Evolution of the Cray Performance Measurement and Analysis Tools

Heidi Poxon
Manager & Technical Lead, Performance Tools
Cray Inc.
Outline

- Evolutionary path of the Cray performance tools
- Characteristics of next generation systems
- Recent enhancements
- What’s coming next
- A peek at something new
Future system basic characteristics:
- Many-core, hybrid multi-core computing
- Increase in on-node concurrency
  - 10s-100s of cores sharing memory
  - With or without a companion accelerator
  - Vector hardware at the low level

Impact on applications:
- Restructure / evolve applications while using existing programming models to take advantage of increased concurrency
- Expand on use of mixed-mode programming models (MPI + OpenMP + accelerated kernels, etc.)
Evolutionary Path of the Cray Performance Tools

- Focus on automation (simplify tool usage, provide feedback based on analysis)
- Enhance support for multiple programming models within a program (MPI, PGAS, OpenMP, SHMEM)
- Scaling (larger jobs, more data, better tool response)
- New processors and interconnects
- Extend performance tools to include pre-runtime information from the Cray compiler
Recent Enhancements

- Latest release: CPMAT 5.2.0 (April 28, 2011)

- Usability
  
  - Combined CrayPat and Cray Apprentice2 license and package
  
  - FLEXlm license
  
  - New perftools modulefile
  
  - pat_report tables available in Cray Apprentice2
Example of *pat_report* Tables in Cray Apprentice2

New text table icon

Right click for table generation options
Recent Enhancements

- Programming models and languages
  - New predefined wrappers (ADIOS, ARMCI, PetSc, PGAS libraries)
  - Access to Gemini network counters
  - More UPC and Co-array Fortran support
  - Support for non-record locking file systems
  - Support for applications built with shared libraries
  - Support for Chapel programs
<table>
<thead>
<tr>
<th>Samp %</th>
<th>Samp</th>
<th>Imb.</th>
<th>Imb.</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PE='HIDE'</td>
</tr>
<tr>
<td>100.0%</td>
<td>77</td>
<td>--</td>
<td>--</td>
<td>Total</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>94.8%</td>
<td>73</td>
<td>--</td>
<td>--</td>
<td>ETC</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>20.8%</td>
<td>16</td>
<td>15.06</td>
<td>50.2%</td>
<td>syscall</td>
</tr>
<tr>
<td>14.3%</td>
<td>11</td>
<td>15.81</td>
<td>60.5%</td>
<td>__pgas_barrier_wait_all</td>
</tr>
<tr>
<td>11.7%</td>
<td>9</td>
<td>7.28</td>
<td>47.0%</td>
<td>__pat_tracing_ea_ptr_by_name_set_addr</td>
</tr>
<tr>
<td>3.9%</td>
<td>3</td>
<td>3.75</td>
<td>55.3%</td>
<td>__pat_thread_get</td>
</tr>
<tr>
<td>3.9%</td>
<td>3</td>
<td>5.00</td>
<td>64.5%</td>
<td>__pgas_barrier_notify_pe</td>
</tr>
<tr>
<td>3.9%</td>
<td>3</td>
<td>19.22</td>
<td>90.2%</td>
<td>__pgas_barrier_wait_children</td>
</tr>
<tr>
<td>3.9%</td>
<td>3</td>
<td>5.88</td>
<td>67.4%</td>
<td>__pgas_sync_nbi</td>
</tr>
<tr>
<td>2.6%</td>
<td>2</td>
<td>4.09</td>
<td>70.4%</td>
<td>__pgas_aand</td>
</tr>
<tr>
<td>2.6%</td>
<td>2</td>
<td>1.84</td>
<td>47.6%</td>
<td>__pgas_barrier</td>
</tr>
</tbody>
</table>

...
### Table 1: Profile by Function

<table>
<thead>
<tr>
<th>Samp %</th>
<th>Samp %</th>
<th>Imb.</th>
<th>Imb.</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>100.0%</td>
<td>7</td>
<td>--</td>
<td>--</td>
<td>Total</td>
</tr>
<tr>
<td>71.4%</td>
<td>5</td>
<td>--</td>
<td>--</td>
<td>USER</td>
</tr>
<tr>
<td>57.1%</td>
<td>4</td>
<td>0.25</td>
<td>8.3%</td>
<td>mpp_broadcast</td>
</tr>
<tr>
<td>14.3%</td>
<td>1</td>
<td>0.50</td>
<td>66.7%</td>
<td>mpp_alloc</td>
</tr>
<tr>
<td>28.6%</td>
<td>2</td>
<td>--</td>
<td>--</td>
<td>ETC</td>
</tr>
<tr>
<td>28.6%</td>
<td>2</td>
<td>0.50</td>
<td>33.3%</td>
<td>bzero</td>
</tr>
</tbody>
</table>
Recent Enhancements (cont’d)

- **Scalability**

  - New .ap2 data format and client / server model
    - Reduced pat_report processing and report generation times
    - Reduced app2 data load times
    - Graphical presentation handled locally (not passed through ssh connection)
    - Better tool responsiveness
    - Minimizes data loaded into memory at any given time
    - Reduced server footprint on Cray XT/XE service node
    - Larger jobs supported

  - Distributed Cray Apprentice2 (app2) client for Linux
    - app2 client for Mac and Windows laptops coming later this year
Scalable Data Format Reduced Processing Times

