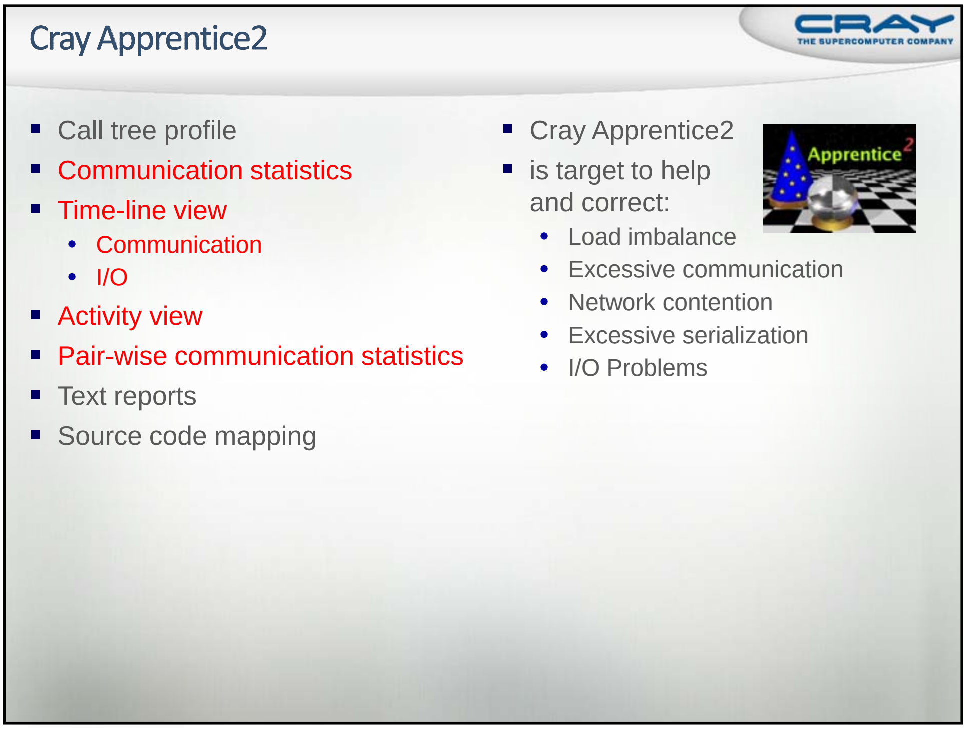
Trace Analysis

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

CSC, Finland

September 21-24, 2009

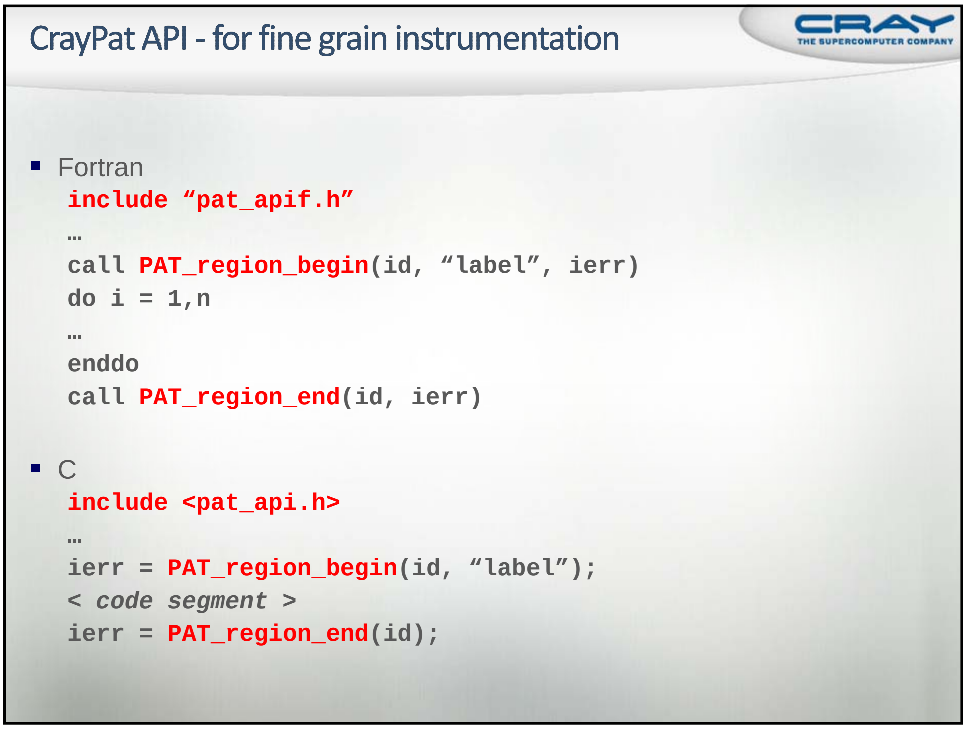


Cray Apprentice2

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

