perhaps most surprising, more than half of the 2018 Gordon Bell finalists featured some form of artificial intelligence, deep learning, graph analysis, or experimental data analysis in conjunction with or instead of a single computational model that solves a system of differential equations.
- POC: Laura Biven (Laura.Biven@science.doe.gov)
- 40+ participants: DOE Labs, academia, & industry

https://orau.gov/insitudata2019/

ISDM Workshop Breakout Sessions

- Data Models: Connection and Communication
- Computational Platforms and Environments
- Analysis Algorithms
- Provenance and Reproducibility
- Programming and Execution Models
- Software Architecture for Usability and Sustainability

Draft ISDM Priority Research Directions

- PEMs that support elastic and dynamic use of resources in in situ workflows
  - Resource requirements change throughout the lifetime of a workflow
- Scheduling, mapping, and optimization of PEMs for in situ workflows
  - Mapping to the machine is currently done by hand
  - Automated ways of mapping the machine to the application workload
- Composability for in situ workflows
  - Interoperability and coordination between PEMs
  - Layers can't assume ownership of all resources
- Codesign and use of heterogeneous hardware for ISDM
  - Effectively new or develop hardware (neuromorphic, persistent memory, edge devices, etc.)

How I Spent My SC'18 Friday Morning

- Parallel Applications Workshop, Alternatives to MPI (PAW-ATM)
- BOF on Revisiting the 2008 ExaScale Computing Study and Venturing Predictions for 2028

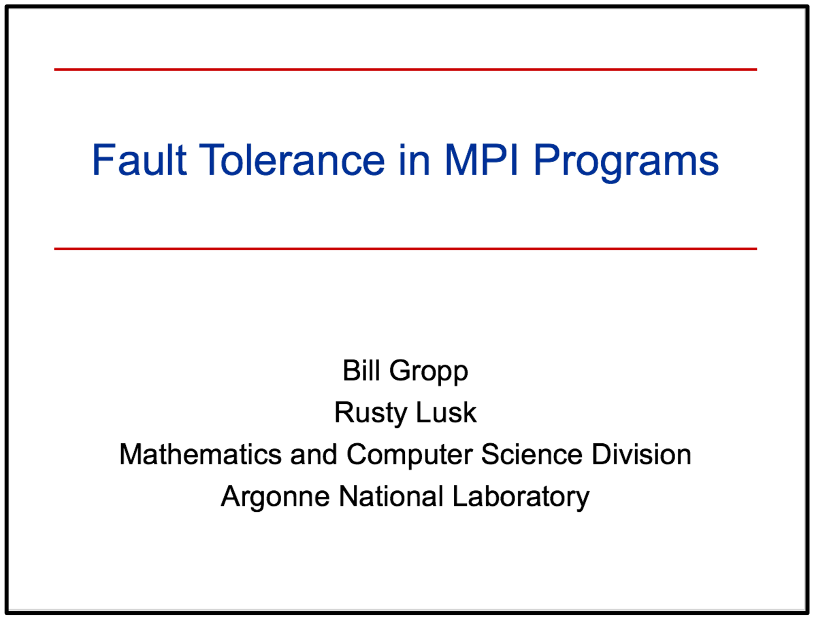
Reasons MPI Won’t be Replaced Anytime Soon

- Too much legacy code and investment
- Application inertia
- Laptop-to-exascale development models precludes specialization
- Overhead of load/store
- Load/store has more fault tolerance issues than MPI
- Load/store hides locality
- Complexity of shared memory (GAS) programming
- PGAS two-level memory abstraction is no different than MPI
- Asynchronous Many Task models need to gain traction in a single address space
- Shift towards higher-level programming abstractions doesn’t embrace load/store
- Any alternative PM will take more than 10 years

Issues With MPI (That I Haven't Mentioned Lately)

