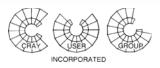
### The New Generation of Cray Performance Tools

Luiz DeRose Sr. Principal Engineer Tools Manager

Albuquerque, NM



CUG 2005

### Outline

- Previous status
- The Cray tools strategy
- Cray Tools
  - HWPC
  - CrayPat
  - Cray Apprentice<sup>2</sup>
- Future directions
- Conclusions



### **CRAY**

### The Starting Point

- Missing a complete solution
  - Cray X1
    - Very good measurement infrastructure
    - Very poor user interface
  - XT3 & XD1
    - No in-house solution
      - Dependence on external providers
- Needed integration across platforms





### The Cray Tools Strategy

- Must be easy to use
- Integrated performance tools solution
- Strategy based on the three main steps normally used for application optimization and tuning:
  - 1. Debug application
    - Nobody really cares how fast the program can compute the wrong answer
      - TotalView is our debugger of choice on all platforms
  - 2. Single processor and vector optimization
    - Make the common case fast
  - 3. Parallel processing and I/O optimization
    - Communication / barriers / etc.



## Single Processor Optimization

- Answer the following questions:
  - 1. Does my program have performance problems?
    - Time / Resource / Hardware Counters measurement
    - Provides overall view of the program execution
      - % pat\_hwpc a.out



## Single Processor Optimization (2)

- Answer the following questions:
  - 2. Where are the main bottlenecks?
    - Sample based profiler (Flat, Call graph, Call path)
    - Event Profiler with runtime summarization



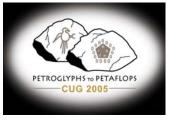
## Single Processor Optimization (3)

- Answer the following questions:
  - 3. What are the causes of performance problems?
    - Compiler feedback
      - Canal / Loopmarks
    - HW counters based Instrumentation library
      - Run-time summarization
      - Hand or automatic instrumentation



### **Parallel Processing Optimization**

- Answer the following questions:
  - 1. Does my program have communication or synchronization problems?
    - Communication Profiler



### Parallel Processing Optimization (2)

- Answer the following questions:
  - 2. What are the causes of parallel performance problems?
    - Task/Thread tracing
      - Tracing library
        - » CrayPat
    - Visualization GUI
      - Cray Apprentice<sup>2</sup>



### Approach

- Take advantage of existing tools infrastructure
  - Improve usability
  - Develop components for missing functionality
  - Port to all Cray platforms
- Close interaction and collaboration with tools builders in Universities, Labs, and Research Centers
- Close interaction with application developers for quick feedback, targeting functionality enhancements



### Collaborations

- Interactions with lead researchers in the performance analysis and optimization field
  - Jack Dongarra's group at UTK
    - PAPI port to all Cray systems
  - Bernd Mohr at Forschungszentrum Jülich
    - KOJAK expert system ported to X1
    - Investigation of CAF and UPC measurement and analysis
    - Automatic performance analysis
  - Jeff Vetter's group at ORNL
    - MPI Profiler (mpiP) port to all Cray platforms
    - Data management techniques in performance analysis
      - Development of filters
      - Hierarchical analysis





## Cray Performance Tools

- PAT\_HWPC
  - Hardware Counters based measurement for whole application
- CrayPat
  - Performance Analysis Toolkit
- Cray Apprentice<sup>2</sup>
  - Performance visualization





# PAT\_HWPC (X1 Series only)

- Collects hardware performance counters information for an application
- No instrumentation required
  - pat\_hwpc [options] a.out
  - Accepts various hardware counters groups
- Results in report with raw counts and derived metrics for the whole execution
  - Hardware counters summed across threads per process





### LIBHWPC (XT3 and XD1)

- Instrumentation library for Fortran, C, and C++
- For each instrumented region provides:
  - Total count & duration (wall clock time & user time)
  - Hardware performance counters information
  - Derived metrics
- Supports:
  - Multiple instrumentation points
  - Nested instrumentation
  - Multiple calls to an instrumented point
- Uses PAPI Version 3
  - PAPI available on base distribution





### CrayPat

### Performance Analysis Toolkit

- pat\_build
  - Utility that automatically instruments the application
    - No source code modification required
- pat runtime library
  - Transparent to the user
  - Collects performance data during execution
  - Writes performance file
- pat\_report
  - Performance analysis utility
- pat\_run
- pat\_help



### pat\_build

- pat\_build <options> a.out instrumented.a.out
- Automatic instrumentation
  - User functions
  - MPI, SHMEM, OpenMP, UPC, CAF, pThreads
  - System calls
  - Dynamic heap
  - Raw I/O, buffered I/O, flexible file I/O
- API available for fine grain instrumentation
- "instrumented.a.out" is executed the same way as the original application
- No recompilation needed!