- Cray Apprentice2
- is target to help and correct:
  - Load imbalance
  - Excessive communication
  - Network contention
  - Excessive serialization
  - I/O Problems

Apprentice2



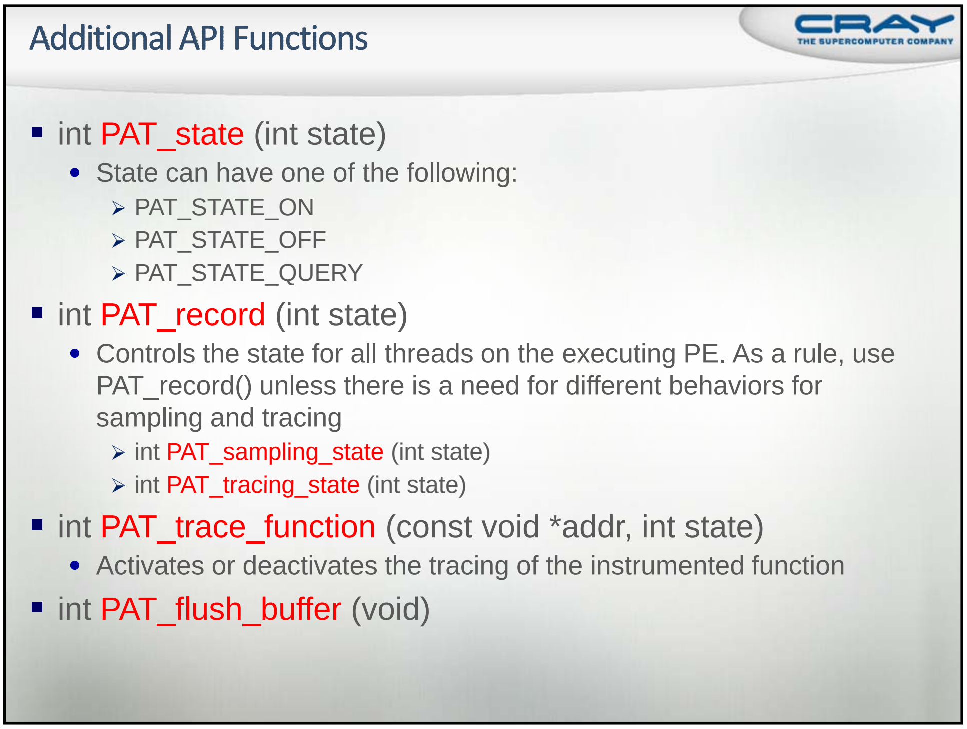
CrayPat API - for fine grain instrumentation

