Performance Measurement and Visualization on the Cray XT

Luiz DeRose
Programming Environment Director
Cray Inc.
ldr@cray.com

CSCS
July 13-15, 2009

The Cray Tools Strategy

- Must be easy and flexible to use
  - Automatic program instrumentation
    - no source code or makefile modification needed
- Integrated performance tools solution
  - Multiple platforms
  - Multiple functionality
    - Measurements of user functions, MPI, I/O, memory, & math SW
    - HW Counters support
- Close interaction with user for feedback targeting functionality enhancements

Cray Performance Analysis Infrastructure

- CrayPat
  - pat_build: Utility for application instrumentation
    - No source code modification required
  - run-time library for measurements
    - transparent to the user
  - pat_report:
    - Performance reports
    - Performance visualization file
  - pat_help
    - Interactive performance tool help utility
- Cray Apprentice^2
  - Graphical performance analysis and visualization tool
  - Can be used off-line on Linux system

Performance Data Collection

- Two dimensions
  - When performance collection is triggered
    - Externally (asynchronous)
      - Sampling
        - » Timer interrupt
        - » Hardware counters overflow
    - Internally (synchronous)
      - Code instrumentation
        - » Event based
        - » Automatic or manual instrumentation
  - How performance data is recorded
    - Profile ::= Summation of events over time
      - run time summarization (functions, call sites, loops, …)
    - Trace file ::= Sequence of events over time

Performance Analysis with Cray Tools

- Important performance statistics:
  - Top time consuming routines
  - Load balance across computing resources
  - Communication overhead
  - Cache utilization
  - FLOPS
  - Vectorization (SSE instructions)
  - Ratio of computation versus communication

Application Instrumentation with pat_build

- No source code or makefile modification required
  - Automatic instrumentation at group (function) level
    - Groups: mpi, io, heap, math SW, …
- Performs link-time instrumentation
  - Requires object files
  - Instruments optimized code
  - Generates stand-alone instrumented program
  - Preserves original binary
  - Supports sample-based and event-based instrumentation
- pat_build [-d dirfile] [-D directive] [-f] [-g tracegroup] [-n] [-O ofile] [-o instr_program] [-t tracefile] [-T tracefunc] [-u] [-V] [-v] [-w] [-z] program [instr_program]

-g tracegroup

- biolibs — Cray Bioinformatics library routines
- blas — Basic Linear Algebra subprograms
- heap — dynamic heap
- io — includes stdio and sysio groups
- lapack — Linear Algebra Package
- math — ANSI math
- mpi — MPI
- omp — OpenMP API
- omp-rtl — OpenMP runtime library (not supported on Catamount)
- pthreads — POSIX threads (not supported on Catamount)
- shmem — SHMEM
- stdio — all library functions that accept or return the FILE* construct
- sysio — I/O system calls
- system — system calls

Running the Instrumented Application

- MUST run on Lustre (/work/…)
- Runtime environment variables
  - Optional timeline view of program available
    - export PAT_RT_SUMMARY=0
    - View trace file with Cray Apprentice^2
  - Number of files used to store raw data:
    - 1 file created for program with 1 – 256 processes
    - $\sqrt{n}$ files created for program with 257 – n processes
    - Ability to customize with PAT_RT_EXPFILE_MAX
  - Request hardware performance counter information:
    - export PAT_RT_HWPC=<HWPC Group>
    - Can specify events or predefined groups

pat_report

- Performs data conversion
  - Combines information from binary with raw performance data
- Generates text report of performance results
- Formats data for input into Cray Apprentice^2
  - pat_report [-V] [-i dir|instrprog] [-o output_file] [-O keyword] [-C 'table_caption'] [-d d-opts] [-b b-opts] [-s key=value] [-H] [-P] [-T] [-z] data_directory | data_file.xf

What is the ".ap2" File?

- The ".ap2" file is a self contained compressed performance file
  - Normally it is about 5 times smaller than the ".xf" file
  - Contains the information needed from the application binary
    - Can be reused, even if the application binary is no longer available or if it was rebuilt
  - It is the only input format accepted by Cray Apprentice^2
- With CrayPat 4.2 and newer, the "ap2" file is generated by default when executing pat_report

Automatic Profiling Analysis (APA)

1. Load CrayPat & Cray Apprentice^2 module files
   - % module load xt-craypat/4.4 apprentice2/4.4
2. Build application
   - % make clean
   - % make
3. Instrument application for automatic profiling analysis
   - % pat_build **–O apa** a.out
     - You should get an instrumented program a.out+pat
4. Run application to get top time consuming routines
   - Remember to modify <script> to run *a.out+pat*
   - Remember to run on Lustre
   - % aprun … *a.out+pat* *(or* qsub <pat script>)
     - You should get a performance file ("<sdatafile>.xf") or multiple files in a directory <sdatadir>
5. Generate .apa file (4.2 only)
   - % pat_report –o *my_sampling_report* [<sdatafile>.xf | <sdatadir>]
     - creates a report file & an automatic profile analysis file <apafile>.apa

