Performance Modeling of Extreme-Scale Systems and Applications

Adolfy Hoisie

work with: Kevin Barker, Kei Davis, Darren Kerbyson, Greg Johnson, Mike Lang, Scott Pakin, Fabrizio Petrini

www.c3.lanl.gov/par_arch

Performance and Architecture Lab (PAL)
Computer and Computational Sciences Division
Los Alamos National Laboratory

Work funded by ASC, Office of Science, DARPA

CESC, April 2005, Washington DC
Outline

- Performance modeling
- Comparison of systems
- Performance prediction for advanced architectures
- Summary/Tentative conclusions
Performance and Architecture Lab (PAL)

- Performance/Architecture team at Los Alamos
  - Modeling
  - Measurement, system software R&D, simulation

- Application-centric performance modeling

- Analyze existing systems (or near-to-market systems)
  - e.g. Cray X1, Earth Simulator, BlueGene/L …

- Examine possible future systems
  - e.g. IBM PERCS (DARPA HPCS), BlueGene/P, … ?

- Recent work includes:
  - Modeling and optimization of ASCI Q
  - Comparison of: Cray X1, Earth Simulator, ASCI Q
  - Scalability analysis of BlueGene/L
  - Initial analysis of BlueGene/P

CESC, April 2005, Washington DC
Why Performance Modeling?

- Other performance analysis methods fall short in either accuracy or practicality:
  - **Simulation**
    - Greatest architectural flexibility but takes too long for real applications
  - **Trace-driven experiments**
    - Results often lack generality
  - **Stochastic**
    - System-wide/throughput analysis
  - **Benchmarking**
    - Limited to current implementation of the code
    - Limited to currently-available architectures
    - Difficult to distinguish between real performance and machine idiosyncrasies

*CESC, April 2005, Washington DC*
Performance Modeling Process

- Basic approach:

  "Fundamental Equation of Modeling"
  \[ T_{\text{run}} = T_{\text{computation}} + T_{\text{communication}} - T_{\text{overlap}} \]

  \[ T_{\text{run}} = f (T_{1-\text{CPU}}, \text{Scalability}) \]

  where \( T_{1-\text{CPU}} \) is the single processor time

- We are not using first principles to model single-processor computation time.
  - Rely on measurements for \( T_{1-\text{CPU}} \). May be:
    » time per subgrid,
    » time per cell,
    » calculated using measured rate and # of FLOPS per subgrid

CESC, April 2005, Washington DC
What is a performance model?

- A model encapsulates the understanding of:
  - What resources an application uses during execution (i.e. a functional description)

- This is coupled with a system model:
  - how long it takes to satisfy various resource requirements

- Parameterized:

Execution: Code + System → Performance Prediction

Code Model + System Model

CESC, April 2005, Washington DC
Modeling Successes

- **Codes:**
  - SWEEP3D,
  - SAGE,
  - Partisn,
  - LBMHD,
  - HYCOM
  - MCNP,
  - POP,
  - TYCHO,
  - UMT2K,
  - RAGE (in progress),
  - RF-CTH (in progress),
  - CICE (in progress)

- **Machines:**
  - ASCI Q,
  - ASCI BlueMountain,
  - ASCI White,
  - ASCI Red,
  - CRAY T3E,
  - Earth Simulator,
  - Various clusters (AMD, IA-64, Alphas, Xeon-64, etc),
  - BlueGene/L,
  - BlueGene/P,
  - CRAY X-1,
  - CRAY XD-1,
  - IBM PERCS (early designs)
  - Red Storm
Outline

- Performance modeling
- Comparison of systems
- An experiment-ahead for advanced architectures
- Tentative conclusions
1) Sweep3D

- 3-D spatial grid, partitioned in 2-D
- Pipelined wavefront processing
  - Dependency in ‘sweep’ direction
- Parallel Characteristics:
  - Logical neighbors in X and Y
  - Small message sizes: 100’s bytes (typical)
  - Number of processors determines pipeline length ($P_x + P_y$)

