

Intel Trace Analyzer / Collector Introduction and Tutorial

Ullrich Becker-Lemgau

Intel GmbH, Solutions & Software Group

ullrich.becker-lemgau@intel.com

October 18, 2004

Istanbul Technical University

Istanbul, Turkey



Intel Trace Analyzer / Collector

—

scalable performance analysis and visualization for parallel MPI programs

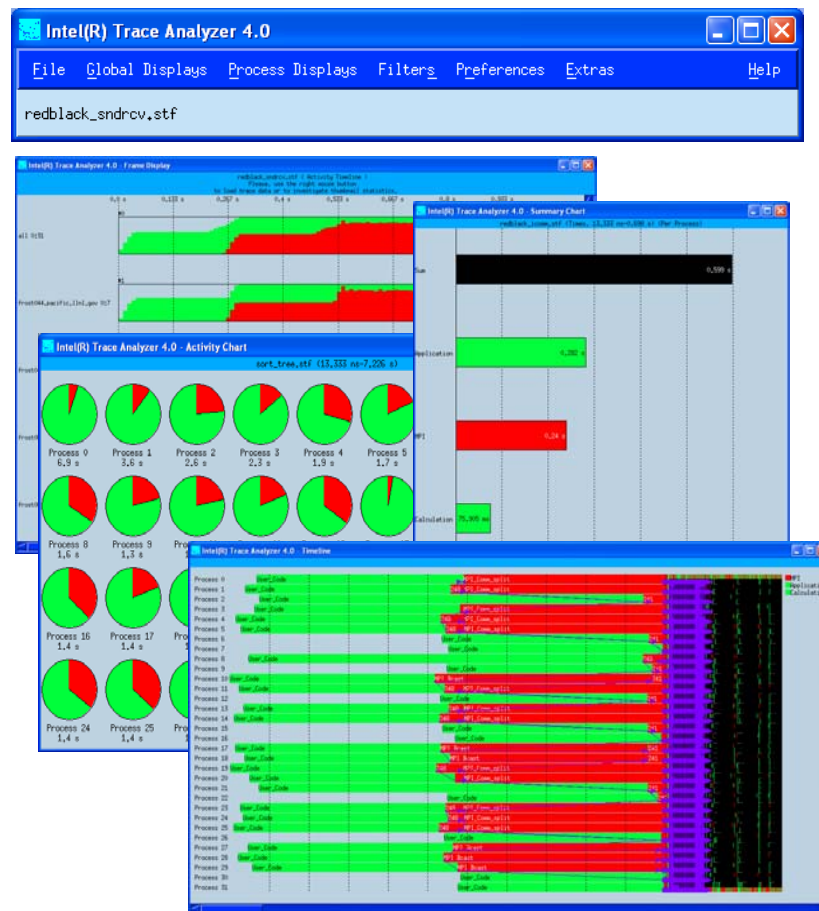


Agenda

- Introduction to Intel Trace Analyzer (ITA)
 - concept, GUI, examples traces, BKM's
- Introduction to Intel Trace Collector (ITC)
 - concept, compiling, linking
 - default tracing
 - recording names of functions or regions
 - recording source-code-locations
 - recording performance counter data
 - selective tracing, filtering
 - handling structured tracefiles
- Using ITC and ITA with a simple application.