APA File Example

```
#  You can edit this file, if desired, and use it
#  to reinstrument the program for tracing like this:
#
#          pat_build -O mhd3d.Oapa.x+4125-401sdt.apa
#
#  These suggested trace options are based on data from:
#
#       /home/crayadm/ldr/mhd3d/run/mhd3d.Oapa.x+4125-401sdt.ap2,
#         /home/crayadm/ldr/mhd3d/run/mhd3d.Oapa.x+4125-401sdt.xf

# ----------------------------------------------------------------------

#       HWPC group to collect by default.

  -Drtenv=PAT_RT_HWPC=1  # Summary with instructions metrics.

# ----------------------------------------------------------------------

#       Libraries to trace.

  -g mpi

# ----------------------------------------------------------------------

#       User-defined functions to trace, sorted by % of samples.
#       Limited to top 200. A function is commented out if it has < 1%
#       of samples, or if a cumulative threshold of 90% has been reached,
#       or if it has size < 200 bytes.

  -w  # Enable tracing of user-defined functions.
      # Note: -u should NOT be specified as an additional option.
```

```
# 43.37%  99659 bytes
        -T mlwxyz_

# 16.09%  17615 bytes
        -T half_

#  7.98%  12666 bytes
        -T full_

#  6.82%  6846 bytes
        -T artv_

. . .

#  1.29%  5352 bytes
        -T currenh_

#  1.03%  25294 bytes
        -T bndbo_

#  Functions below this point account for less than 10% of samples.


#  1.03%  31240 bytes
#         -T bndto_

#  0.51%  11169 bytes
#         -T bncto_

. . .

# ----------------------------------------------------------------------

  -o mhd3d.x+apa               # New instrumented program.

 /work/crayadm/ldr/mhd3d/mhd3d.x  # Original program.
```

Steps to collecting performance data

6. Look at <apafile>.apa file and at the "*my_sampling_report*
   - Verify if additional instrumentation is wanted
7. Instrument application for further analysis (a.out+apa)
   - % pat_build **–O <apafile>.apa**
     - You should get an instrumented program a.out+apa
8. Run application
   - Remember to modify <script> to run *a.out+apa*
   - % aprun … *a.out+apa* *(or* qsub <apa script>)
     - You should get a performance file ("<datafile>.xf") or multiple files in a directory <datadir>
9. Create text report
   - % pat_report –o *my_text_report.txt* [<datafile>.xf | <datadir>]
     - Will generate a compressed performance file (<datafile>.ap2)
10. View results in text (my_text_report.txt) and/or with Cray Apprentice^2
    - % app2 <*datafile*>.ap2

CrayPat API - for fine grain instrumentation

- Fortran

```
include "pat_apif.h"
…
call PAT_region_begin(id, "label", ierr)
do i = 1,n
…
enddo
call PAT_region_end(id, ierr)
```

- C

```
include <pat_api.h>
…
ierr = PAT_region_begin(id, "label");
< code segment >
ierr = PAT_region_end(id);
```

Pat_report Output

```
CrayPat/X:  Version 4.3.0 Revision 1803 (xf 1649)  06/10/08 12:01:46
Number of PEs (MPI ranks):      32
Number of Threads per PE:       1
Number of Cores per Processor:  4
Execution start time:   Mon Jun 23 14:07:31 2008
System type and speed:   x86_64  2100 MHz
Current path to data file:
  /home/crayadm/ldr/mhd3d/run/mhd3d.Oapa.x+4681-295sdt.ap2  (RTS)
  /home/crayadm/ldr/mhd3d/run/mhd3d.Oapa.x+4681-295sdt.xf   (RTS)
Notes:
    Sampling interval was 10000 microseconds (100.0/sec)
  BSD timer type was ITIMER_PROF


  Trace option suggestions have been generated into a separate file,
  from the data in the next table.  You can examine the file, edit it
  if desired, and use it to reinstrument the program like this:

            pat_build -O mhd3d.Oapa.x+4681-295sdt.apa
  (To see the list, specify:  -s traced_functions=show)
...
```


Sampling Output (Notes for Table 1)

```
Notes for table 1:

  Table option:
    -O samp_profile
  Options implied by table option:
    -d sa%@0.95,sa,imb_sa,imb_sa% -b gr,fu,pe=HIDE

  Options for related tables not shown by default:
    -O samp_profile+src

  The Total value for Samp is the sum of the Group values.
  The Group value for Samp is the sum of the Function values.
  The Function value for Samp is the avg of the PE values.
    (To specify different aggregations, see:  pat_help report options s1)

  This table shows only lines with Samp% > 0.95.
    (To set thresholds to zero, specify:  -T)

  Percentages at each level are of the Total for the program.
    (For percentages relative to next level up, specify:
      -s percent=r[elative])
```

Sampling Output (Table 1)

