A Performance Tools Strategy for Petascale Systems

Subhash Saini

NASA Ames Research Center, Moffett Field, California

The International Workshop on Performance Analysis and Optimization of High-End Computing Systems

Supercomputing 2006

November 12, 2006 Tampa, Florida
Outline

- **Performance of existing HEC systems**
  - Computing subsystem
  - Memory subsystem
  - I/O subsystem
  - Interconnect subsystem

- **Performance Tools**
  - Existing tools
  - Future tools

- **Software tools**
  - Compilers, math libraries, debuggers, programming languages

- **Summary**
Workshop Panel Theme

- The push towards petascale computing platforms is challenging fundamental views towards the programming, operation, and performance evaluation of systems and applications of massive scale.
- There have been significant advances in the development of performance tools during the last several years of terascale computing.
- Will the methods and techniques embodied in the tools carry forward to the next-generation of HEC, or is there a need for a paradigm shift in approach?
## NSF-DOD Petaflop Benchmarks

<table>
<thead>
<tr>
<th>Code</th>
<th>WRF</th>
<th>OOCORE</th>
<th>GAMESS</th>
<th>MILC</th>
<th>PARATEC</th>
<th>HOMME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Climate</td>
<td>Math/EM</td>
<td>Chemistry</td>
<td>QCD</td>
<td>Materials</td>
<td>Climate</td>
</tr>
<tr>
<td>Source code size (zipped tar)</td>
<td>4 GB</td>
<td>453 KB</td>
<td>76 MB</td>
<td>500 KB</td>
<td>601 KB</td>
<td>2.3 MB</td>
</tr>
<tr>
<td>Lines of code*</td>
<td>large</td>
<td>medium</td>
<td>X large</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>Language</td>
<td>Fortran + some C</td>
<td>Fortran + some C</td>
<td>Fortran + some C</td>
<td>C</td>
<td>Fortran</td>
<td>Fortran + some C</td>
</tr>
<tr>
<td>Parallelism</td>
<td>MPI + optional OpenMP</td>
<td>MPI</td>
<td>Sockets, MPI, or shmem</td>
<td>MPI</td>
<td>MPI</td>
<td>MPI and/or OpenMP</td>
</tr>
<tr>
<td>Libraries</td>
<td>NetCDF</td>
<td>ScaLAPACK, BLACS, BLAS</td>
<td>BLAS</td>
<td>none</td>
<td>ScaLAPACK, BLACS, BLAS</td>
<td>NetCDF, PNetCDF, or none</td>
</tr>
<tr>
<td>DoD HPCMP TI-06</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

* medium: 40k – 60k lines, large: 100k – 200k lines, X large: 500k – 1M lines
## NSF-DOD Petaflop Benchmarks

<table>
<thead>
<tr>
<th>Application</th>
<th>Problem Size</th>
<th>CPU Count</th>
<th>Wall-time (sec)</th>
<th>Total Flop Count (TFLOP)</th>
<th>Total rate (GFLOPS)</th>
<th>Per CPU rate (MFLOPS)</th>
<th>Percentage of peak per CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRF</td>
<td>standard</td>
<td>64</td>
<td>1324</td>
<td>83.9</td>
<td>63.3</td>
<td>989.8</td>
<td><strong>15.5</strong></td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>256</td>
<td>5499</td>
<td>1314.0</td>
<td>239.0</td>
<td>933.4</td>
<td><strong>14.6</strong></td>
</tr>
<tr>
<td>OOCORE</td>
<td>standard</td>
<td>64</td>
<td>912</td>
<td>178.7</td>
<td>195.9</td>
<td>3061.6</td>
<td><strong>47.8</strong></td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>256</td>
<td>2101</td>
<td>921.7</td>
<td>438.7</td>
<td>1713.7</td>
<td><strong>26.8</strong></td>
</tr>
<tr>
<td>GAMESS</td>
<td>standard</td>
<td>64</td>
<td>4504</td>
<td>64.7</td>
<td>14.4</td>
<td>224.5</td>
<td><strong>3.5</strong></td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>256</td>
<td>5253</td>
<td>579.3</td>
<td>110.3</td>
<td>430.8</td>
<td><strong>6.7</strong></td>
</tr>
<tr>
<td>MILC</td>
<td>medium</td>
<td>64</td>
<td>401</td>
<td>10.3</td>
<td>25.7</td>
<td>401.3</td>
<td><strong>6.3</strong></td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>256</td>
<td>3484</td>
<td>303.7</td>
<td>87.2</td>
<td>340.5</td>
<td><strong>5.3</strong></td>
</tr>
<tr>
<td>PARATEC</td>
<td>medium</td>
<td>64</td>
<td>567</td>
<td>128.9</td>
<td>227.5</td>
<td>3554.5</td>
<td><strong>55.5</strong></td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>256</td>
<td>1125</td>
<td>717.5</td>
<td>637.9</td>
<td>2491.9</td>
<td><strong>38.9</strong></td>
</tr>
<tr>
<td>HOMME</td>
<td>standard</td>
<td>384</td>
<td>16134</td>
<td>4960.0</td>
<td>307.4</td>
<td>800.6</td>
<td><strong>12.5</strong></td>
</tr>
</tbody>
</table>