Visualization and Analysis of MPI Programs

ITA is the visualization and analysis tool for ITC



Introduction to Intel Trace Analyzer / Collector

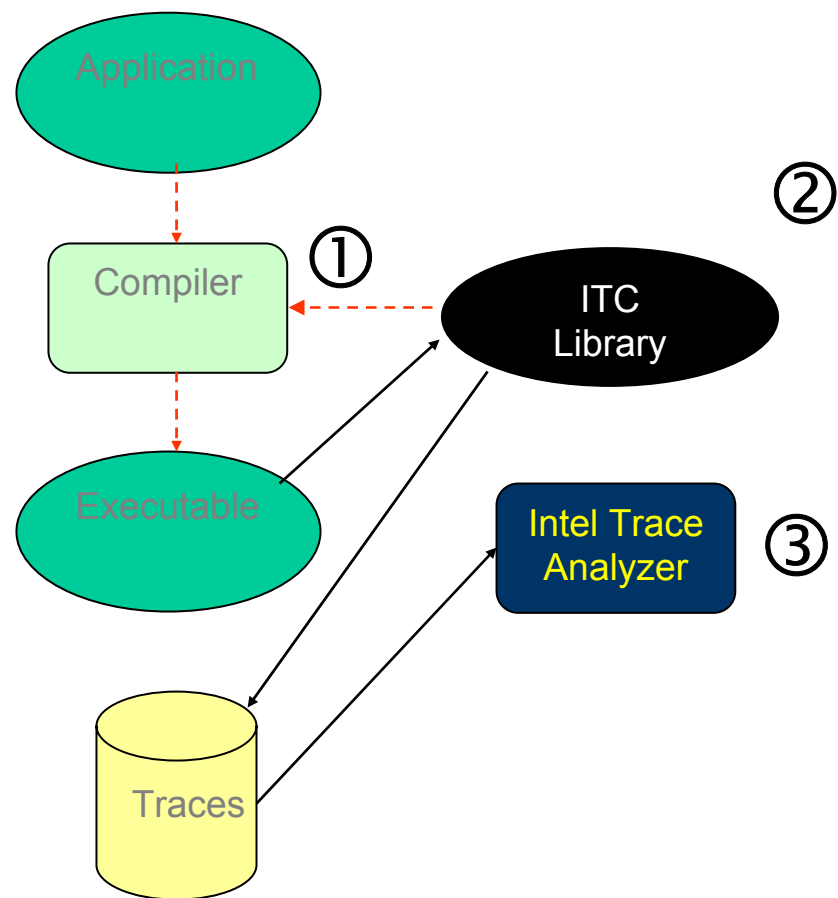
- What is the Intel Trace Analyzer/Collector?
 - It is a tool that focuses on instrumentation for two parallel programming paradigms for High Performance Computing (HPC), namely:
 - OpenMP
 - MPI
- ITA/ITC provides the following features:
 - Ease of Use – basic how-to's, to apply to your development
 - Integration – from default performance analysis to mastering the workbench, e.g. instrumenting functions, extracting hardware monitors, detailing the runtime environment
 - Scalability – works with 10,000+ Processors, how to manage more or less performance data

Intel Trace Analyzer/Collector Features

- Offline trace analysis for MPI (and others ...)
- Traces generated by Intel Trace Collector tool (``ld ... -lVT -lpmpi -lmpi``)
- **Scalability** in time and processor-space (STF)
- High-performance graphics, excellent **zooming** and **filtering**
- Display and analysis of **MPI** and **application** events:
 - execution of **MPI** routines, p2p, and collectives
 - MPI-2 I/O operations
 - performance counter displays (events, PAPI, ...)
 - execution of application subroutines (optional, gnu-func.)
- Easy customization

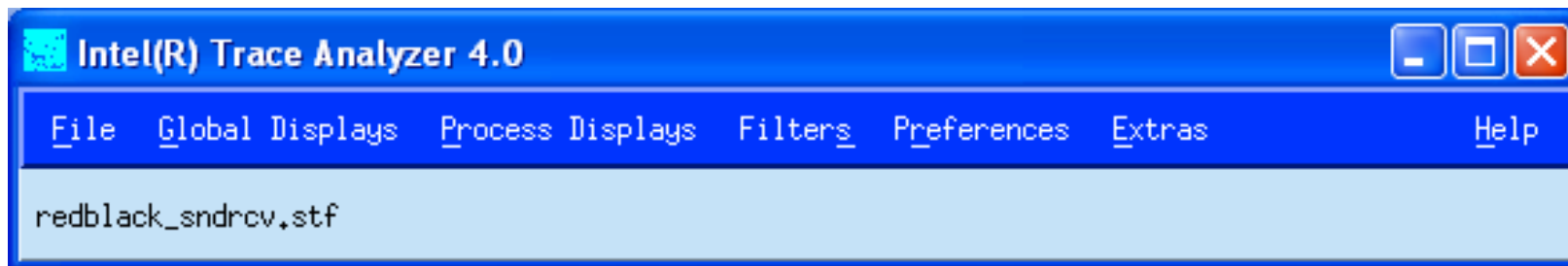
Intel Trace Analyzer/Collector Components and Interaction