- Fortran

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

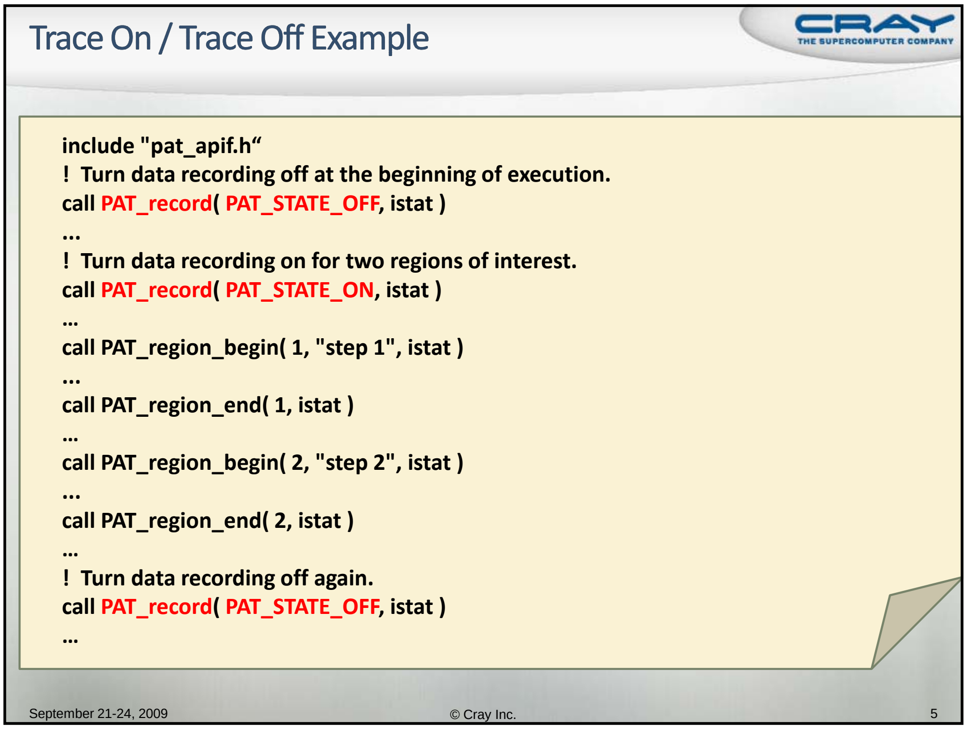
- C

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



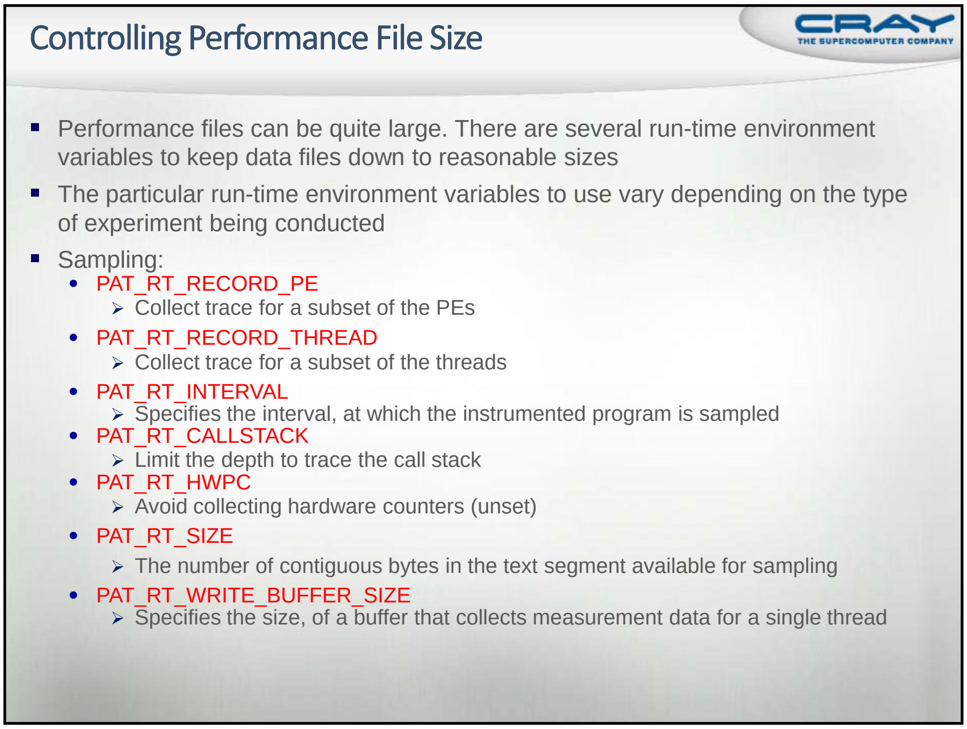
Additional API Functions

- int PAT_state (int state)
  - State can have one of the following:
    - PAT_STATE_ON
    - PAT_STATE_OFF
    - PAT_STATE_QUERY
- int PAT_record (int state)
  - Controls the state for all threads on the executing PE. As a rule, use PAT_record() unless there is a need for different behaviors for sampling and tracing
    - int PAT_sampling_state (int state)
    - int PAT_tracing_state (int state)
- int PAT_trace_function (const void *addr, int state)
  - Activates or deactivates the tracing of the instrumented function
- int PAT_flush_buffer (void)



Trace On / Trace Off Example

```
include "pat_apif.h“
! Turn data recording off at the beginning of execution.
call PAT_record( PAT_STATE_OFF, istat )
…
! Turn data recording on for two regions of interest.
call PAT_record( PAT_STATE_ON, istat )
…
call PAT_region_begin( 1, "step 1", istat )
…
call PAT_region_end( 1, istat )
…
call PAT_region_begin( 2, "step 2", istat )
…
call PAT_region_end( 2, istat )
…
! Turn data recording off again.
call PAT_record( PAT_STATE_OFF, istat )
…
```



