Exascale: Applications, Architectures y Codesign

Andy White
Los Alamos National Laboratory

T = 39,134
Exascale applications and technology have been identified by community.

- Town Hall Meetings April-June 2007
- Scientific Grand Challenges Workshops Nov, 2008 – Oct, 2009
  - Climate Science (11/08),
  - High Energy Physics (12/08),
  - Nuclear Physics (1/09),
  - Fusion Energy (3/09),
  - Nuclear Energy (5/09),
  - Biology (8/09),
  - Material Science and Chemistry (8/09),
  - National Security (10/09)
  - Architecture & Technology (12/2009)
  - Cross-cutting technologies (2/10)
- Exascale Steering Committee
  - “Denver” vendor NDA visits 8/2009
  - SC09 vendor feedback meetings
  - …
- International Exascale Software Project

- Architectures and Technology for Extreme Scale Computing
  - Collaboration and co-design
  - Focus on node software and hardware architecture
  - Managing greater than 1 billion way parallelism
  - Managing errors
Co-design is the key ingredient of the exascale initiative.

Uncertainty Quantification

DUAL
Application #1
Codes, algorithms, models, theory
Application #2
Codes, algorithms, models, theory
Application #3
Codes, algorithms, models, theory

COMMUNITY
Applied mathematics:
Solvers, grids & meshes, PDEs, multi-scale, multi-physics, …

Computer Science:
Programming models, debuggers, performance, OS, file system, …

PROPRIETARY
Focused technology R&D (e.g. memory, optics, storage, …)

Laboratory-Industry Partnership #1
Integrated technology R&D
System acquisition & deployment

May be both classified and unclassified
Success is clearly defined for the exascale initiative.

- Success of the initiative is:
  - Transformational capabilities in national nuclear security, climate, energy and science enabled by predictive exascale simulations
  - U.S. industry leadership in information technology lead by aggressive exascale technology development
  - Competitive advantage for U.S. energy-related and other industries
- Co-design of applications, computational environment and platforms is critical
  - Application teams must have dual responsibility
  - Simulation environment will
    - Be common across all applications and platforms
    - Leverage open source software and product support
  - Long term industry partnerships are essential to success of this 10 year initiative
    - Must leverage and influence the business plan of vendor partners
    - Joint R&D and leveraged community efforts reduce risk
You can run but you can’t hide.

Joseph Louis Barrow

- **System power** is a first class constraint on exascale system performance, effectiveness and TCO.
- **Exascale processor** will have an 100 – 1000x increase in parallelism, design is critical to meet power, performance, price, productivity and predictive goals.
- **System memory** is an important component of meeting exascale power (bandwidth) and applications (storage) goals.
- **Programming model**. Existing programming models will not be effective on nodes developed over the next decade, whether exascale or not. Early investment is critical to provide applications effective access to 2015 system.
- **Reliability and resiliency are very difficult at this** scale and require new checkpoint restart implementation and better understanding of effects and management of errors.
- **Operating System redesign for exascale** is essential for node performance at scale and for efficient support of new programming models and run time systems.
- **HPC co-design strategy and implementation** requires a set of hierarchical performance models and simulators as well as commitment from apps, software and architecture communities.
Swim lanes affect the number of threads that the system needs to support.

There are currently two basic design points for achieving high performance in technical applications. In the future it is expected that these design points may (or may not) become more integrated.

Many-core vs. many-thread machines: stay away from the valley, IEEE 2009
Investments in architecture R&D and application locality are critical.

“The Energy and Power Challenge is the most pervasive … and has its roots in the inability of the [study] group to project any combination of currently mature technologies that will deliver sufficiently powerful systems in any class at the desired levels.”

* DARPA IPTO exascale technology challenge report
Embedded processor example: it’s about architecture and moving data.

![Diagram of Ensemble Processor](image)

**TABLE I**

<table>
<thead>
<tr>
<th>Processor Configuration and Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ensemble Processor</strong></td>
</tr>
<tr>
<td>Technology</td>
</tr>
<tr>
<td>Clock Frequency</td>
</tr>
<tr>
<td>Average Power</td>
</tr>
<tr>
<td>Multipliers</td>
</tr>
<tr>
<td>IRFs</td>
</tr>
<tr>
<td>XRFs</td>
</tr>
<tr>
<td>ORFs</td>
</tr>
<tr>
<td>ARF</td>
</tr>
<tr>
<td>Ensemble Memory</td>
</tr>
</tbody>
</table>

| **RISC Processor**                 |
| Technology                         | TSMC CL013G (V\textsubscript{DD}=1.2V) |
| Clock Frequency                    | 200 MHz |
| Average Power                      | 72 mW |
| Multiplier                         | 16-bit + 40-bit acc. 16.5 pJ/op |
| Register File                      | 40 32-bit registers 17 pJ/read 22 pJ/write |
| Instruction Cache                  | 8KB (2-way) 107 pJ/read 121 pJ/write |
| Data Cache                         | 8KB (2-way) 131 pJ/read 121 pJ/write |

Factors affecting resilience @ exascale

- Smaller circuit sizes, running at lower voltages to reduce power consumption, increases the probability of errors
- Heterogeneous systems make error detection and recovery even harder, for example, error recovery on GPU system will require managing up to 100 threads
- Increasing system and algorithm complexity makes improper interaction of separate components more likely.
- It will cost power, performance and $ to add additional HW detection and recovery logic right on the chips to detect silent errors.