2) SAGE

- Adaptive Mesh Refinement (AMR) hydro-code
- 3 repeated phases
  - Gather data (including processor boundary data)
  - Compute
  - Scatter data (send back results)
- 3-D spatial grid, partitioned in 1-D
- Parallel characteristics
  - Message sizes vary, typically 10 - 100’s Kbytes
  - Distance between neighbors increases with scale

SAGE model: Supercomputing 2001, Denver
BlueGene/Light Overview

- **Node**
  - Dual Core Embedded PowerPC 440
  - 256MB or 512MB memory

- **700MHz (500MHz prototype)**

- **Network**
  - 3-D torus (point-to-point) & Tree network (broadcast, …)

- **4 floating-point per cycle**
  - 2.8 GFlops per processor core

- **Use either 1 PE or 2 PEs per node**

- **Largest system - Lawrence Livermore, 2005 (ASC)**
  - 32 x 32 x 64 nodes (64K nodes, 128K processor cores)
  - Peak performance: 360 Tflops

- **Small physical footprint**
  - 2 nodes per compute card, 16 cards per board, 32 boards per rack
RedStorm Overview

- **Node**
  - Single AMD Opteron
  - 1-8 GB main memory
- **2.0GHz, 2.4GHz**
- **Network**
  - 3-D torus (in two dimensions at Sandia)
  - SeaStar Network processor (uses embedded PowerPC)
- **2 floating-point per cycle**
  - 4 GFlops per processor
- **Largest system - Sandia, 2005 (ASC)**
  - 27 x 16 x 24 nodes (10,368 nodes)
  - Peak performance: 41.5 Tflops
## System Comparison

<table>
<thead>
<tr>
<th></th>
<th>BlueGene/L</th>
<th>RedStorm</th>
<th>ASCI Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Peak</td>
<td>360 Tflops</td>
<td>41.5 Tflops</td>
<td>20 Tflops</td>
</tr>
<tr>
<td>Processor Type</td>
<td>PowerPC 440</td>
<td>AMD Opteron</td>
<td>Alpha EV68</td>
</tr>
<tr>
<td>Node Count</td>
<td>65,536</td>
<td>10,368</td>
<td>2,048</td>
</tr>
<tr>
<td>Processor Count</td>
<td>131,072</td>
<td>10,368</td>
<td>8,192</td>
</tr>
<tr>
<td>Processor Speed</td>
<td>0.7GHz</td>
<td>2.0GHz</td>
<td>1.25GHz</td>
</tr>
<tr>
<td>Processor Peak</td>
<td>2.8 GF/s</td>
<td>4.0 GF/s</td>
<td>2.5 GF/s</td>
</tr>
<tr>
<td>Topology</td>
<td>3D-torus</td>
<td>3D-torus</td>
<td>Fat-tree</td>
</tr>
<tr>
<td>MPI network Latency</td>
<td>3µs</td>
<td>2-3µs</td>
<td>6µs</td>
</tr>
<tr>
<td>MPI network Bandwidth</td>
<td>160MB/s</td>
<td>2.0GB/s</td>
<td>300MB/s</td>
</tr>
</tbody>
</table>

**Notes:**

- BlueGene/L based on Lawrence Livermore installation
- RedStorm based on Sandia installation
- ASCI Q production machine at Los Alamos (2002)

*CESC, April 2005, Washington DC*
Sweep3D (measured on BG/L)

- 3 sets of measurements:
  - 50x50x50 grid per PE (blocked in two different ways)
  - 5x5x400 grid per PE (one blocking regime)

Overall: good scaling (using best blocking)

CESC, April 2005, Washington DC
Sweep3D Model Validation (BG/L)

- Model prediction error:
  - 7.2% (maximum)
  - 1.7% (average)

- 700MHz

- NB: VNM vs. COP mode (using 2 vs 1 PEs per node):
  - Factor of ~1.9x higher performance