### CrayPat Functionality: Instrumentation

- Capture of:
  - Software state
    - Thread
    - Call Stack
    - parameter values passed into a function
  - Hardware state
    - Program counter (PC)
    - Hardware performance counters (HWPC)
  - Time stamps
    - Real time clock
    - HWPC Cycle counter





### **Performance Collection**

- Trigger mechanism:
  - Activated by external agent (sampling)
    - Timer
    - Hardware counters overflow
    - (not supported on the XT3 yet)
  - Activated internally (event tracing or synchronous)
    - Code inserted through instrumentation
- Data recording:
  - Summarized during runtime
    - Stored in the form of a profile
  - Trace records for each event
    - Stored in the form of a trace file





Ti	.me%   Cu	m.Time%	Time	Tr	aces  F	Junction
	7.4%	93.8%	3.737219		24000	MPI_Waitall
	1.4%	96.7%	0.707859		56383	MPI_Isend
	0.9%	98.9%	0.455456		56383	MPI_Irecv
	0.1%	99.8%	0.051986		56383	mpi_
	0.1%	99.9%	0.040626		4	InitalCalc1Calc2#88
	0.1%	100.0%	0.026122		24000	mpi_waitall_
	0.0%	100.0%	0.009164		4800	Calc1Calc2#10
	0.0%	100.0%	0.007347		4800	Calc2#20
	0.0%	100.0%	0.006329		4792	Calc3#30
	0.0%	100.0%	0.001980		60	MPI_Reduce
	0.0%	100.0%	0.000083		60	mpi_reduce_
	0.0%	100.0%	0.000030		60	MPI_Type_get_true_extent
	0.0%	100.0%	0.000016		4	main
	0.0%	100.0%	0.000011		4	MPI_Finalize
	0.0%	100.0%	0.000005		4	mpi_comm_rank_
	0.0%	100.0%	0.000005		4	mpi_comm_size_
	0.0%	100.0%	0.00003		4	MPI_Comm_size
	0.0%	100.0%	0.00003		4	_Exit
	0.0%	100.0%	0.00003		4	MPI_Comm_rank
	0.0%	100.0%	0.00003		4	MPI_Init
==				==	======	



5/19/2005

RAY

Luiz DeRose

ldr@cray.com



### pat\_report Call Tree Profile

100.0%       100.0%       50.313549       316932       Total
<pre>  </pre>
39.0%         39.0%         19.645725         115200  calc2_                    36.4%         36.4%         18.336529         4800  pat_region_begin_                    36.4%         36.4%         18.336529         4800  pat_region_begin_                    1.1%         37.6%         0.559553         19200  mpi_waitall_                    1.1%         37.5%         0.550431         9600  MPI_Waitall
36.4%       36.4%       18.336529       4800       pat_region_begin_                    36.4%       18.336529       4800       pat_region_begin_                    1.1%       37.6%       0.559553       19200       mpi_waitall_                    1.1%       37.5%       0.550431       9600       MPI_Waitall
Do 200#21      1.1% 37.6% 0.559553 19200  mpi_waitall_       1.1% 37.5% 0.550431 9600  MPI_Waitall
0.0%   37.6%   0.009122   9600  mpi_waitall_(exclusive)
0.6%   38.2%   0.300126   43200  mpi_isend_
0.6%   38.1%   0.280955   21600  MPI_Isend      0.0%   38.2%   0.019171   21600  mpi_isend_(exclusive)
0.5%   38.6%   0.246678   4800  calc2_(exclusive)      0.4%   39.0%   0.202839   43200  mpi_irecv_
0.4%   39.0%   0.180679   21600  MPI_Irecv       0.0%   39.0%   0.022161   21600  mpi_irecv_(exclusive)
==================================
23.7%   62.8%   11.933617   4800  pat_region_begin_                Do 100#11
4.5%   67.3%   2.282388   19200  mpi_waitall_