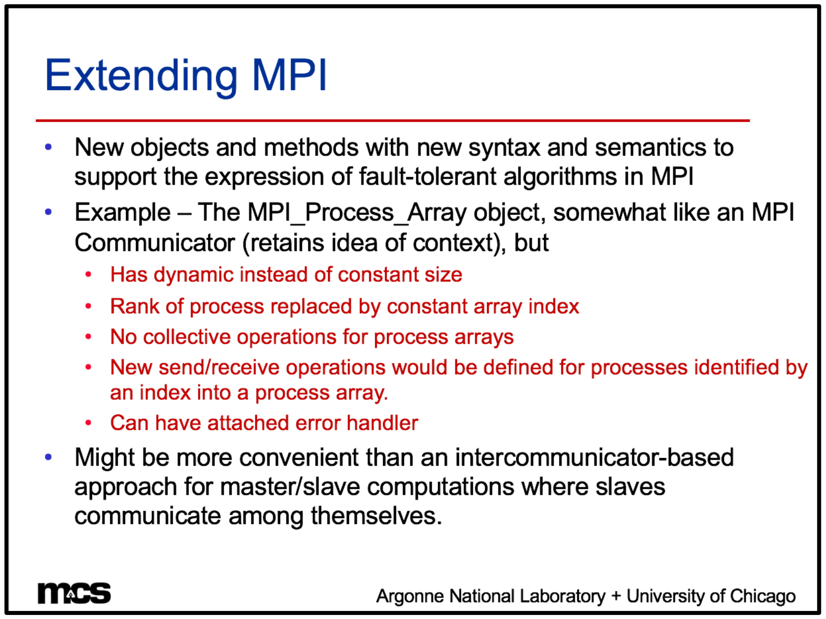
- Too low-level to abstract away complexity
  - Not really intended for domain scientists
- Too high-level to build basic communication services
  - No flexibility in supporting alternative ordering, granularity (packets), reliability (timeouts)
  - Buffering, tag matching, etc.
  - What about MPI_RDMA()?
- Model assumes the destination has an execution context
  - Data movement is between memories, not cores
  - A persistent memory region may not have an active execution context
- Combining point-to-point and collective communication on the same communicator
- Conflating the notion of a group and a communicator
  - It would be easier to join and leave a communicator without notion of a group
  - Safe communication space would still work, for pt2pt would still work
  - Why collectives and pt2pt were separate in the first place

Not a New Idea



Fault Tolerance in MPI Programs

Bill Gropp
Rusty Lusk
Mathematics and Computer Science Division
Argonne National Laboratory



Extending MPI

- New objects and methods with new syntax and semantics to support the expression of fault-tolerant algorithms in MPI
- Example – The MPI_Process_Array object, somewhat like an MPI Communicator (retains idea of context), but
  - Has dynamic instead of constant size
  - Rank of process replaced by constant array index
  - No collective operations for process arrays
  - New send/receive operations would be defined for processes identified by an index into a process array.
  - Can have attached error handler
- Might be more convenient than an intercommunicator-based approach for master/slave computations where slaves communicate among themselves.

mcs Argonne National Laboratory + University of Chicago

Center for Computing Research

Cloud and Enterprise Future Discoveries

- They aren't really doing distributed computing
- All of the hardware is the same
  - No need to dynamically negotiate and discover the capabilities of the other endpoint
- Network security and trust model looks more like HPC systems
  - Virtualization hardware provides isolation
- Eventually applications and libraries will want to reach through the abstraction layers
- There are latency performance and scalability costs to SDN
- There's a difference between parallelism and concurrency
- Network bisection bandwidth can be a critical resource