CESC, April 2005, Washington DC
Relative performance
BG/L and RedStorm to ASCI Q

- 2 regions in graph:
  - equal processor count (up to 8,192 processors)
  - ASCI Q fixed size (above 8,192 processors)

Equal PE count:
- BG/L is ~0.42x speed of Q
- RedStorm is 1.25x - 2x faster than Q

Full-sized system:
- BG/L is ~5.5x faster than Q
- RedStorm is ~1.85x faster than Q

(5x5x400 sub-grids with best blocking)
SAGE (measured performance BG/L)

- Two inputs:
  - timing: 13,500 cells per PE
  - timing_h: 35,000 cells per PE

**Diagrams:**

- **Timing (500MHz) vs. Node Count**
- **Timing (700MHz) vs. Node Count**
- **Timing_h (500MHz) vs. Node Count**
- **Timing_h (700MHz) vs. Node Count**
SAGE model validation BG/L

- 700MHz
- Model prediction error:
  - 10.1% (maximum)
  - 4.1% (average)
- NB: VNM vs. COP mode (using 2 vs 1 PEs per node):
  - Factor of ~1.1x higher performance

CESC, April 2005, Washington DC
Outline

- Performance modeling
- Applications of modeling: rational system integration, large-scale system comparison, performance prediction, system design
- Performance predictions for advanced architectures
- Tentative conclusions
An Experiment-Ahead for Petaflop Architectures – The “Racehorse”

- Processor roadmap based on Moore’s law
- Network bandwidth based on projected Infiniband for 2008
- Network latency based on current lowest value, with a small assumed improvement over time

<table>
<thead>
<tr>
<th></th>
<th>2004</th>
<th>2006</th>
<th>2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor Peak (Gflop/s)</td>
<td>8</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td>Network Bandwidth (GB/s)</td>
<td>1</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>Network Latency (µs)</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

CESC, April 2005, Washington DC
An Experiment-Ahead for Petaflop Architectures

• What would the Racehorse look like?

<table>
<thead>
<tr>
<th></th>
<th>2004</th>
<th>2006</th>
<th>2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Peak (Pflops)</td>
<td>0.1</td>
<td>0.4</td>
<td>1.0</td>
</tr>
<tr>
<td>Processor Count</td>
<td>12,500</td>
<td>20,000</td>
<td>20,000</td>
</tr>
</tbody>
</table>

• What would the input deck for the app look like?

<table>
<thead>
<tr>
<th></th>
<th>2004</th>
<th>2006</th>
<th>2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total problem size</td>
<td>$2 \times 10^8$</td>
<td>$5 \times 10^8$</td>
<td>$1 \times 10^9$</td>
</tr>
<tr>
<td>cells size per processor</td>
<td>16,000</td>
<td>25,000</td>
<td>50,000</td>
</tr>
</tbody>
</table>
An Experiment-Ahead for Petaflop Architectures

- Use expected system characteristics and problem sizes
- Use PAL’s performance models
- Provide expected performance improvements over time
- Assume we will get same % of peak on a single CPU
Performance improvements relative to ASCI Q

- SN Transport (Sweep3D)
- Hydro (SAGE)

Year:
- 2004: 100Tf
- 2006: 400Tf
- 2008: 1Pf
Performance improvements relative to ASCI Q