5/19/2005

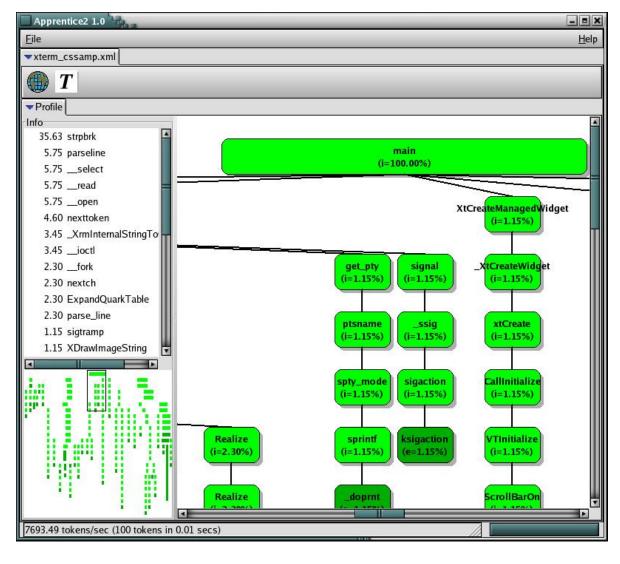
## Cray Apprentice<sup>2</sup>

- Call graph profile
- Communication statistics
- Time-line view
  - Communication
  - I/O
- Activity view
- Pair-wise communication statistics
- Text reports
- Source code mapping

- Target to help identify and correct:
  - Load imbalance
  - Excessive serialization
  - Excessive communication
  - Network contention
  - Poor use of the memory hierarchy
  - Poor functional unit use



### **Call Graph Profile View**

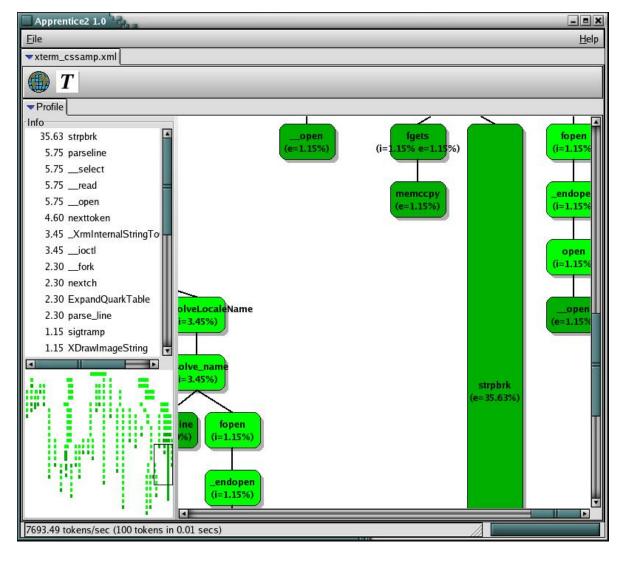




5/19/2005

- RAY

### **Call Graph Profile View**





5/19/2005

RAY

### **Call Graph Profile View**

		Help
alo_prof.xml		±-•r
Profile Text Report		
#0 (total=NA)		
func	hits	
0. MPI_CRAY_reduce[20]	33	
1. barrier[12]	12	
<ol> <li>_mcs_tree_bar_sym[35]</li> </ol>	11	
<ol> <li>MPI_CRAY_send_wait[32]</li> </ol>	10	
4. mpi_sendrecv_[37]	4	
5. MPI_CRAY_request_recv[43]	4	
<pre>6. mpi_waitall_[40]</pre>	3	
7bcopy_prv[72]	3	
<pre>3bcopy_piv(72) 3sma_barrier_local[61]</pre>	2	
9. MPI_Wait[58]	2	
0. write[49]	1	
1. MPI_CRAY_progress[39]	1	
<ol> <li>MPI_CRAI_progress[39]</li> <li>gethostname[24]</li> </ol>	1	
<ol> <li>GetHosthame [24]</li> <li>MPI_CRAY_request_send[71]</li> </ol>	1	
4. halo2a_[38]	1	
5apteamct1[78]	1	
5. start_recv[82]	1	
7. barrier_jw[89]	1	
<pre>8xfer_iolist[98]</pre>	1	
#1 (total=NA)		
func	hits	
	nits 	
0. MPI_CRAY_bcast[23]	24	
L. barrier[12]	19	
2. MPI_CRAY_send_wait[32]	13	
3mcs_tree_bar_sym[35]	9	
4. MPI_CRAY_progress[39]	5	
5. MPI_Wait[58]	5	
5. mpi_waitall_[40]	4	
7. MPI CRAY request recv[43]	3	

