CRAY
THE SUPERCOMPUTER COMPANY

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

Next Generation Systems

- 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.)

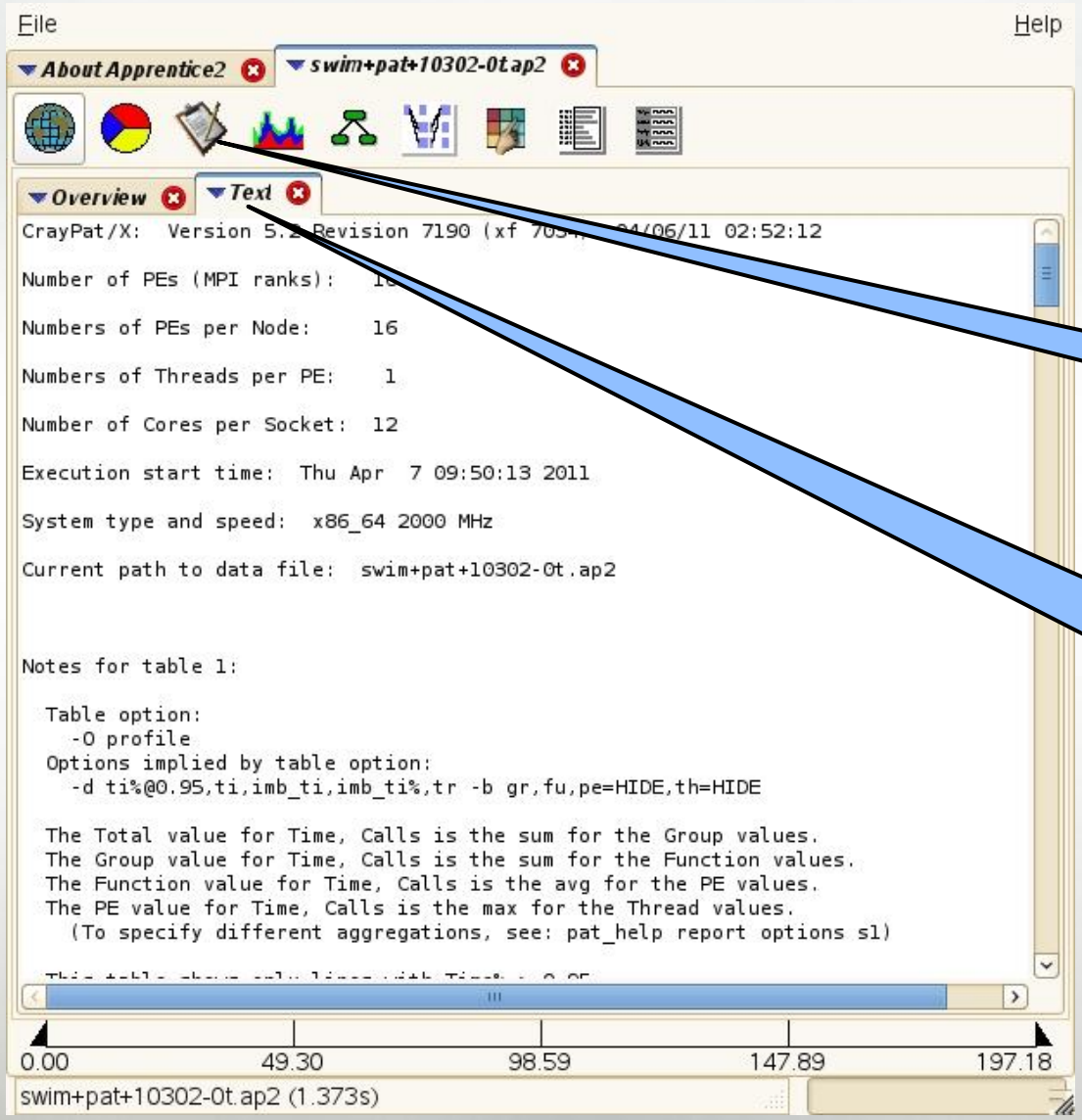
Evolutionary Path of the Cray Performance Tools