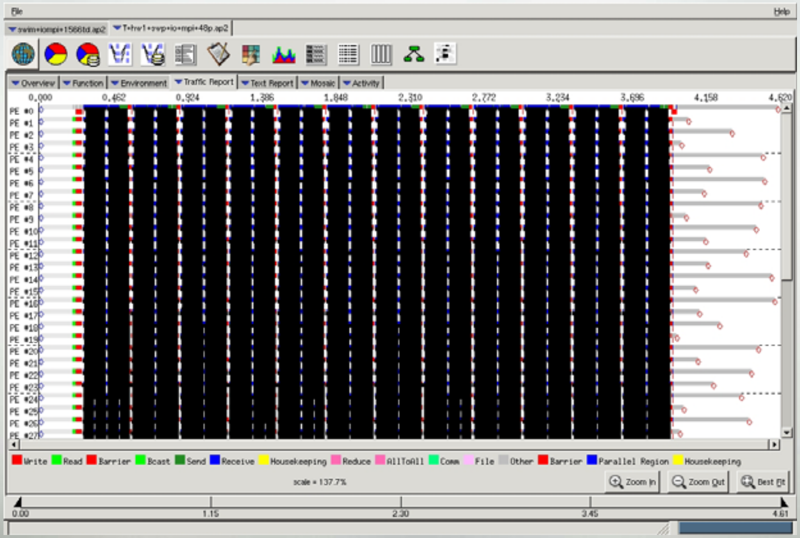
Controlling Performance File Size

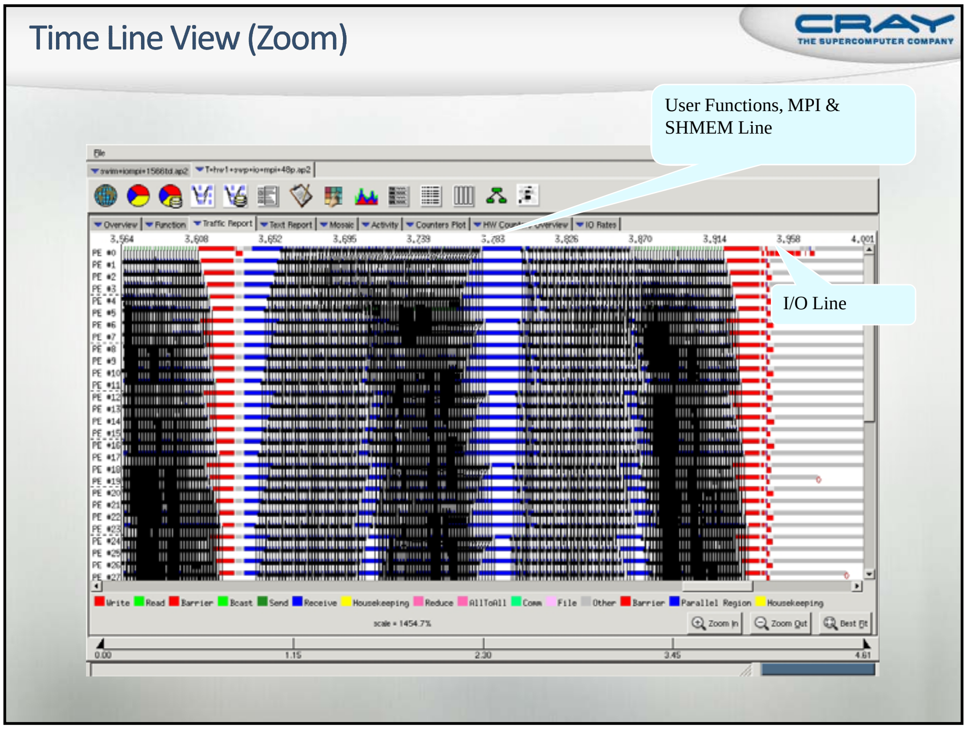
- Performance files can be quite large. There are several run-time environment variables to keep data files down to reasonable sizes
- The particular run-time environment variables to use vary depending on the type of experiment being conducted
- Sampling:
  - PAT_RT_RECORD_PE
    - Collect trace for a subset of the PEs
  - PAT_RT_RECORD_THREAD
    - Collect trace for a subset of the threads
  - PAT_RT_INTERVAL
    - Specifies the interval, at which the instrumented program is sampled
  - PAT_RT_CALLSTACK
    - Limit the depth to trace the call stack
  - PAT_RT_HWPC
    - Avoid collecting hardware counters (unset)
  - PAT_RT_SIZE
    - The number of contiguous bytes in the text segment available for sampling
  - PAT_RT_WRITE_BUFFER_SIZE
    - Specifies the size, of a buffer that collects measurement data for a single thread

Controlling Trace File Size

- Tracing:
  - PAT_RT_CALLSTACK
    - Limit the depth to trace the call stack
  - PAT_RT_HWPC
    - Avoid collecting hardware counters (unset)
  - PAT_RT_RECORD_PE
    - Collect trace for a subset of the PEs
  - PAT_RT_RECORD_THREAD
    - Collect trace for a subset of the threads
  - PAT_RT_TRACE_FUNCTION_ARGS
    - Limit the number of function arguments to be traced
  - PAT_RT_TRACE_FUNCTION_LIMITS
    - Avoid tracing indicated functions
  - PAT_RT_TRACE_FUNCTION_MAX
    - Limit the maximum number of traces generated for all functions for a single process
  - PAT_RT_TRACE_THRESHOLD_PCT
    - Specifies a % of time threshold to enforce when executing in full trace mode
  - PAT_RT_TRACE_THRESHOLD_TIME
    - Specifies a time threshold to enforce when executing in full trace mode
- Use the limit built-in command for ksh(1) or csh(1) to control how much disk space the trace file can consume

Time Line View (Sweep3D)