• Workload assuming 80% Sn transport and 20% Hydro

![Performance Improvement Graph](image-url)
# A specific petaflop architecture

<table>
<thead>
<tr>
<th>System</th>
<th>ASCI Q</th>
<th>BlueGene/L</th>
<th>BlueGene/P*</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Peak Performance</td>
<td>20 Tflops</td>
<td>360 Tflops</td>
<td>1 Pflops</td>
</tr>
<tr>
<td>Processor Type</td>
<td>Alpha EV68-CD</td>
<td>PowerPC 440</td>
<td>PowerPC 440</td>
</tr>
<tr>
<td>Node Count</td>
<td>2,048</td>
<td>65,536</td>
<td>65,536</td>
</tr>
<tr>
<td>Processor Count</td>
<td>8,192</td>
<td>131,072</td>
<td>262,144</td>
</tr>
<tr>
<td>Processor Speed</td>
<td>1.25GHz</td>
<td>0.7GHz</td>
<td>1.0GHz</td>
</tr>
<tr>
<td>Nodes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Processors per Node</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Peak Floating-Point rate</td>
<td>10 GF/s/node</td>
<td>5.6 GF/s/node</td>
<td>16 GF/s/node</td>
</tr>
<tr>
<td>Cache</td>
<td>8MB (L2 / PE)</td>
<td>4MB (L3 / node)</td>
<td>8MB (L3 / node)</td>
</tr>
<tr>
<td>Main Memory (node)</td>
<td>8-16 GB</td>
<td>256-512MB</td>
<td>0.5-4 GB</td>
</tr>
<tr>
<td>Network</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Topology</td>
<td>Fat-tree</td>
<td>3D-torus</td>
<td>3D-torus</td>
</tr>
<tr>
<td>MPI network Latency</td>
<td>6µs</td>
<td>3µs</td>
<td>1µs</td>
</tr>
<tr>
<td>MPI network Bandwidth</td>
<td>300MB/s</td>
<td>160MB/s</td>
<td>500MB/s</td>
</tr>
</tbody>
</table>

* BG/P parameters shown with permission from IBM. The system characteristics are likely to evolve.

CESC, April 2005, Washington DC
BG/P Performance characteristics (assumed)

- No hardware available for measurement
  - Use assumed hardware performance characteristics

  network latency: 1.0 µs
  network bandwidth: 500 MB/s
  achievable processor speed: 3.75% of peak (single node)

- Optimistic view

- 2 types of analysis:
  - 1) Sensitivity analysis
  - 2) Expected performance

- Sweep3D: 10K cells per processor (5x5x400 sub-grid size)
Expected Performance improvement

Sweep3D

- Comparison with ASCI Q - 2 regions in graph
  - Equal processor count (up to 8,192 processors)
  - ASCI Q fixed size (above 8192 processors)

Up to 8192 PEs:
- BG/L is $\sim0.42x$ speed of Q
- BG/P is $\sim0.62x$ speed of Q

Full-sized system:
- BG/L is $5.3x$ faster than Q
- BG/P is $16.7x$ faster than Q
Expected Performance improvement

SAGE

- Performance depends on the problem, distance between communicating neighbors, and size of each dimension on the 3-D torus
- Mapping assumed is a default one (may be pessimistic)
- Kinks in the curve are a result of this mapping with changing PE count

CESC, April 2005, Washington DC
Tentative conclusions

- 80% Sweep + 20% Sage: BG/P would be faster by a factor of 18 on the full BG/P configuration – the Racehorse was 19 X…
- We ended up in the same place!
- Is it a new law of nature?
- We are chasing the technology (the Racehorse) or we are throwing more (less powerful) resources at the problem.
- The 2 approaches are very similar in nature – and the interplay between surface-to-volume and resources just happened to end up at the same point
- We are falling behind in efficiency
- Did not consider important issues such as system software, power, footprint, etc
Summary

- Accurate performance models of entire applications developed by PAL based on novel methodology
- Multiple uses of modeling in the system and application areas
- Models can be used to predict performance on future architectures, and conversely, design future systems for a workload of interest
- Performance analysis of BG/L and RedStorm, compared against Q
- Point design studies for a hypothetical and a specific petaflop architecture
- Disclaimer: other applications will result in different performance!
Acknowledgements

- Thanks to:
  - IBM BlueGene Team, especially:
    » George Almasi, Manish Gupta, Jose Moreira, Bob Walkup
  - Sandia RedStorm Team, especially:
    » Jim Tomkins, Sue Kelly
  - US Department of Energy through Los Alamos National Laboratory contract W-7405-ENG-36
  - Work funded by the ASC program
The End