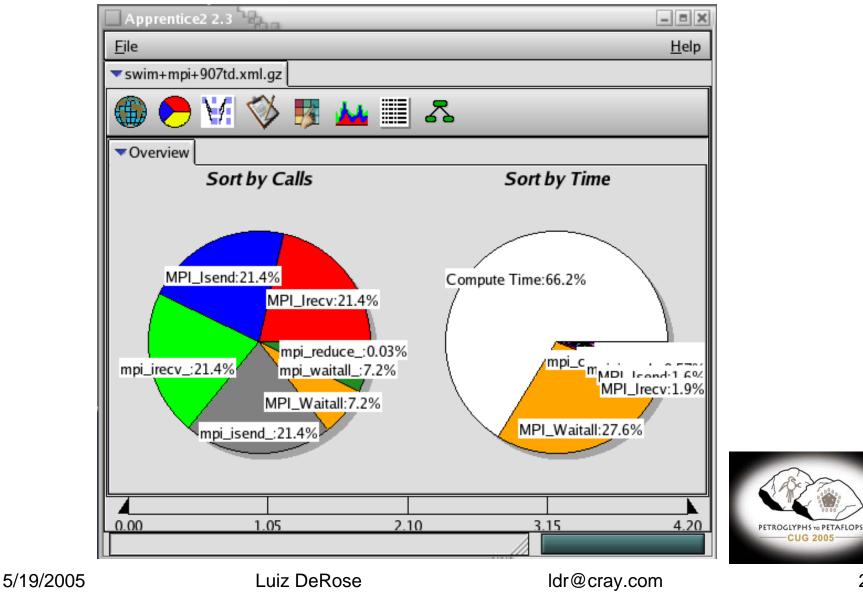
5/19/2005

BAY

PETROGLYPHS TO PETAFLOPS

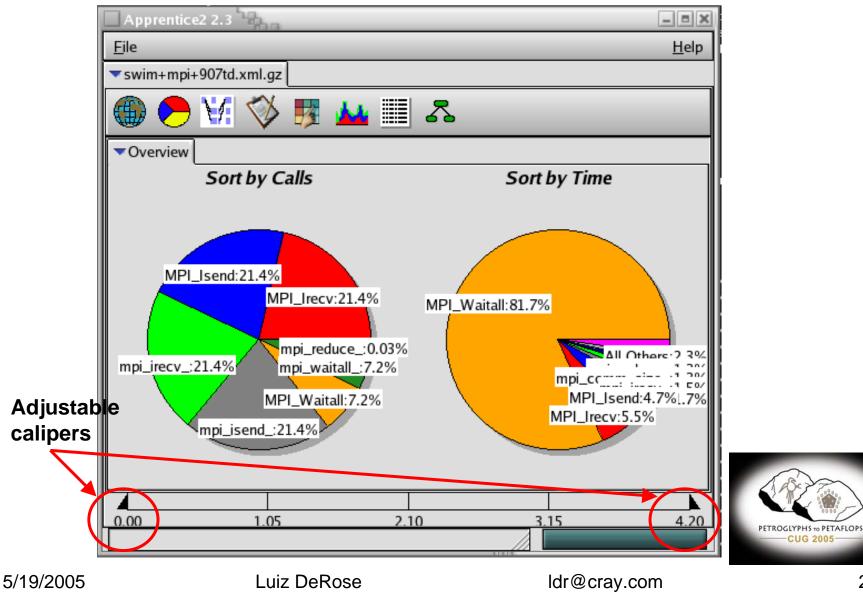
### **Communication Statistics**

RAY



25

### **Communication Statistics**



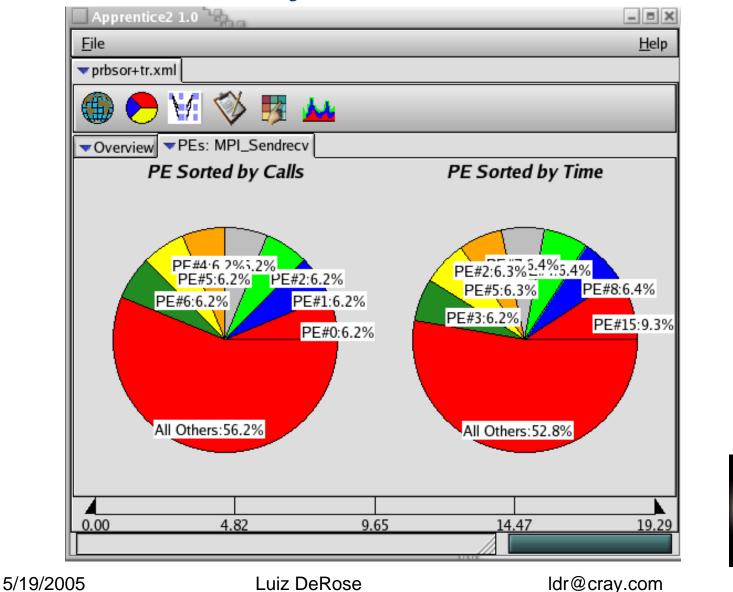
The Supercomput	ter Company
-----------------	-------------