- 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



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

Example Default Report for UPC code (Before)

```
Table 1:  Profile by Function
Samp % | Samp |  Imb. |   Imb. |Group
       |      |  Samp | Samp % | Function
       |      |       |        |  PE='HIDE'
100.0% |   77 |    -- |     -- |Total
|-----------------------------------------
|  94.8% |   73 |    -- |     -- |ETC
||----------------------------------------
||  20.8% |   16 | 15.06 |  50.2% |syscall
||  14.3% |   11 | 15.81 |  60.5% |__pgas_barrier_wait_all
||  11.7% |    9 |  7.28 |  47.0% |__pat_tracing_ea_ptr_by_name_set_addr
||   3.9% |    3 |  3.75 |  55.3% |__pat_thread_get
||   3.9% |    3 |  5.00 |  64.5% |__pgas_barrier_notify_pe
||   3.9% |    3 | 19.22 |  90.2% |__pgas_barrier_wait_children
||   3.9% |    3 |  5.88 |  67.4% |__pgas_sync_nbi
||   2.6% |    2 |  4.09 |  70.4% |__pgas_aand
||   2.6% |    2 |  1.84 |  47.6% |__pgas_barrier
…
||=========================================
|   5.2% |    4 |    -- |     -- |USER
||----------------------------------------
||   5.2% |    4 |  4.91 |  56.3% |mpp_alloc
|=========================================
```

Example Default Report for UPC code (After)

```
Table 1:  Profile by Function

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

  100.0% |    7 |   -- |     -- |Total
|-----------------------------------------
|  71.4% |    5 |   -- |     -- |USER
||----------------------------------------
||  57.1% |    4 | 0.25 |   8.3% |mpp_broadcast
||  14.3% |    1 | 0.50 |  66.7% |mpp_alloc
||========================================
|  28.6% |    2 |   -- |     -- |ETC
||----------------------------------------
||  28.6% |    2 | 0.50 |  33.3% |bzero
|========================================
```

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

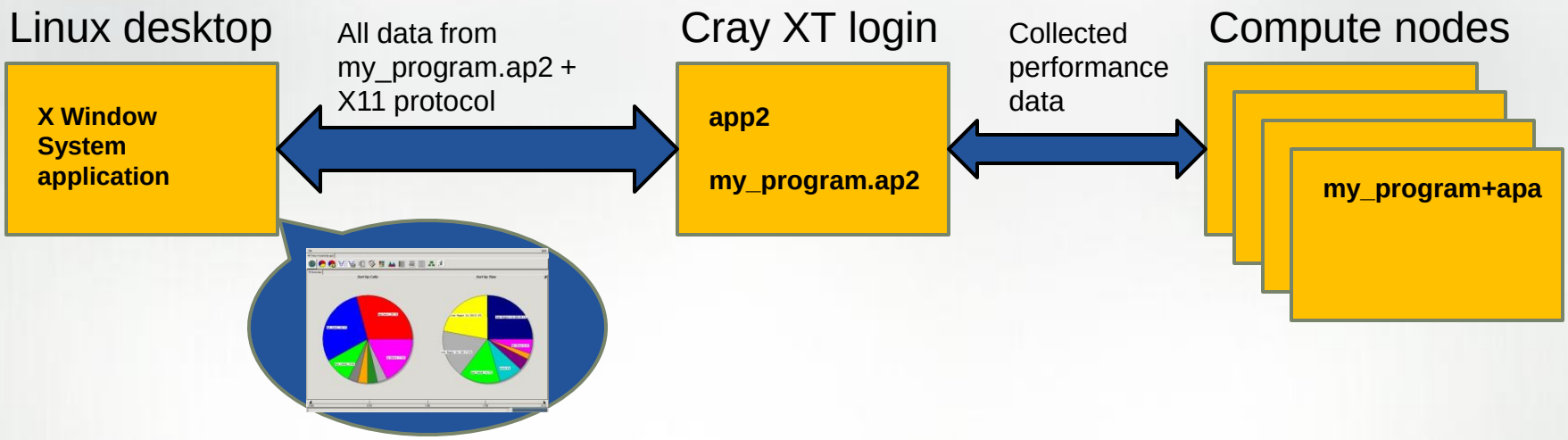
- CPMD
  - MPI, instrumented with pat_build –u, HWPC=1
  - 960 cores

	Perftools 5.1.3	Perftools 5.2.0
.xf -> .ap2	88.5 seconds	22.9 seconds
ap2 -> report	1512.27 seconds	49.6 seconds