**Number of components**
both memory and processors will increase mean time to failure, interrupt

<table>
<thead>
<tr>
<th></th>
<th>Transient</th>
<th>Persistent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detected</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Undetected</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Uncertainty comes in a variety of shapes and sizes

<table>
<thead>
<tr>
<th></th>
<th>Parametric</th>
<th>Structural</th>
<th>Relational</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory &amp; models</td>
<td>• Calibrated parameters in models</td>
<td>• Unknown effects omitted from models</td>
<td>• Multi-scale, multi-physics effects</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Extrapolation</td>
<td></td>
</tr>
<tr>
<td>Algorithms</td>
<td>• Discretization error</td>
<td>• Extrapolation</td>
<td>• Multiple time scales in operator split algorithms</td>
</tr>
<tr>
<td>Applications code</td>
<td>• Convergence criteria</td>
<td>• Errors in apps code</td>
<td>• Data mapping among different components</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computation and communication</td>
<td>• Rounding errors</td>
<td>• Silent data corruption</td>
<td>• Race conditions among separate components of system</td>
</tr>
<tr>
<td>Operating system &amp; environment</td>
<td>• ECC error rates (chip bit errors)</td>
<td>• System parameters set incorrectly</td>
<td>• System policy mismatch (e.g. memory management)</td>
</tr>
<tr>
<td>Observations &amp; data assimilation</td>
<td>• Statistical variation in experimental data</td>
<td>• Unknown systematic errors in data</td>
<td>• Contextual mismatch of observational and computational data</td>
</tr>
</tbody>
</table>

2/22/2011
Concurrency is one key ingredient in getting to exaflop/sec

and power, resiliency, programming models, memory bandwidth, I/O, …
Programming models and environments require early investment.

- **Barriers:** Delivering a large-scale scientific instrument that is productive and fast.
  - O(1B) way parallelism in Exascale system
    - Maybe 100B threads!
  - O(1K) way parallelism in a processor chip
    - Massive lightweight cores for low power
    - Some “full-feature” cores lead to heterogeneity
  - Data movement costs power and time
  - Software-managed memory (local store)
- **Programming for resilience**
- **Science goals require complex codes**
- **Technology Investments**
  - Extend inter-node models for scalability and resilience, e.g., MPI, PGAS (includes HPCS)
  - Develop intra-node models for concurrency, hierarchy, and heterogeneity by adapting current scientific ones (e.g., OpenMP) or leveraging from other domains (e.g., CUDA, OpenCL)
  - Develop common low level runtime for portability and to enable higher level models
- **Technical Gap:**
  - No portable model for variety of on-chip parallelism methods or new memory hierarchies
  - Goal: Hundreds of applications on the Exascale architecture; Tens running at scale

*From Peter Kogge (on behalf of Exascale Working Group), "Architectural Challenges at the Exascale Frontier", June 20, 2008*
Programming Model Approaches

- Hierarchical approach: intra-node + inter-node
  - Part I: Inter-node model for communicating between nodes
    - MPI scaling to millions of nodes: Importance high; risk low
    - One-sided communication scaling: Importance medium; risk low
  - Part II: Intra-node model for on-chip concurrency
    - Overriding Risk: No single path for node architecture
    - OpenMP, Pthreads: High risk (may not be feasible with node architectures); high payoff (already in some applications)
    - New API, extended PGAS, or CUDA/OpenCL to handle hierarchies of memories and cores: Medium risk (reflects architecture directions); Medium payoff (reprogramming of node code)
- Unified approach: single high level model for entire system
  - High risk; high payoff for new codes, new application domains
System architecture targets are aggressive in schedule and scope.

<table>
<thead>
<tr>
<th>System attributes</th>
<th>2010</th>
<th>“2015”</th>
<th>“2018”</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System peak</strong></td>
<td>2 PF/s</td>
<td>200 Petaflop/sec</td>
<td>≥ 1 Exaflop/sec</td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td>6 MW</td>
<td>15 MW</td>
<td>≤ 20 MW</td>
</tr>
<tr>
<td><strong>System memory</strong></td>
<td>0.3 PB</td>
<td>5 PB</td>
<td>64 PB</td>
</tr>
<tr>
<td><strong>Node performance</strong></td>
<td>125 GF/s</td>
<td>500 GF/s</td>
<td>1 TF/s</td>
</tr>
<tr>
<td><strong>Node memory BW</strong> (consistent with 0.4 B/F)</td>
<td>25 GB/s</td>
<td>200 GB/s</td>
<td>400 GB/s</td>
</tr>
<tr>
<td><strong>Node concurrency</strong></td>
<td>12</td>
<td>100</td>
<td>1,000</td>
</tr>
<tr>
<td><strong>System size</strong> (nodes)</td>
<td>18,700</td>
<td>400,000</td>
<td>1,000,000</td>
</tr>
<tr>
<td><strong>Node link BW</strong> (consistent with 0.1 B/F)</td>
<td>1.5 GB/s</td>
<td>50 GB/sec</td>
<td>100 GB/s</td>
</tr>
<tr>
<td><strong>Mean time before application failure</strong></td>
<td>days</td>
<td>≥ 24 hours</td>
<td>≥ 24 hours</td>
</tr>
<tr>
<td><strong>IO</strong></td>
<td>0.2 TB/s</td>
<td></td>
<td>60 TB/s</td>
</tr>
</tbody>
</table>

2/22/2011
Co-design is essential to manage complexity and optimize results.

Application driven: Find the best technology to run this code. *Sub-optimal*

Application

- Model
- Algorithms
- Code

Technology

- programming model
- operating system
- architecture

Key issues
- Power?
- Performance?
- Price?
- Parallelism?
- Productivity?

Now, we must expand the co-design space to find better solutions:
- new applications & algorithms,
- better technology and performance.

Technology driven: Fit your application to this technology. *Sub-optimal.*
The trade space for exascale is very complex.
The history of high performance computing is the history of DOE HPC.

1976-1980

1992-1996

2008-2012

2024-2028

Exascale
2018
1,000,000 trillion