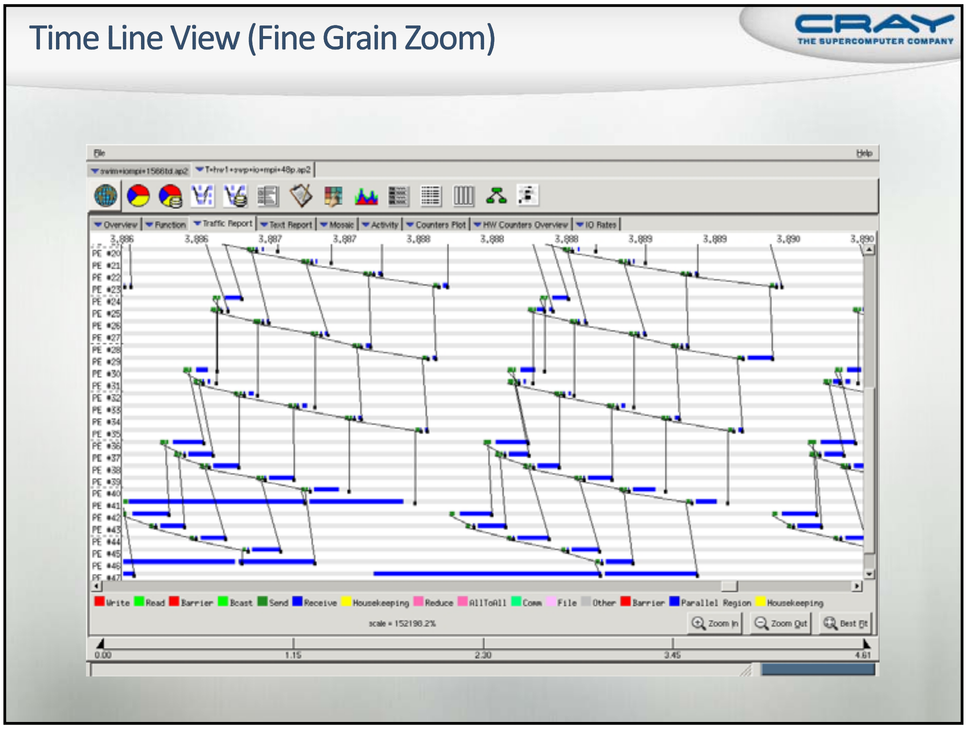




Time Line View (Zoom)