### **Communication Statistics**

Apprentice2 1.0		- = ×	PE Distribution (mpi_ise	nd)
<u>F</u> ile		<u>H</u> elp		πuj
▼halo32p.xml			Apprentice2 1.0	- <b>B</b> X
🌐 🗢 🦌 🗇 👼 🚣			Eile	<u>H</u> elp
▼Overview ▼PEs: mpi_isend_ ▼All C			▼halo32p.xml	
	thers by Calls		🌐 色 🧺 🗇 🐺 📥	
Percents Name			▼Overview ▼PEs: mpi_isend_ ▼All Others: mpi_isend_	
4.76 MPI_Wtime			PE Sorted by Calls: mpi_i	
2.52 mpi_barrier_			Percents Name	
2.38 mpi_allreduce_			3.12 PE #0	
1.90 mpi_send_init_			3.12 PE #1	
1.90 mpi_recv_init_			3.12 PE #2	
1.17 MPI_Type_get_extent			3.12 PE #3	
1.17 MPI_Type_lb			3.12 PE #4	
0.12 MPI_Wait			3.12 PE #5	
0.07 MPI_Comm_rank			3.12 PE #6	
0.05 mpi_comm_rank_			3.12 PE #7	
0.05 mpi_comm_size_		•	3.12 PE #8	
	1.02		3.12 PE #9	
0.00 0.52	1.03 1.55	2.07	3.12 PE #10	
All Others %			0.00 0.52 1.03	1.55 2.07
				PETROGLYPHS to PETAFLOPS CUG 2005
5/19/2005	Luiz DeRose		ldr@cray.com	27

### Statistics by PE

IRAY



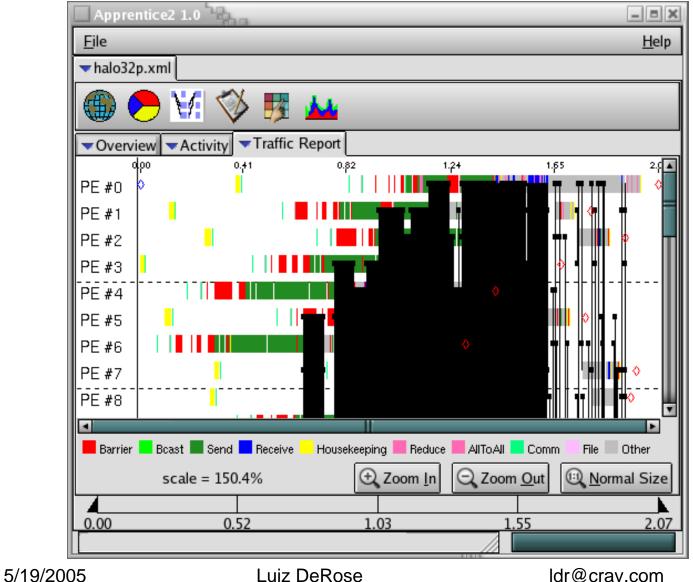


PETROGLYPHS TO PETAFLOPS

#### Statistics by PE Profile Apprentice2 1.0 ¬ - = × File <u>H</u>elp prbsor+tr.xml Apprentice2 1.0 📲 ▼Overview ▼PEs: MPI\_Sendrecv ▼Callers ▼Traffic Report ▼Text Report <u>F</u>ile prbsor+tr.xml PE #0 (total=NA) 4 func time calls . . . . . . . . . - - - -▼Overview ▼PEs: MPI\_Sendrecv ▼Callers MPI\_Sendrecv[28] 2.788269 4000 1. MPI\_Reduce[26] 0.721298 1000 Hits Caller Line Source MPI\_Bcast[12] 0.021593 5 290 /scratch/scr4tb/derose/prbsor/prbsor.c 1000 main 3. MPI\_Init[23] 0.006056 1 ioct1[4] 0.003599 4 1000 main 286 /scratch/scr4tb/derose/prbsor/prbsor.c 49 MPI\_Type\_get\_extent[29] 0.000879 1000 relax 76 /scratch/scr4tb/derose/prbsor/prelax.c MPI\_Type\_lb[30] 0.000383 49 1000 relax 72 /scratch/scr4tb/derose/prbsor/prelax.c 7. MPI\_Get\_processor\_...[21] 0.000134 1 8. MPI\_Wait[32] 0.000067 4 9. MPI\_Finalize[19] 0.000065 1 10. MPI\_Comm\_dup[14] 0.000044 1 11. MPI\_Comm\_rank[16] 0.000040 4 12. MPI\_Comm\_size[17] 0.000034 3 13. MPI Comm free[15] 0.000015 1 4 9.65 0.00 4.82 14.47 19.29 0.00 4.82 9.65 14.47 19.29 Source code mapping PETROGLYPHS TO PETAFLOPS CUG 2005