- Compiler
 - MPI, ...
 - ITC API
- Intel Trace Collector
 - Data collection
 - Run-time configuration
- Intel Trace Analyzer
 - MPI analysis



Intel Trace Analyzer Main Window

Intel Trace Analyzer main window:

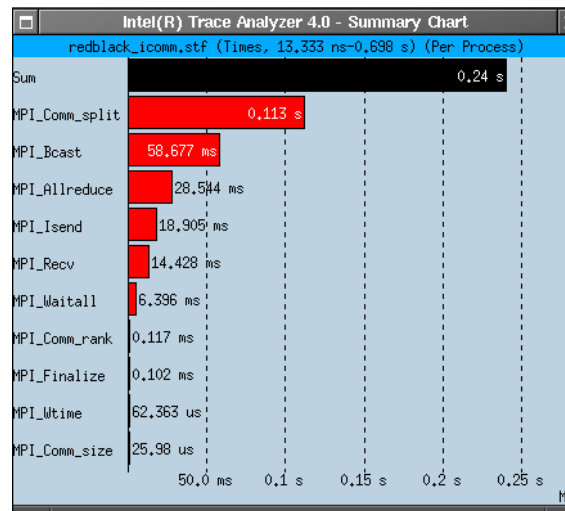
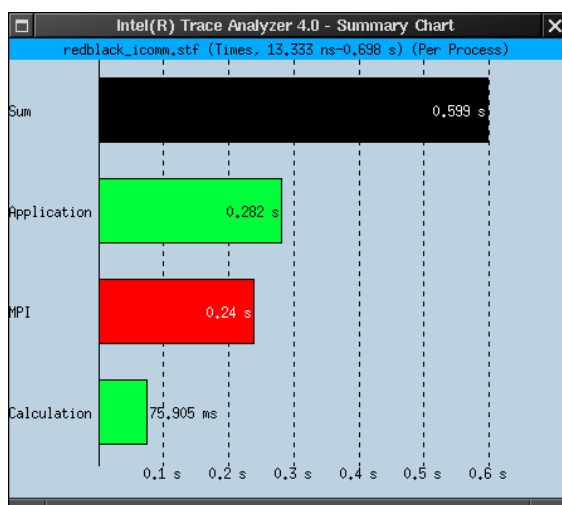


- Tracefile loading can be interrupted at any time
- Tracefile loading can be resumed
- Tracefile can be loaded starting at a specified time offset
- Tracefile can be re-written (re-grouped symbols)

Intel Trace Analyzer Displays

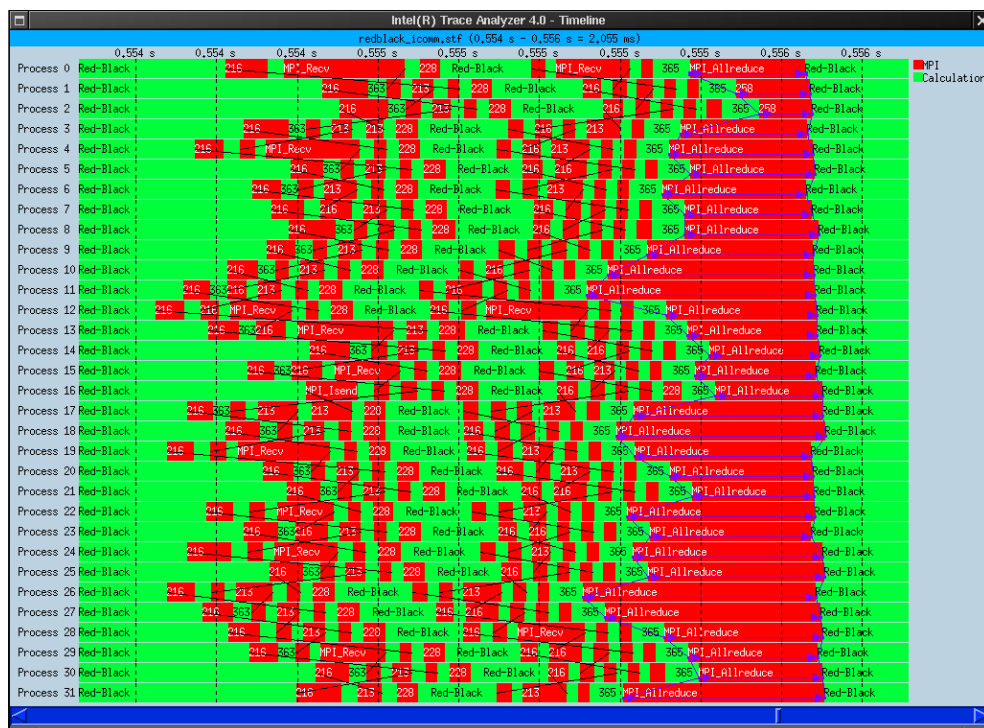
- **Global displays** show all selected processes
 - **Summary Chart**: aggregated profiling information
 - **Timeline**: detailed application execution over time axis
 - **Communication statistics**: message statistics for each process pair
 - **Activity Chart**: presents per-process profiling information
 - **Global Comm. Statistics**: collective operations statistics
 - **I/O Statistics**: MPI I/O operation statistics
 - **Calling Tree**: draws global or local dynamic calling trees
- **Process displays** show a single process per window
 - **Activity Chart**
 - **Timeline**
 - **Calling Tree**