Samp % provides absolute percentages

```
Table 1:  Profile by Function

 Samp % |  Samp  |  Imb.  |  Imb.  |Group
        |        |  Samp  |  Samp % | Functio
                                     |  PE=HIDE'

 100.0% |   775  |    -- |        |Total
|-------------------------------------------
|  94.2% |   730  |       |     -- |USER
||------------------------------------------
||  43.4% |   336  |  8.75 |   2.6% |mlwxyz_
||  16.1% |   125  |  6.28 |   4.9% |half_
||   8.0% |    62  |  6.25 |   9.5% |full_
||   6.8% |    53  |  1.88 |   3.5% |artv_
||   4.9% |    38  |  1.34 |   3.6% |bnd_
||   3.6% |    28  |  2.00 |   6.9% |currenf_
||   2.2% |    17  |  1.50 |   8.6% |bndsf_
||   1.7% |    13  |  1.97 |  13.5% |model_
||   1.4% |    11  |  1.53 |  12.2% |cfl_
||   1.3% |    10  |  0.75 |   7.0% |currenh_
||   1.0% |     8  |  5.28 |  41.9% |bndbo_
||   1.0% |     8  |  8.28 |  53.4% |bndto_
|=============================================
|   5.4% |    42  |    -- |     -- |MPI
||------------------------------------------
||   1.9% |    15  |  4.62 |  23.9% |mpi_sendrecv_
||   1.8% |    14  | 16.53 |  55.0% |mpi_bcast_
||   1.7% |    13  |  5.66 |  30.7% |mpi_barrier_
|=============================================
```


Sampling Output (Notes for Table 2)

```
Notes for table 2:

  Table option:
    -O samp_profile+src
  Options implied by table option:
    -d sa%@0.95,sa,imb_sa,imb_sa% -b gr,fu,so,li,pe=HIDE

  Options for related tables not shown by default:
    -O samp_profile

  The Total value for Samp is the sum of the Group values.
  The Group value for Samp is the sum of the Function values.
  The Function value for Samp is the sum of the Source values.
  The Source value for Samp is the sum of the Line values.
  The Line value for Samp is the avg of the PE values.
    (To specify different aggregations, see:  pat_help report options s1)

  This table shows only lines with Samp% > 0.95.
    (To set thresholds to zero, specify:  -T)

  Percentages at each level are of the Total for the program.
    (For percentages relative to next level up, specify:
      -s percent=r[elative])
```

Sampling Output (Table 2)

```
Table 2:  Profile by Group, Function, and Line

 Samp % | Samp |  Imb.  | Imb.  |Group
        |      |  Samp  | Samp % | Function
        |      |        |        |  Source
        |      |        |        |   Line
        |      |        |        |    PE='HIDE'

 100.0% |  777 |    -- |    -- |Total
|-----------------------------------------------
|  94.2% |  732 |    -- |    -- |USER
||----------------------------------------------
||  43.4% |  337 |    -- |    -- |mlwxyz
3||                                |  ldr/mhd3d/src/mlwxyz.f
||||-------------------------------------------
4|||   2.1% |   16 |  1.47 |   8.9% |line.39
4|||   2.8% |   22 |  2.25 |   9.7% |line.78
4|||   1.2% |    9 |  1.09 |  11.3% |line.116
4|||   1.4% |   11 |  1.22 |  10.5% |line.129
4|||   2.2% |   17 |  2.12 |  11.5% |line.139
4|||  .
4|||   2.7% |   21 |  0.84 |   4.0% |line.568
4|||   1.3% |   10 |  1.72 |  14.8% |line.604
4|||   2.4% |   19 |  0.72 |   3.7% |line.634
||||==========================================
||||==========================================
||  16.1% |  125 |    -- |    -- |half
3||                                |  ldr/mhd3d/src/half.f
||||-------------------------------------------
4|||   5.4% |   42 |  6.41 |  13.8% |line.28
4|||  10.7% |   83 |  5.91 |   6.9% |line.40
||||==========================================
||   8.0% |   62 |    -- |    -- |full
3||                                |  ldr/mhd3d/src/full.f
||||-------------------------------------------
4|||   8.0% |   62 |  6.31 |   9.6% |line.22
||||==========================================
||...
||==============================================
|   5.4% |   42 |    -- |    -- |MPI
||---------------------------------------------
||  1.9% |   15 |  4.62 |  23.9% |mpi_sendrecv_
||||-------------------------------------------
||||==========================================
||  1.8% |   14 | 16.53 |  55.0% |mpi_bcast_
||  1.7% |   13 |  5.66 |  30.7% |mpi_barrier_
|==============================================
```


Sampling Output (Table 3)

```
Notes for table 3:

  Table option:
    -O program_time
  Options implied by table option:
    -d pt -b pe=[mmm]

  The Total value for Process Time is the avg of the PE values.
    (To specify different aggregations, see:  pat_help report options s1)

Table 3:  Program Wall Clock Time

   Process |PE[mmm]
      Time |

 13.926882 |Total
|-------------------
|  14.093036 |pe.0
|  13.924961 |pe.13
|  13.744948 |pe.14
|====================
```