**Mean** 21%
**Geometric Mean** 15%
**Harmonic Mean** 10%
LINPACK on Dual Core Systems

HPCC Benchmark G-HPL

Tflop/s

SGI Altix  AMD Opteron  Intel Xeon

Four processors  Eight processors

November 11, 2006  Subhash Saini - NASA Ames
Memory Bandwidth Dual Core Systems

![Bar chart showing memory bandwidth for different systems: SGI Altix, AMD Opteron, Intel Xeon with four and eight processors.](chart.png)
HPCC G-FFT on Dual Core Systems

![Bar chart showing performance of SGI Altix, AMD Opteron, and Intel Xeon processors with 'Four processors' and 'Eight processors' configurations.]
OpenMP on Dual Core Systems on Linux and MAC Operating Systems

Benchmark NPB 3.2 FT Class B

![Graph showing performance comparison of different processors](image_url)
OVERFLOW-2 on Dual Core Systems

![Bar Chart: NASA Application OVERFLOW-2 on 8 Processors]

- **Time (Seconds)**: Y-axis
- **Processes**:
  - **Green**: AMD Opteron Dual Core
  - **Blue**: Intel Xeon Dual Core
  - **Red**: SGI Altix BX2

The chart compares the performance of different dual core processors on NASA Application OVERFLOW-2 across 8 processors.
FLASH I/O Bandwidth for Case 16 x 16 x 16

I/O Bandwidth for Case 16 x 16 x 16

I/O Bandwidth (MB/s)

- Checkpoint
- Plot without corners
- Plot with corners

Number of Processors

0 64 128 192 256 320 384 448 512
NSF-DOD SPIO Benchmark on Altix and SX-8

Bandwidth of SPIO Benchmark

Bandwidth (MiB/s)

Number of Processes

Altix Read
Altix Write
SX-8 Read
SX-8 Write
Automatic Performance Analysis

- Traditional approach is off-line with strong involvement of the user with the following five phases
  - Program instrumentation before execution
  - Measurement of predefined specific events during execution
  - Post-mortem and user-controlled analysis of the performance data
  - Presentation of the data by textual and graphical tools
  - Optimization of the program and its data structures under control of the user

For petascale systems, overall control of the above approach needs a fundamental redesign and generalization for the following reasons ...
Reasons for Fundamental Redesign

- Amount of performance data produced by program monitoring for trace files can exceed capacity of the file system.
- User can no longer be involved in all aspects of performance analysis and tuning.
- Most traditional tools for visual presentation results do not scale in an obvious way to a large number of components in petascale systems.
- Applications exhibiting adaptive behavior, off-line approach cannot handle critical events that arise during execution time and need immediate attention and action, such as algorithm changes, data restructuring and/or necessary modifications of the monitoring and analysis strategies.
Summary

- More progress in hardware to achieve peak petaflop performance at the expense of software
- Sustained percentage performance on real applications per processor is less than 10%
- Multi-core core systems introduce more parallelism at the hardware level
- Will require a new strategy to redesign of applications, math libraries, and scalable algorithms to reach the level of hardware parallelism to fully utilize a petascale class systems.
- Fault tolerant systems
- New approach to check-pointing for large systems