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

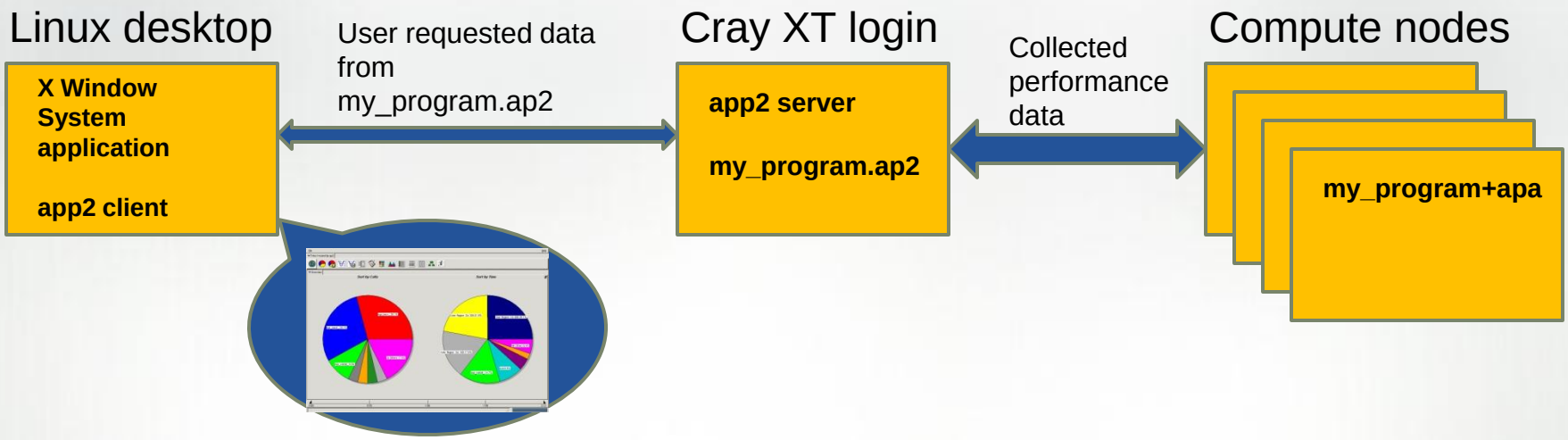
	Perftools 5.1.3	Perftools 5.2.0
.xf -> .ap2	45.2 seconds	15.9 seconds
ap2 -> report	796.9 seconds	28.0 seconds

Old Client/Server (Cray Performance Tools 5.0.0)



- 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)

New Client/Server (Cray Performance Tools 5.2.0)