Table 1: Flat Profile (Default)

```
Notes for table 1:

  Table option:
    -O profile
  Options implied by table option:
    -d ti%@0.95,ti,imb_ti,imb_ti%,tr -b gr,fu,pe=HIDE
  Other options:
    -H

  Options for related tables not shown by default:
    -O profile_pe.th           -O callers
    -O profile_th_pe           -O callers+src
    -O profile+src             -O calltree
    -O load_balance            -O calltree+src

  The Total value for each of Time, Calls is the sum of the Group values.
  The Group value for each of Time, Calls is the sum of the Function values.
  The Function value for each of Time, Calls is the avg of the PE values.
    (To specify different aggregations, see:  pat_help report options s1)

  This table shows only lines with Time% > 0.95.
    (To set thresholds to zero, specify:  -T)

  Percentages at each level are of the Total for the program.
    (For percentages relative to next level up, specify:
      -s percent=r[elative])
```


Table 1: Flat Profile (Continuation)

```
Table 1:  Profile by Function Group and Function

 Time % |       Time |Imb. Time |   Imb. | Calls |Group
        |            |          | Time % |       | Function
        |            |          |        |       |  PE='HIDE'

 100.0% | 104.593634 |       -- |     -- | 22649 |Total
|-----------------------------------------------------------
|  71.0% |  74.230520 |       -- |     -- | 10473 |MPI
||----------------------------------------------------------
||  69.7% |  72.905208 | 0.508369 |   0.7% |   125 |mpi_allreduce_
||   1.0% |   1.050931 | 0.030042 |   2.8% |    94 |mpi_alltoall_
||==========================================================
|  25.3% |  26.514029 |       -- |     -- |    73 |USER
||----------------------------------------------------------
||  16.7% |  17.461110 | 0.329532 |   1.9% |    23 |selfgravity_
||   7.7% |   8.078474 | 0.114913 |   1.4% |    48 |fft4_
||==========================================================
|   2.5% |   2.659429 |       -- |     -- |   435 |MPI_SYNC
||----------------------------------------------------------
||   2.1% |   2.207467 | 0.768347 |  26.2% |   172 |mpi_barrier_(sync)
||==========================================================
|   1.1% |   1.188998 |       -- |     -- | 11608 |HEAP
||----------------------------------------------------------
||   1.1% |   1.166707 | 0.142473 |  11.1% |  5235 |free
|===========================================================
```

Table 2: Load Balance

```
Table 2:  Load Balance with MPI Sent Message Stats
 Time %  |        Time |  Sent Msg |  Sent Msg |Avg Sent |Group
         |             |      Count |     Total |Msg Size | PE[mmm]
         |             |            |     Bytes |         |
 100.0% | 104.628869 |  2041 |  9272032 | 4542.89 |Total
|-------------------------------------------------------------
|  71.0% |  74.246813 |  2041 |  9272032 | 4542.89 |MPI
|-------------------------------------------------------------
|    1.1% |  74.820272 |  2332 | 10745856 | 4608.00 |pe.41
|    1.1% |  74.149777 |  2332 | 10745856 | 4608.00 |pe.25
|    1.1% |  73.934590 |  2332 | 10745856 | 4608.00 |pe.19
|=============================================================
|  25.3% |  26.514143 |    -- |       -- |      -- |USER
|-------------------------------------------------------------
|    0.4% |  26.960079 |    -- |       -- |      -- |pe.22
|    0.4% |  26.367891 |    -- |       -- |      -- |pe.37
|    0.4% |  26.212226 |    -- |       -- |      -- |pe.48
|=============================================================
|   2.5% |   2.660105 |    -- |       -- |      -- |MPI_SYNC
|-------------------------------------------------------------
|    0.1% |   3.492460 |    -- |       -- |      -- |pe.34
|    0.0% |   2.625068 |    -- |       -- |      -- |pe.36
|    0.0% |   1.683962 |    -- |       -- |      -- |pe.15
|=============================================================
|   1.2% |   1.207056 |    -- |       -- |      -- |HEAP
|-------------------------------------------------------------
|    0.0% |   1.346021 |    -- |       -- |      -- |pe.6
|    0.0% |   1.215907 |    -- |       -- |      -- |pe.55
|    0.0% |   0.960383 |    -- |       -- |      -- |pe.49
|=============================================================
```


Table 3: MPI Send Stats by Caller

```
Notes for table 3:

  Table option:
    -O mpi_callers
  Options implied by table option:
    -d sm,sc@,mb1..7 -b fu,ca,pe=[mmm]

  Options for related tables not shown by default:
    -O mpi_dest_bytes          -O mpi_dest_counts

  The Total value for each data item is the sum of the Function values.
  The Function value for each data item is the sum of the Caller values.
  The Caller value for each data item is the avg of the PE values.
    (To specify different aggregations, see:  pat_help report options s1)

  This table shows only lines with Sent Msg Count > 0.


Table 3:  MPI Sent Message Stats by Caller

 Sent Msg |  Sent |  4KB<= |Function
    Total |   Msg |  MsgSz |  Caller
    Bytes | Count |  <64KB |   PE[mmm]
          |       |  Count |

  9272032 |  2041 |  2041 |Total
|---------------------------------
|  9272032 |  2041 |  2041 |mpi_isend
|          |       |       |  mpi_shift_
||||--------------------------------
3||  10745856 |  2332 |  2332 |pe.33
3||  10745856 |  2332 |  2332 |pe.38
3||   3829760 |   880 |   880 |pe.63
|=================================
```