5/19/2005

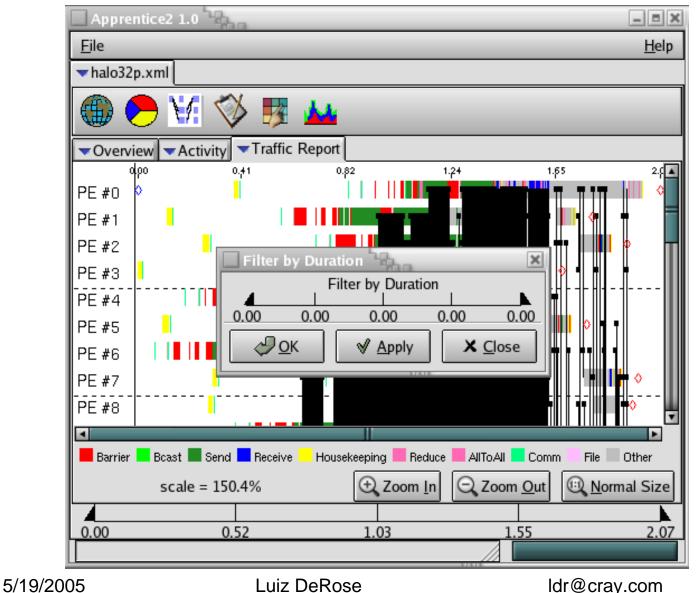
### **Timeline View**





### **Timeline View**

IRAY

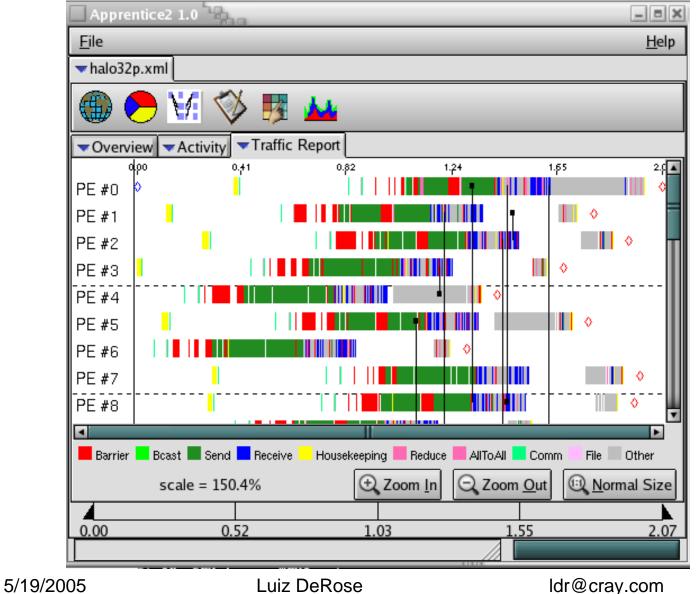






### **Timeline** View

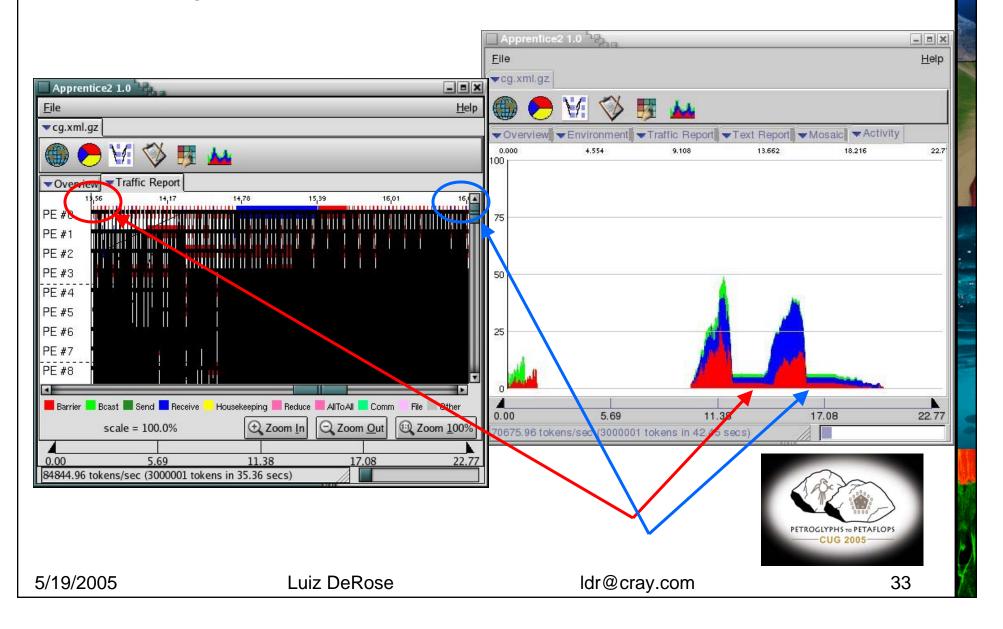
- Ray





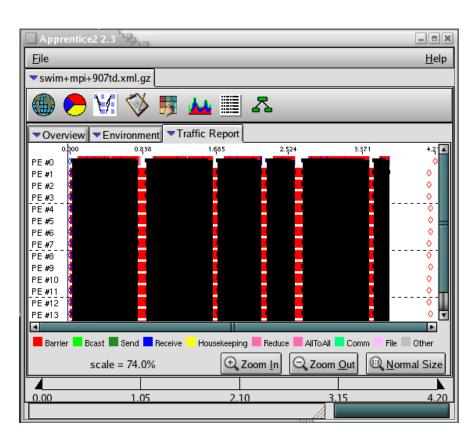
32

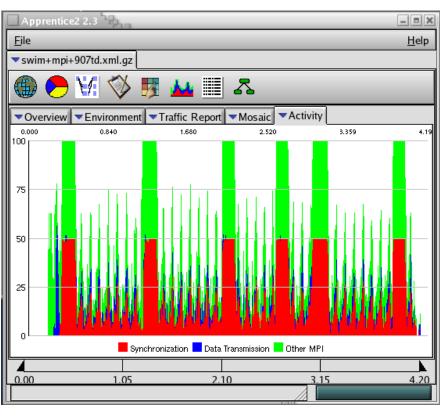
### **Activity Panel**



### **Timeline View**

ERAY





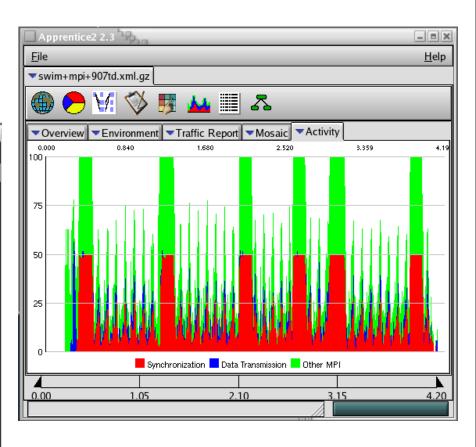


#### 5/19/2005

### **Timeline View**

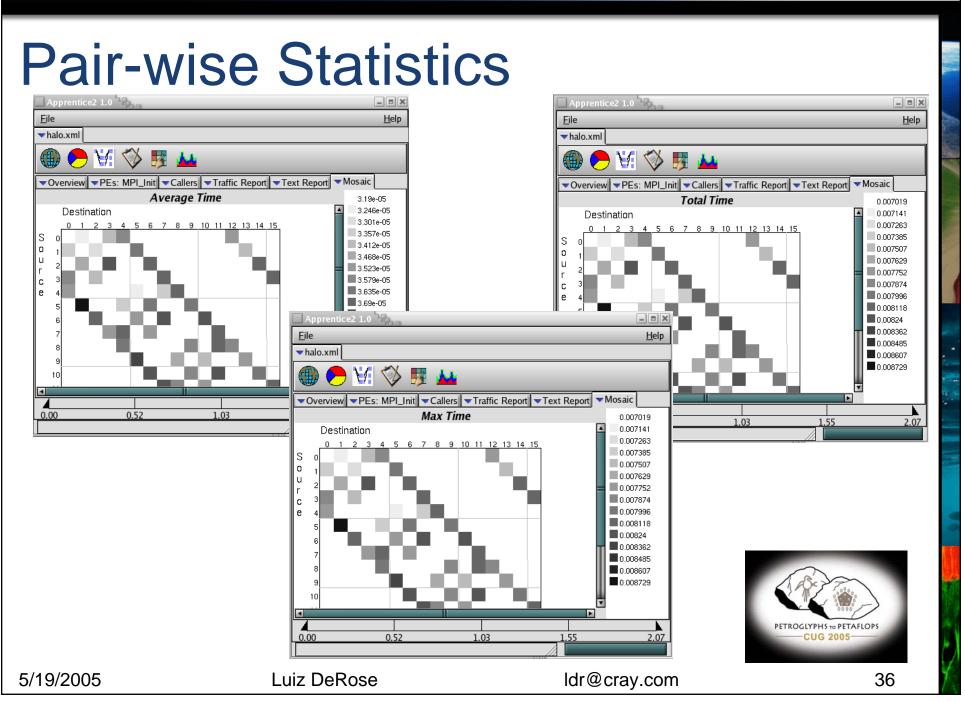
BAY

Apprentice2 2.3	- <b>=</b> ×
<u>File</u>	<u>H</u> elp
▼swim+mpi+907td.xml.gz	
🌐 🗩 🧐 😻 🎫 🏬 🏝	
▼Overview ▼Environment ▼Traffic Report	
0.855 1.034 1.113 1.192 1.272 PE#0 ###################################	1.3!
PE#3_ PE#4_ PE#4_ PE#4_ PE#6_ PE#6_ PE#7_	
PE#7	
PE #12 PE #13	
PE#12 74.1111 74.1111 74.1111 74.1111 74.1111 74.1111 74.1111 74.1111 74.1111 74.1111 74.1111 74.1111 74.1111 74.1111 74.1111 74.1111 74.1111 74.1111 74.11111 74.1111 74.	Ţ
Barrier EBcast ESend EReceive Housekeeping Reduce AllToAll Comm File Of	ther
scale = 786.8% 🕀 Zoom In 📿 Zoom Out	Size
0.00 1.05 2.10 3.15	4.20





#### 5/19/2005



#### The Supercomputer Company

### **Run Information and On-line Help**

T  Profile Text Report Environment Environment Env Vars System Info Resource Limits Name Value	Eile	<u>H</u> el
Environment Env Vars System Info Resource Limits		