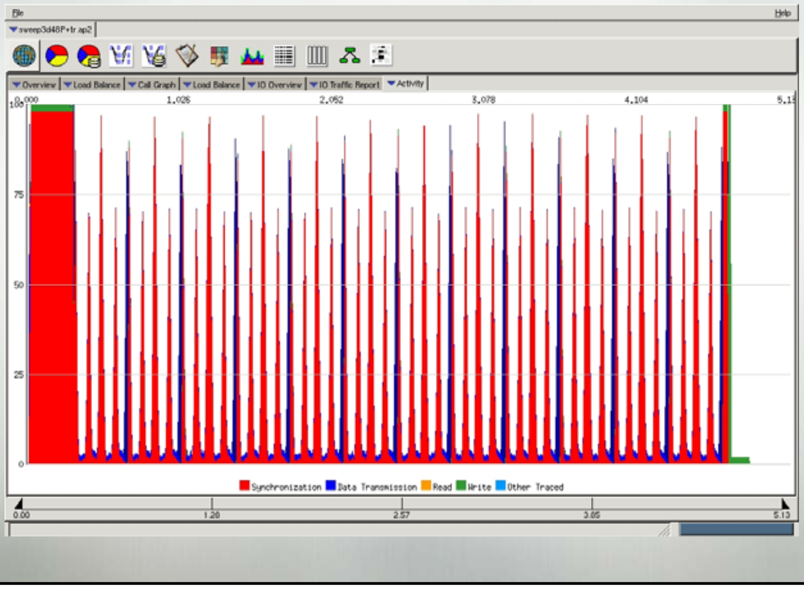
User Functions, MPI & SHMEM Line

I/O Line

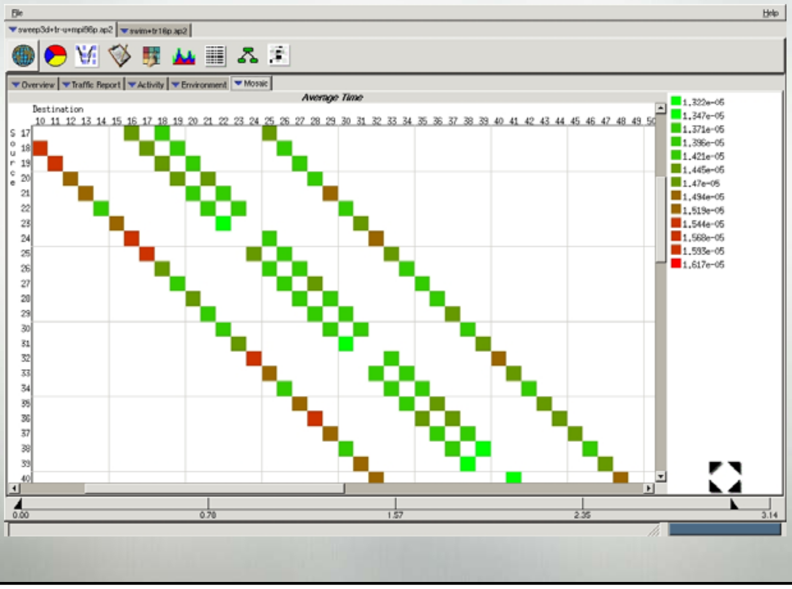


Time Line View (Fine Grain Zoom)

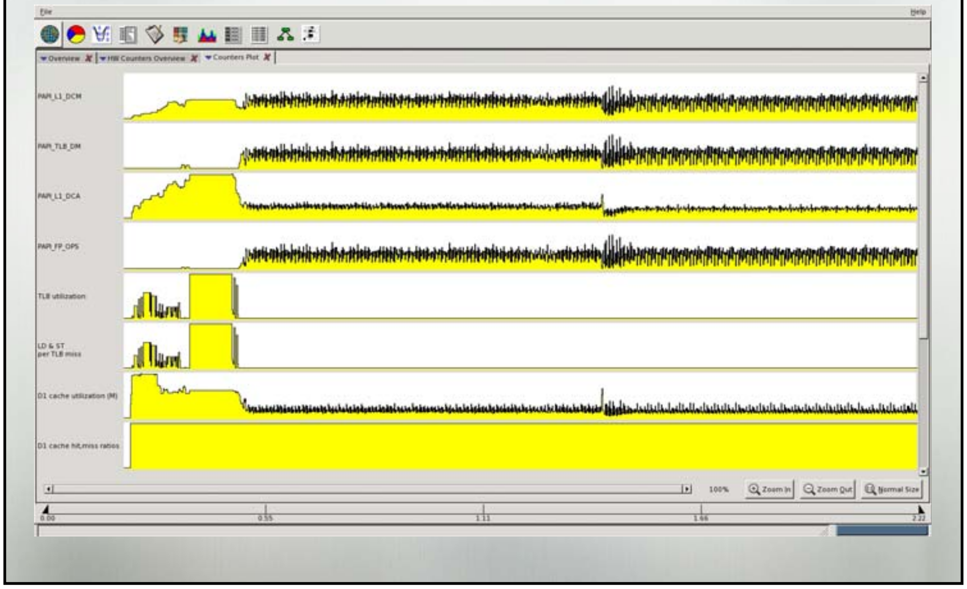
Activity View



Pair-wise Communication



Hardware Counters Time Line



Trace Analysis

Questions / Comments

Thank You!

CSC, Finland © Cray Inc. September 21-24, 2009