Table 5: Heap Statistics

```
Notes for table 5:

  Table option:
    -O heap_hiwater
  Options implied by table option:
    -d am@,ub,ta,ua,tf,nf,ac,ab -b pe=[mmm]
  Other options:
    -H

  The Total value for each data item is the avg of the PE values.
    (To specify different aggregations, see:  pat_help report options s1)

  This table shows only lines with Tracked Heap HiWater MBytes > 0.


Table 5:  Heap Stats during Main Program

 Tracked |  Total | Total | Tracked | Tracked |PE[mmm]
    Heap | Allocs | Frees | Objects |  MBytes |
 HiWater |        |       |     Not |     Not |
  MBytes |        |       |  Freed  |  Freed  |

 139.669 |   6372 |  5235 |    1137 |   9.671 |Total
|-----------------------------------------------------
| 140.946 |   6389 |  4989 |    1400 |  10.999 |pe.38
| 139.645 |   5983 |  4892 |    1091 |   9.641 |pe.14
| 138.758 |   5695 |  4656 |    1039 |   8.754 |pe.63
|=====================================================
```


Table 6: Heap Leaks

```
Notes for table 6:

  Table option:
    -O heap_leaks
  Options implied by table option:
    -d lb%@1,lb@0.0005,lc -b ca,pe=[mmm]

  The Total value for each of Tracked Objects Not Freed, Tracked MBytes Not
    Freed is the sum of the Caller values.
  The Caller value for each of Tracked Objects Not Freed, Tracked MBytes Not
    Freed is the avg of the PE values.
    (To specify different aggregations, see:  pat_help report options s1)

  This table shows only lines with:
    Tracked MBytes Not Freed% > 1
    Tracked MBytes Not Freed > 0.0005    (To set thresholds to zero, specify:  -T)

Table 6:  Heap Leaks during Main Program
 Tracked | Tracked | Tracked |Caller
  MBytes |  MBytes | Objects | PE[mmm]
     Not |     Not |     Not |
 Freed % |   Freed |   Freed |

   100.0% |   0.010 |       1 |Total
|-----------------------------------
|   95.7% |   0.010 |       1 |main
||----------------------------------
||    4.2% |   0.010 |       1 |pe.11
||    4.2% |   0.010 |       1 |pe.6
||    4.2% |   0.010 |       1 |pe.5
|===================================
```

Table 7: I/O (Read) Statistics

```
Notes for table 7:

  Table option:
    -O read_stats
  Options implied by table option:
    -d rt,rb,rR,rd@,rC -b fi,pe=[mmm],fd

  The Total value for each data item is the sum of the File Name values.
  The File Name value for each of Read B/Call, Read Time, Reads is the avg of
    the PE values.
  The File Name value for each of Read Rate MB/sec, Read MB is the sum of the
    PE values.
  The PE value for each data item is the sum of the File Desc values.
    (To specify different aggregations, see:  pat_help report options s1)

  This table shows only lines with Reads > 0.


Table 7:  File Input Stats by Filename

  Read |  Read MB |Read Rate | Reads |    Read |File Name
  Time |          |   MB/sec |       | B/Call |  PE[mmm]
       |          |          |       |        |   File Desc

 0.000 | 0.000065 | 0.925430 |     3 |  22.67 |Total
|-----------------------------------------------------------
| 0.000 | 0.000065 | 0.925430 |     3 |  22.67 |input
||----------------------------------------------------------
|| 0.002 | 0.000065 | 0.038560 |    68 |   1.00 |pe.0
3|                                                 fd.12
|| 0.000 |       -- |       -- |    -- |     -- |pe.6
|| 0.000 |       -- |       -- |    -- |     -- |pe.5
|===========================================================
```


Table 8: I/O (Write) Statistics

```
Table 8:  File Output Stats by Filename

 Write |  Write MB |  Write Rate |  Writes |   Write |File Name
  Time |           |      MB/sec |         | B/Call |  PE[mmm]
       |           |             |         |        |   File Desc

 0.000 | 0.002298 |  14.088269 |      12 | 200.83 |Total
|-----------------------------------------------------------
| 0.000 | 0.000298 |   2.070438 |       1 | 312.00 |stderr
||----------------------------------------------------------
|| 0.000 | 0.000012 |   0.026614 |       1 |  13.00 |pe.17
3|                                                  fd.2
|| 0.000 | 0.000012 |   0.253220 |       1 |  13.00 |pe.7
3|                                                  fd.2
|| 0.000 | 0.000012 |   2.840267 |       1 |  13.00 |pe.0
3|                                                  fd.2
||==========================================================
| 0.000 | 0.002001 | 102.986588 |      11 | 190.73 |stdout
||----------------------------------------------------------
|| 0.000 | 0.002001 |   4.291078 |     269 |   7.80 |pe.0
3|                                                  fd.1
|| 0.000 |       -- |         -- |      -- |     -- |pe.6
|| 0.000 |       -- |         -- |      -- |     -- |pe.5
|===========================================================
```