Env Vars System Info Resource Limits		
Name Value	A REAL PROPERTY AND A REAL	
Value	1. Introduction	APPRENTICE2 - Performance Visualization
USER derose	2. Instrumenting Your Co	
LOGNAME derose	8	
HOME /dfs/home/de	CONTRACTOR CONTRACTOR CONTRACTOR CONTRACTOR	
PATH /opt/ctl/crayli	2 2 I race-based MPL EX	
MAIL /var/mail/de SHELL /bin/tcsh	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1. Introduction
TZ ESTSEDT	3. Using Apprentice2	
SSH_CLIENT 136.162.12.1	0 4852 2 3.1 Panel Summaries	Apprentice2 is an X-windows based performance tool for
SSH_CONNECTION 136.162.12.1	16	visualizing data produced by CrayPat on the Cray X1 (and
SSH_TTY /dev/ttyq7	2	others). It has a notebook-style tabbed interface that displays
TERM xterm	3.2.1 Limiting Input Data	a variety of data panels based upon the type of performance
DISPLAY phoenix:20.0	3.2.2 Informational popu	debugging that is being done. It does a reasonable job of
HOSTTYPE cray		
VENDOR cray	3.2.3 Remapping Filenar	detecting, from the datafile, which type of debugging needs
OSTYPE unicosmp	4. Example Data Files	to be done. Some of the modes of debugging have 'caliper
MACHTYPE cray_x1	1	points' that all the user to dynamically limit the scope of the
SHLVL 1 PWD /dfs/home/de	5. Troubleshooting	performance debugging being done and the current (and