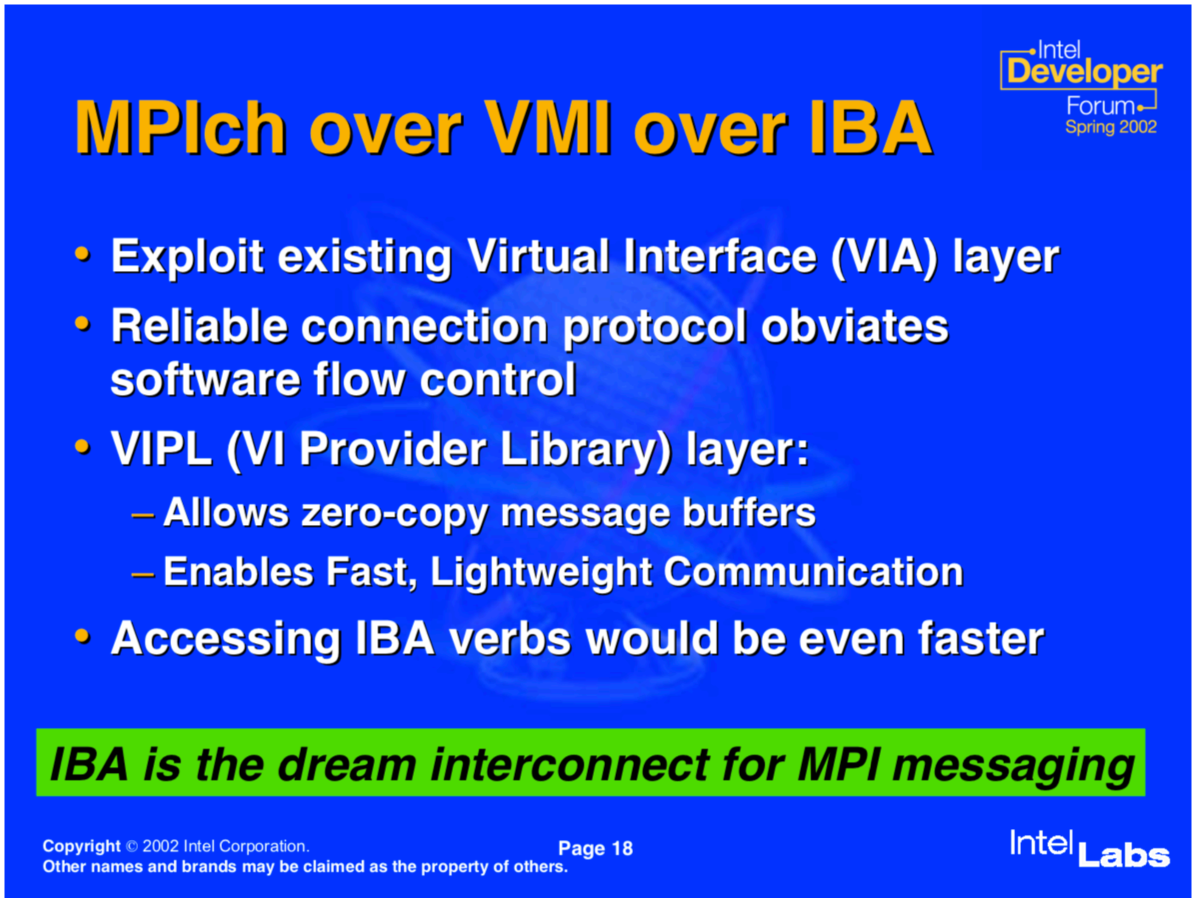
HPC Future Discoveries

- The Ethernet physical layer may actually work for HPC with the right tweaks
- The cost of hardware-based virtualization capabilities is acceptable
- Few HPC application developers really care about high performance
- High-level languages and DSLs in moderation can be effective

Other Thoughts

- The MPI infrastructure has become responsible for more than data movement
  - Startup, locality, placement, topology discovery
  - Even if MPI doesn't directly provide these capabilities
- No time for common practice anymore
  - Everything has to become part of MPI to be used
  - Sustainability issue
  - Application view of MPI needs is different from MPI developer needs
- Complexity in MPI standard and implementations is largely due to complexity of systems
  - Implementation focus has largely been on performance rather than usability
- Need to explicitly separate experimental work from Standard process

Sandia National Laboratories



MPIch over VMI over IBA

Intel Developer Forum Spring 2002

- Exploit existing Virtual Interface (VIA) layer
- Reliable connection protocol obviates software flow control
- VIPL (VI Provider Library) layer:
  - Allows zero-copy message buffers
  - Enables Fast, Lightweight Communication
- Accessing IBA verbs would be even faster

IBA is the dream interconnect for MPI messaging

Copyright © 2002 Intel Corporation.
Other names and brands may be claimed as the property of others.

Page 18

Intel Labs

CCR Center for Computing Research

Questions?