Callers Profile (Bottom Up)

```
Notes for table 1:

 Table option:
   -O callers
 Options implied by table option:
   -d ti%@0.95,ti,tr -b gr,fu,ca,pe=HIDE
 Other options:
   -H

Table 1:  Profile by Function and Callers

 Time % |        Time | Calls |Group
        |             |       | Function
        |             |       |  Caller
        |             |       |   PE='HIDE'

 100.0% | 104.593634 | 22651 |Total
|-------------------------------------------
|  71.0% |  74.230520 | 10473 |MPI
||------------------------------------------
||  69.7% |  72.905208 |   125 |mpi_allreduce_
|||-----------------------------------------
3||  47.0% |  49.206835 |    48 |ffte4_
||||----------------------------------------
4|||  46.1% |  48.193285 |    47 |ffte4_(exclusive)
4|||   1.0% |   1.013550 |     1 |selfgravity_
||||========================================
3||  22.7% |  23.694255 |    23 |selfgravity_
||||========================================
3||   1.0% |   1.050931 |    94 |mpi_alltoall_
||                       |       | pztrans_
||=========================================
```


Callers Profile – MPI (Cont.)

```
Table 1:  Profile by Function and Callers

 Time % |        Time | Calls |Group
        |             |       | Function
        |             |       |  Caller
        |             |       |   PE='HIDE'

 100.0% | 103.200155 | 10981 |Total
|-------------------------------------------
|  72.2% |  74.494482 | 10473 |MPI
||------------------------------------------
||  71.1% |  73.332263 |   125 |mpi_allreduce_
|||-----------------------------------------
3||  47.9% |  49.425113 |    48 |ffte4_
||||----------------------------------------
4|||  46.9% |  48.386489 |    47 |ffte4_(exclusive)
4|||   1.0% |   1.038624 |     1 |selfgravity_
||||========================================
3||  23.2% |  23.904200 |    23 |selfgravity_
||=========================================
|  25.5% |  26.347519 |    73 |USER
||------------------------------------------
||  16.8% |  17.304612 |    23 |selfgravity_
|||-----------------------------------------
3||   7.9% |   8.201878 |    11 |firststep_
3||   7.9% |   8.201270 |    11 |secondstep
||=========================================
||   7.8% |   8.094927 |    48 |ffte4_
3|                       |       | selfgravity_
|==========================================
|   2.3% |   2.358154 |   435 |MPI_SYNC
||------------------------------------------
||   2.0% |   2.060709 |   172 |mpi_barrier_(sync)
|||-----------------------------------------
3||   2.0% |   2.025665 |    96 |ffte4_
||||----------------------------------------
4|||   1.1% |   1.172514 |    49 |selfgravity_
|==========================================

|=====================================================
```

Call Tree Profile (Top Down)

```
Table 1:  Function Calltree View

 Time % |      Time | Calls |Calltree
        |           |       | PE='HIDE'

 100.0% | 12.306348 |  1396 |Total
|-----------------------------------------
|  100.0% | 12.302515 |   664 |main
|  |--------------------------------------
|  |  99.4% | 12.226615 |   661 |MAIN_
|  |  |-----------------------------------
3  |  |  80.5% |  9.911106 |   390 |mlwxyz_
|  |  |  |--------------------------------
4  |  |  |  43.8% |  5.394583 |    10 |mlwxyz_(exclusive)
4  |  |  |  14.9% |  1.839600 |    80 |half_
4  |  |  |   8.0% |  0.985683 |    80 |full_
4  |  |  |   7.0% |  0.859738 |    80 |artv_
4  |  |  |   4.6% |  0.560140 |    70 |currenf_
|  |  |  |--------------------------------
5  |  |  |   4.2% |  0.514998 |    10 |currenf_(exclusive)
|  |  |  |================================
4  |  |  |   2.2% |  0.271362 |    70 |currenh_
|  |  |  |--------------------------------
5  |  |  |   1.9% |  0.231544 |    10 |currenh_(exclusive)
|  |  |================================
3  |  |  11.4% |  1.406021 |    99 |bnd
|  |  |--------------------------------
4  |  |  |   7.1% |  0.870392 |    11 |bnd_(exclusive)
4  |  |  |   2.1% |  0.259125 |    11 |bndsf_
4  |  |  |   1.6% |  0.201370 |    22 |bndbo_
|  |  |  |--------------------------------
5  |  |  |   1.0% |  0.127880 |    11 |bndbo_(exclusive)
|  |  |================================
3  |  |   3.2% |  0.393540 |     3 |mpi_bcast_(sync)
3  |  |   2.0% |  0.245981 |     1 |model_
|  |  |--------------------------------
4  |  |   2.0% |  0.245981 |     1 |model_(exclusive)
|  |  |================================
3  |  |   1.5% |  0.188091 |    33 |cfl_
|  |  |--------------------------------
4  |  |   1.5% |  0.184726 |    11 |cfl_(exclusive)
```