GROUP users	5.1 I grabbed a copy a	future) data panels are updated on the fly to reflect the new
HOST phoenix		limits.
REMOTEHOST ip-136-162-12	5.2 I get cryptic warnin	
	6. Known Bugs	Apprentice2 currently supports two types of PAT performance data, callstack sampling and trace-based MPI experiments.
		Tip: Much of the panel specific functionality is located on the
	68675.69 tokens/sec (300	0001 tokens in 43.68 secs)

### **CRAY**

### **New Features Summary**

- Improved usability of Cray PAT on the X1 Series
  - Cray Apprentice<sup>2</sup>
  - pat\_run
  - Linux Cross Compiler environment support
  - Improved performance of trace generation
  - Runtime summarization
  - Better OpenMP support
- Cray XT3 and XD1
  - Port of Cray PAT to both platforms
    - Beta available by the end of the year
  - Cray Apprentice<sup>2</sup>





### Conclusions

- Previous status
  - Good infrastructure
    - Lacking user interface
    - Lacking support on all platforms
- Current status
  - State of the art tools on all Cray systems
    - Functionality to answer all questions in the performance tuning and optimization cycle

### Near future

• Best performance analysis tools in the industry

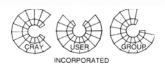


### The New Generation of Cray Performance Tools

# Questions? / Comments!

### **Thank You!**

Luiz DeRose



ldr@cray.com