- 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

Example Accelerator Report

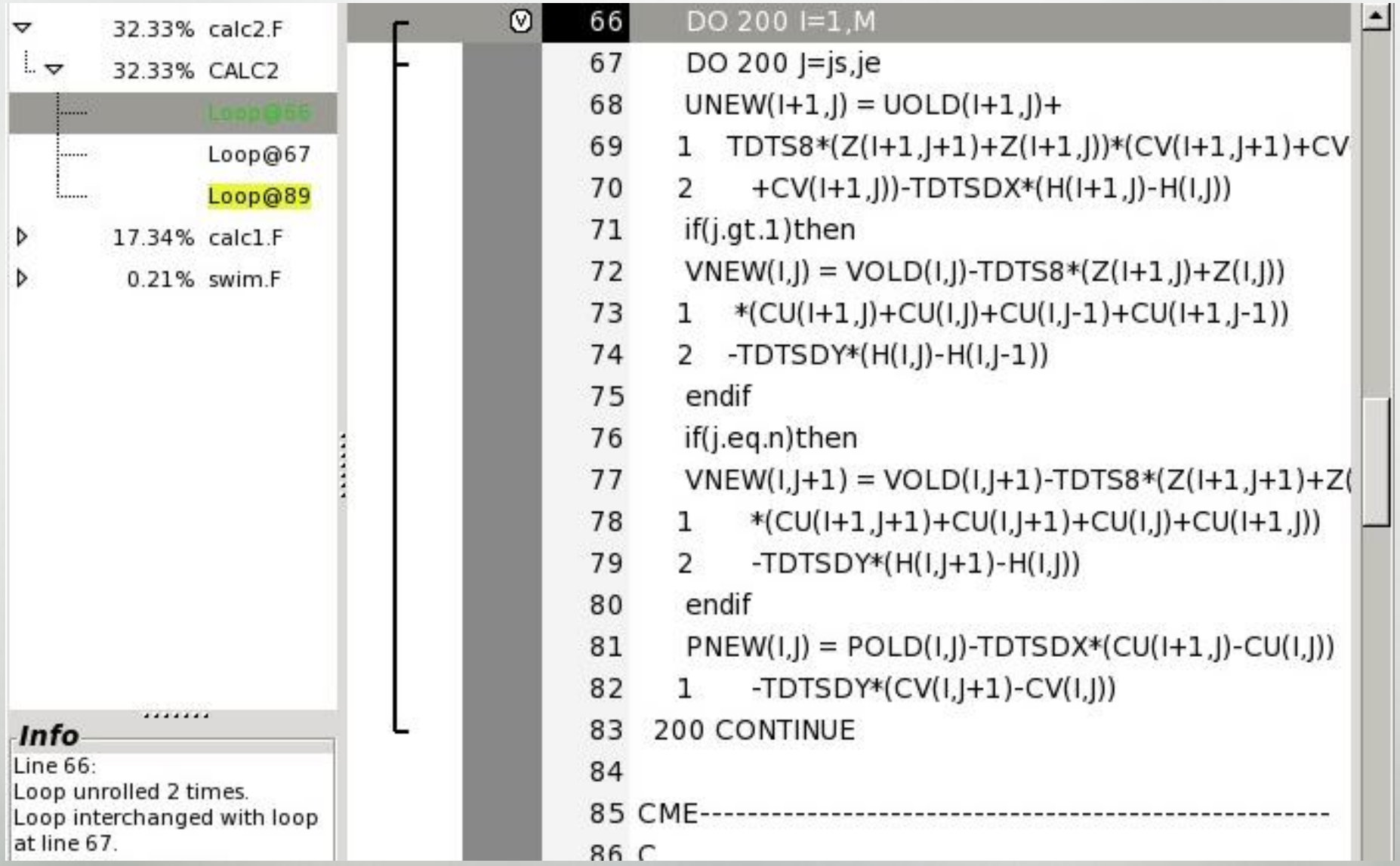
```
Table 3:  Time and Bytes Transferred for Accelerator Regions
   Host | Host Time |  Acc Time |  Acc Copy |  Acc Copy | Calls |Group='ACCELERATOR'
 Time % |           |           |   In (MB) |  Out (MB) |       | PE=0
        |           |           |           |           |       | Thread=0
        |           |           |           |           |       | Calltree
        |           |           |           |           |       |  Function
 100.0% | 14.84495 | 13.615016 | 14550.536 | 10461.216 |  1777 |Total
|-------------------------------------------------------------------------------
| 100.0% | 14.84495 | 13.615016 | 14550.536 | 10461.216 |  1777 |ACCELERATOR
||------------------------------------------------------------------------------
||  93.7% | 13.909414 | 12.418942 | 13274.781 |  9675.075 |  1777 |mg_
|||-----------------------------------------------------------------------------
3||  51.8% |  7.692439 |  7.645484 |  7902.816 |  6399.489 |  1630 |mg3p_
||||----------------------------------------------------------------------------
4|||  21.7% |  3.229140 |  3.216513 |  3758.31 |  2254.986 |   420 |resid_
|||||---------------------------------------------------------------------------
5|||| 11.9% |  1.767674 |  1.763377 |  2254.986 |   751.662 |   140 |resid_(exclusive)
||||||--------------------------------------------------------------------------
6|||||   7.8% |  1.158744 |  1.158958 |  2254.986 |     0.000 |    35 |resid_.ASYNC_COPY@li.459
6|||||   4.1% |  0.604365 |  0.337742 |     0.000 |   751.662 |    35 |resid_.ASYNC_COPY@li.492
6|||||   0.0% |  0.003903 |  0.000000 |     0.000 |     0.000 |    35 |resid_.SYNC_WAIT@li.492
6|||||   0.0% |  0.000662 |  0.266677 |     0.000 |     0.000 |    35 |resid_.ASYNC_KERNEL@li.459
|||||===========================================================================
```

A Peek at Something New

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



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.

CRAY
THE SUPERCOMPUTER COMPANY

Evolution of the Cray Performance Measurement and Analysis Tools

Questions / Comments
Thank You!