Callers Profile with Line Numbers

```
Notes for table 1:
  Table option:
    -O calltree+src
  Options implied by table option:
    -d ti%@0.95,ti,tr -b ct,pe=HIDE -s show_ca='fu,so,li' \
    -s source_limit='1'
  Other options:
    -H

Table 1:  Calltree View with Callsite Line Numbers

 Time % |      Time | Calls |Calltree
        |           |       | PE='HIDE'

 100.0% | 12.381493 |  1155 |Total
|-----------------------------------------
|  100.0% | 12.380694 |   663 |main:...:line.0
|  |--------------------------------------
|  |  91.8% | 11.370849 |   560 |MAIN_:mhdmain.f:line.368
|  |  |-----------------------------------
3  |  |  43.5% |  5.387737 |    10 |mlwxyz_:mlwxyz.f:line.2
3  |  |   6.9% |  0.849967 |    80 |mlwxyz_:mlwxyz.f:line.604
|  |  |  |--------------------------------
4  |  |  |                |       |artv_:artv.f:line.2
|  |  |  |================================
3  |  |  6.5% |  0.806207 |    80 |mlwxyz_:mlwxyz.f:line.283
|  |  |  |--------------------------------
4  |  |  |   4.2% |  0.514861 |    10 |currenf_:currenf.f:line.2
4  |  |  |   2.0% |  0.247048 |    10 |half_:half.f:line.2
|  |  |  |================================
3  |  |  6.5% |  0.798654 |    10 |bnd_:bnd.f:line.2
3  |  |  3.9% |  0.487620 |    80 |bnd_:bnd.f:line.15
|  |  |  |--------------------------------
4  |  |  |   1.9% |  0.234759 |    10 |bndsf_:bndsf.f:line.2
|  |  |  |================================
3  |  |  2.2% |  0.271831 |    70 |mlwxyz_:mlwxyz.f:line.39
|  |  |  |--------------------------------
4  |  |  |   1.9% |  0.232804 |    10 |currenh_:currenh.f:line.2
|  |  |  |================================
3  |  |  2.1% |  0.253929 |    10 |mlwxyz_:mlwxyz.f:line.253
4  |  |  |                |       | half_:half.f:line.2
3  |  |  2.0% |  0.252611 |    10 |mlwxyz_:mlwxyz.f:line.129
4  |  |  |                |       | half_:half.f:line.2
3  |  |  2.0% |  0.249683 |    10 |mlwxyz_:mlwxyz.f:line.273
```

Load Balancing Function per PE

```
Notes for table 1:

  High level option:  -O load_balance_program
  Low level options:  -d ti%@0.05,cum ti%,ti,tr -b exp,pe

  This table shows only lines with Time% > 0.05.

  Percentages at each level are relative
    (for absolute percentages, specify:  -s percent=a).


Table 1:  Load Balance across PE's

 Time % |   Cum. |     Time |  Calls |Experiment=1
        | Time % |          |        |PE

 100.0% | 100.0% | 3.798177 | 579653 |Total
|-------------------------------------------------
|   2.1% |   2.1% | 3.823080 |   7160 |pe.0
|   2.1% |   4.2% | 3.799148 |  13753 |pe.8
| ...
|   2.1% |  97.9% | 3.796151 |   7683 |pe.5
|   2.1% | 100.0% | 3.796144 |  10431 |pe.29
|=================================================
```


Table 3: LB Across PE's by Function

```
Notes for table 3:

  High level option:  -O load_balance_function
  Low level options:  -d ti%@0.05,cum_ti%,ti,tr -b exp,gr,fu,pe

  This table shows only lines with Time% > 0.05.

  Percentages at each level are relative
    (for absolute percentages, specify:  -s percent=a).


Table 3:  Load Balance across PE's by Function

 Time % |   Cum. |     Time |  Calls |Experiment=1
        | Time % |          |        |Group
        |        |          |        | Function
        |        |          |        |  PE

 100.0% | 100.0% | 3.798177 | 579653 |Total
|-------------------------------------------------
|  70.9% |  70.9% | 2.692783 | 245380 |USER
||------------------------------------------------
||  97.1% |  97.1% | 2.615916 |    576 |sweep_
|||-----------------------------------------------
|||   2.2% |   2.2% | 2.753279 |     12 |pe.0
|||   2.1% |   4.3% | 2.654725 |     12 |pe.5
||| . . .
|||   2.0% |  98.0% | 2.525587 |     12 |pe.43
|||   2.0% | 100.0% | 2.523325 |     12 |pe.37
|||===============================================
|| . . .
||================================================
||   0.4% |  99.2% | 0.010300 | 118080 |snd_real_
|||-----------------------------------------------
|||   2.4% |   2.4% | 0.011699 |   2880 |pe.26
|||   2.3% |   4.7% | 0.011475 |   2880 |pe.27
||| . . .
|||   1.5% |  98.6% | 0.007266 |   1440 |pe.0
|||   1.4% | 100.0% | 0.006907 |   1440 |pe.5
|||===============================================
```