- **CPMD**
  - MPI, instrumented with `pat_build -u`, HWPC=1
  - 960 cores

<table>
<thead>
<tr>
<th></th>
<th>Perftools 5.1.3</th>
<th>Perftools 5.2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>.xf -&gt; .ap2</td>
<td>88.5 seconds</td>
<td>22.9 seconds</td>
</tr>
<tr>
<td>ap2 -&gt; report</td>
<td>1512.27 seconds</td>
<td>49.6 seconds</td>
</tr>
</tbody>
</table>

- **VASP**
  - MPI, instrumented with `pat_build -gmpi -u`, HWPC=3
  - 768 cores

<table>
<thead>
<tr>
<th></th>
<th>Perftools 5.1.3</th>
<th>Perftools 5.2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>.xf -&gt; .ap2</td>
<td>45.2 seconds</td>
<td>15.9 seconds</td>
</tr>
<tr>
<td>ap2 -&gt; report</td>
<td>796.9 seconds</td>
<td>28.0 seconds</td>
</tr>
</tbody>
</table>
- Log into Cray XT login node
  \% ssh -Y seal

- Launch Cray Apprentice2 on Cray XT login node
  \% app2 /lus/scratch/mydir/my_program.ap2
  - User Interface displayed on desktop via ssh trusted X11 forwarding
  - Entire my_program.ap2 file loaded into memory on XT login node
    (can be Gbytes of data)
Launch Cray Apprentice2 on desktop, point to data

% app2 seal:/lus/scratch/mydir/my_program.ap2

- User Interface displayed on desktop via X Windows-based software
- Minimal subset of data from my_program.ap2 loaded into memory on Cray XT/XE service node at any given time
- Only data requested sent from server to client
What’s Coming Next

- Move from perfmon2 to Linux perf_events subsystem for access to hardware performance counters

- Support for Interlagos
  - Core Power Boost (CPB), Interlagos hardware counter events

- Support for Cray XK6 systems

- Analysis and hints
  - Automatic grid detection
  - Hardware counter thresholds
  - Memory traffic outliers
Table 3: Time and Bytes Transferred for Accelerator Regions

<table>
<thead>
<tr>
<th>Host Time</th>
<th>Acc Time</th>
<th>Acc Copy</th>
<th>Acc Copy</th>
<th>Calls</th>
<th>Group='ACCELERATOR'</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time %</td>
<td>In (MB)</td>
<td>Out (MB)</td>
<td>PE=0</td>
<td></td>
<td>Thread=0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Calltree</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Function</td>
</tr>
</tbody>
</table>

| Total     | 100.0%   | 14.84495 | 13.615016 | 14550.536 | 10461.216 | 1777 |

| mg_       | 93.7%    | 13.909414| 12.418942 | 13274.781 | 9675.075  | 1777 |
|-----------|----------|----------|-----------|----------|-----------|
| mg3p_     | 51.8%    | 7.692439 | 7.645484  | 7902.816 | 6399.489  | 1630 |
| resid_    | 21.7%    | 3.229140 | 3.216513  | 3758.31  | 2254.986  | 420  |
| resid_    | 11.9%    | 1.767674 | 1.763377  | 2254.986 | 751.662   | 140  |
| resid_.ASYNC_COPY@li.459 | 7.8% | 1.158744 | 1.158958 | 2254.986 | 0.000 | 35 |
| resid_.ASYNC_COPY@li.492 | 4.1% | 0.604365 | 0.337742 | 0.000 | 751.662 | 35 |
| resid_.SYNC_WAIT@li.492 | 0.0% | 0.003903 | 0.000000 | 0.000 | 0.000 | 35 |
| resid_.ASYNC_KERNEL@li.459 | 0.0% | 0.000662 | 0.266677 | 0.000 | 0.000 | 35 |
New code restructuring and analysis assistant…

- Presents **annotated source code** with compiler optimization information ("loopmark on wheels")

- Offers **source code navigation** based on performance data collected through CrayPat

- Provides infrastructure for user to investigate high level looping structures for parallelization

- Highlights loops that could not be optimized

- Presents **feedback on critical dependencies** that prevent optimizations
Restructuring and Analysis Assistant

Line 66:
Loop unrolled 2 times.
Loop interchanged with loop at line 67.

66 DO 200 I=1,M

67 DO 200 J=js,je

68 UNEW(I+1,J) = UOLD(I+1,J)+

69 1 TDT8*(Z(I+1,J+1)+Z(I+1,J))*CV(I+1,J+1)+CV

70 2 +CV(I+1,J)) TDTDX*(H(I+1,J)-H(I,J))

71 if(j.gt.1)then

72 VNEW(I,J) = VOLD(I,J)-TDT8*(Z(I+1,J)+Z(I,J))

73 1 *(CU(I+1,J)+CU(I,J)+CU(I,J-1)+CU(I,J+1))

74 2 -TDTDSY*(H(I,J)-H(I,J-1))

75 endif

76 if(j.eq.n)then

77 VNEW(I,J+1) = VOLD(I,J+1)-TDT8*(Z(I+1,J+1)+Z(I,J+1))

78 1 *(CU(I+1,J+1)+CU(I,J+1)+CU(I,J)+CU(I+1,J))

79 2 -TDTDSY*(H(I,J+1)-H(I,J))

80 endif

81 PNEW(I,J) = POLD(I,J)-TDTDX*(CU(I+1,J)-CU(I,J))

82 1 -TDTDSY*(CV(I,J+1)-CV(I,J))

83 200 CONTINUE

84

85 CME--

86 C
Summary

Performance tools vision:

Evolve the current set of performance measurement and analysis tools to be part of a more tightly coupled programming environment solution with compilers, libraries, and tools that will help users port and optimize applications for many-core or hybrid multi-core computing.
Evolution of the Cray Performance Measurement and Analysis Tools

Questions / Comments
Thank You!