The NP-ASCR Exascale Meeting

Rusty Lusk
Argonne National Laboratory

James Vary
Iowa State University
Outline

• The DOE Office of Science Exascale Workshops
• The (near) future of high-performance computing
• Summary of workshop findings on nuclear structure and reactions (James, standing in for Steve)
The Exascale Workshop Series

- The DOE Office of Advanced Scientific Computing Research (ASCR) is partnering with other offices in the Office of Science on a series of workshops in extreme scale computing.
  - Climate
  - Nuclear Physics
  - Nuclear Energy
  - High-energy Physics
  - Fusion
  - Materials Science
  - Biology

- The workshops are organized by leaders in the application fields.

- Reports written by application committees

- Web site: http://extremecomputing.labworks.org/index.stm
ASCR High Performance and Leadership Computing Facilities

• NERSC
  – 104 teraflop Cray XT4 with approximately 9,600 dual core processors; **will upgrade to approximately 360 teraflops with quad core in Summer, 2008**
  – 6.7 teraflop IBM Power 5 (Bassi) with 888 processors, 3.5 terabytes aggregate memory
  – **3.1 teraflop LinuxNetworx Opteron cluster (Jacquard) with 712 processors, 2.1 terabytes aggregate memory**

• LCF at Oak Ridge
  – 263 teraflop Cray XT4 (Jaguar) with 7,832 quad core 2.1 GHz AMD Opteron processor nodes, 46 terabytes aggregate memory
  – **18.5 teraflop Cray X1E (Phoenix) with 1,024 multi-streaming vector processors**
  – **Delivery of 1 Petaflop Cray Baker expected in late 2008**

• Argonne LCF
  – **5.7 teraflop IBM Blue Gene/L (BGL) with 2,048 PPC processors**
  – 100 teraflop IBM Blue Gene/P began operations April 1, 2008
  – **446 teraflop IBM Blue Gene/P upgrade accepted in March, 2008 in transition to operations**
Argonne’s IBM Blue Gene/P – 556 TFs
Cray XT5 at ORNL -- 1 Pflop/s

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>XT5</th>
<th>XT4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak Performance</td>
<td>1,645</td>
<td>1,382</td>
<td>263</td>
</tr>
<tr>
<td>AMD Opteron Cores</td>
<td>181,504</td>
<td>150,176</td>
<td>31,328</td>
</tr>
<tr>
<td>System Memory (TB)</td>
<td>362</td>
<td>300</td>
<td>62</td>
</tr>
<tr>
<td>Disk Bandwidth (GB/s)</td>
<td>284</td>
<td>240</td>
<td>44</td>
</tr>
<tr>
<td>Disk Space (TB)</td>
<td>10,750</td>
<td>10,000</td>
<td>750</td>
</tr>
<tr>
<td>Interconnect Bandwidth (TB/s)</td>
<td>532</td>
<td>374</td>
<td>157</td>
</tr>
</tbody>
</table>
Application Alert!

• These may be the last machines usefully programmable with the “traditional” Fortran +MPI programming model.

• Already many applications for these machines use a hybrid programming model consisting of Fortran (or C) plus something else.

• Usually OpenMP because it is most mature.
Traditional Sources of Performance Improvement are Flat-Lining (2004)

- New Constraints
  - 15 years of \textit{exponential} clock rate growth has ended

- Moore’s Law reinterpreted:
  - How do we use all of those transistors to keep performance increasing at historical rates?
  - Industry Response: \#cores per chip doubles every 18 months \textit{instead} of clock frequency!

Figure courtesy of Kunle Olukotun, Lance Hammond, Herb Sutter, and Burton Smith
Multicore comes in a wide variety

- Multiple parallel general-purpose processors (GPPs)
- Multiple application-specific processors (ASPs)

“The Processor is the new Transistor” [Rowen]
What’s Next?

The question is not whether this will happen but whether we are ready.

Source: Jack Dongarra, ISC 2008
Outline of the Situation

• Million core systems and beyond are on the horizon
• Today labs and universities have general purpose systems with 10k-200K cores (BGL@ LLNL 200K, BGP@Argonne 160K, XT5@ORNL 150K cores)
• By 2012 there will be more systems deployed in the 200K-1M core range
• By 2020 there will be systems with perhaps 100M cores
• Personal systems with > 1000 cores within 5
• Personal systems with requirement for 1M threads is not too far fetched (GPUs for example)
E3 Advanced Architectures - Findings

- Exascale systems are likely feasible by 2017±2
- 10-100 Million processing elements (mini-cores) with chips as dense as 1,000 cores per socket, clock rates will grow slowly
- 3D chip packaging likely
- Large-scale optics based interconnects
- 10-100 PB of aggregate memory
- > 10,000’s of I/O channels to 10-100 Exabytes of secondary storage, disk bandwidth to storage ratios not optimal for HPC use
- Hardware and software based fault management
- Simulation and multiple point designs will be required to advance our understanding of the design space
- Achievable performance per watt will likely be the primary metric of progress
Top Technical Challenges

• Power Consumption
  – Proc/mem, I/O, optical, memory, delivery
• Chip-to-Chip Interface Scaling
  – pin/wire count $\Rightarrow$ 3D packaging
• Package-to-Package Interfaces (optics?)
• Fault Tolerance
  – FIT rates and Fault Management
  – Reliability of irregular logic, design practice
• Cost Pressure in Optics and Memory
Looking out to Exascale
Concurrency will be Doubling every 18 months
How Will We Program It?

• Still an unsolved problem
• Some believe a totally new programming model (e.g. X10).
• Some mechanism for dealing with shared memory will be necessary
  – This (whatever it is) plus MPI is the conservative view (my own)
• Whatever it is, it will need to interact properly with MPI
• May also need to deal with on-node heterogeneity
• The situation is somewhat like message-passing before MPI
  – And it is too early to standardize
The Bottom Line

- Levels of concurrency ($10^6 \Rightarrow 10^9$)
- Clock rate of Core (1-4 GHz $\Rightarrow$ 1-4 GHz)
- RAM per Core (1-2GB now to 1-4GB)
- Total Number of cores (200K $\Rightarrow$ 100M)
- Number of cores per node (8 $\Rightarrow$ 64-512)
- Aggressive Fault Management in HW and SW
- I/O channels ($>10^3 \Rightarrow 10^5$)
- Power Consumption (10MW $\Rightarrow$ 40MW-150MW)
- Programming Model (MPI $\Rightarrow$ MPI + X)
Parallel Programming Models: Twenty Years and Counting

• In large-scale scientific computing today essentially all codes are message passing based. Additionally many will use some form of multithreading on SMP or multicore nodes.

• Multicore is challenging programming models but there has not yet emerged a dominate model to augment message passing

• There is a need to identify new hierarchical programming models that will be stable over long term and can support the concurrency doubling pressure
Quasi Mainstream Programming Models

• C, Fortran, C++ and MPI
• OpenMP, pthreads
• (CUDA, RapidMind, Cn) $\rightarrow$ OpenCL
• PGAS (UPC, CAF, Titanium)
• HPCS Languages (Chapel, Fortress, X10)
• HPC Research Languages and Runtime
• HLL (Parallel Matlab, Grid Mathematica, etc.)
Conclusion

• Faster computers than those we have are coming
  – Exascale is within view
• The programming model picture is not yet clear.
• An evolutionary approach is preferred.