Table 3 (Cont.)

```
||=============================================
|  28.8% |  99.7% | 1.092307 | 238224 |MPI
||---------------------------------------------
|  76.1% |  76.1% | 0.831311 | 118080 |mpi_recv_
||---------------------------------------------
|   2.7% |   2.7% | 1.066077 |   1440 |pe.47
|   2.6% |   5.3% | 1.034307 |   2160 |pe.41
|   . . .
|   1.8% |  98.6% | 0.700970 |   2160 |pe.1
|   1.4% | 100.0% | 0.573420 |   1440 |pe.0
||=============================================
. . .
||=============================================
|   0.2% |  99.8% | 0.007329 |  95597 |HEAP
||---------------------------------------------
|  61.1% |  61.1% | 0.004481 |  47861 |malloc
||---------------------------------------------
|   2.7% |   2.7% | 0.005884 |   1242 |pe.12
|   2.6% |   5.4% | 0.005658 |   1226 |pe.19
|   . . .
|   1.3% |  99.5% | 0.002827 |    618 |pe.34
|   0.5% | 100.0% | 0.001164 |    417 |pe.0
||=============================================
|  38.9% | 100.0% | 0.002848 |  47735 |free
||---------------------------------------------
|   2.7% |   2.7% | 0.003748 |   1422 |pe.37
|   2.7% |   5.5% | 0.003706 |   1469 |pe.43
|   . . .
|   1.4% |  99.3% | 0.001867 |    616 |pe.34
|   0.7% | 100.0% | 0.000896 |    385 |pe.0
||=============================================
|   0.2% | 100.0% | 0.005758 |    452 |IO
||---------------------------------------------
|  81.3% |  81.3% | 0.004679 |    309 |fwrite
||---------------------------------------------
|  34.3% |  34.3% | 0.077141 |    262 |pe.0
|   2.1% |  36.4% | 0.004615 |      1 |pe.8
||  . . .
```


Documentation

- The pat_help utility is an interactive viewer used to access information about and examples of using CrayPat
  - pat_help [topic [subtopic...]]
- See man pages:
  - intro_craypat (all runtime environment variables are here)
  - pat_build (application instrumentation)
  - pat_help
  - pat_report (report generation)
  - hwpc (all HW groups are here)
  - app2 (performance visualization)

pat_help Example

```
% pat_help

    The top level CrayPat/X help topics are listed below.
    A good place to start is:

        overview

    If a topic has subtopics, they are displayed under the heading
    "Additional topics", as below.  To view a subtopic, you need
    only enter as many initial letters as required to distinguish
    it from other items in the list.  To see a table of contents
    including subtopics of those subtopics, etc., enter:

        toc

    To produce the full text corresponding to the table of contents,
    specify "all", but preferably in a non-interactive invocation:

        pat_help all . > all_pat_help
        pat_help report all . > all_report_help

  Additional topics:

    API                     execute
    balance                 experiment
    build                   first_example
    counters                overview
    demos                   report
    environment             run

pat_help (.=quit ,=back ^=up /=top ~=search)
=>
```


pat_help Example

```
pat_help (.=quit ,=back ^=up /=top ~=search)
=> FAQ

    Answers to a few frequently asked questions are given under the
    topics listed below.

  Additional topics that may follow "FAQ":

    Can not access the file ...
    User sampling with compiler hooks present is not allowed
    callers missing; calltree incomplete
    callstack buffer overflow
    can not be traced because ... not writable
    fewer samples than expected
    fractional numbers are shown for calls
    instrumented programs with both MPI and OpenMP
    link information was not found
    pat_report is 'Killed'
    xf_next: Unexpected eof

pat_help FAQ (.=quit ,=back ^=up /=top ~=search)
=>
```

Hands-on Tasks

1. Generate an ".apa" file and a sampling report from your application
2. Read the ".apa" file and add I/O instrumentation
3. Use the .apa file to generate a profile of the application
4. Look at the sampling report and identify areas where work is concentrated. Using the CrayPat API add instrumentation around the important loop(s)
5. Generate a second profile of the application with code regions
6. Obtain MFLOPS, TLB Utilization, Cache Hit/Miss ratios (L1 and L2), Cache utilization (L1, and L2), FP Mix, and Vectorization information for the main regions and functions of the application
7. Visualize the performance file (.ap2) with Cray Apprentice2 and identify the most imbalanced function or region of the application
8. Generate a trace file of the application (if the application is large, limit the size of the trace file)
9. Visualize the trace file with Cray Apprentice2
10. Optimize your application with the data that you collected
    - If you have problems, ask John Levesque for help in tuning and optimizing your application
        - He will not leave Fairbanks till your code gets at least 20% performance improvement

Performance Measurement and Visualization on the Cray XT

Questions / Comments

Thank You!

CSCS
July 13-15, 2009