Summary Chart



- Aggregated profiling information
 - execution time
 - number of calls
- **Inclusive** or **exclusive** of called routines
- Look at all/any category or all states
- Values can be exported/imported
- Tracefiles can be compared

Timeline Display



- Now displays MPI collective and I/O operations
- To zoom, draw rectangle with the mouse
- Also used to select sub-intervals for statistics

Timeline Display (Message In message line)

message send

message received

Intel(R) Trace Analyzer 4.0 - Timeline

redblack_icomm.stf (0,554 s - 0,556 s = 2,055 ms)

Process 0 Red-Black 216 MPI_Recv 228 Red-Black MPI_Recv 365 MPI_Allreduce Red-Black

Process 1 Red-Black 216 363 213 228 Red-Black 216 365 258 Red-Black

Process 2 Red-Black 216 363 213 228 Red-Black 216 365 258 Red-Black

Process 3 Red-Black 216 363 213 228 Red-Black 216 213 365 MPI_Allreduce Red-Black

Process 4 Red-Black 216 MPI_Recv 228 Red-Black 216 213 365 MPI_Allreduce Red-Black

Process 5 Red-Black 216 363 213 228 Red-Black 216 213 365 MPI_Allreduce Red-Black

Process 6 Red-Black 216 MPI_Recv 228 Red-Black 216 213 365 MPI_Allreduce Red-Black

Intel(R) Trace Analyzer 4.0 - Identified Message

Intel(R) Trace Analyzer 4.0 - Source View Process 21

```

pardat_icomm.f90: Line 136, Col 1
130 ic0=ic+1
131 DO i=ifrst_n,imax,2
132   ic=ic+1
133   combuf(ic) = x(i,jmax)
134 ENDDO
135
136 CALL MPI_SEND(combuf(ic0), ic-ic0+1, MPI_DOUBLE_PRECISION, neigh_n,
137             Communicator, request(1), ierr)
138 CALL errmsg(ierr)
139
140 ENDIF
141
142
    
```

554557 s

Intel(R) Trace Analyzer 4.0 - Source View Process 22

```

pardat_icomm.f90: Line 235, Col 1
229
230
231 ENDIF
232
233 IF( neigh_s /= MPI_PROC_NULL ) THEN
234
235 CALL MPI_RECV(combuf(ptr_r), len_r, MPI_DOUBLE_PRECISION, neigh_s, 1000,
236             Communicator, status, ierr)
237 CALL errmsg(ierr)
238
239 ifrst_s = ifrst_s-1
240 IF( ifrst_s < imin ) THEN
241   ifrst_s = ifrst_s+2
    
```

additional information

Process 24 Red-Black 216 MPI_Recv 228 Red-Black

Process 25 Red-Black 216 363 213 228 Red-Black

Process 26 Red-Black 216 363 213 228 Red-Black

Process 27 Red-Black 216 363 213 228 Red-Black

Process 28 Red-Black 216 363 213 228 Red-Black

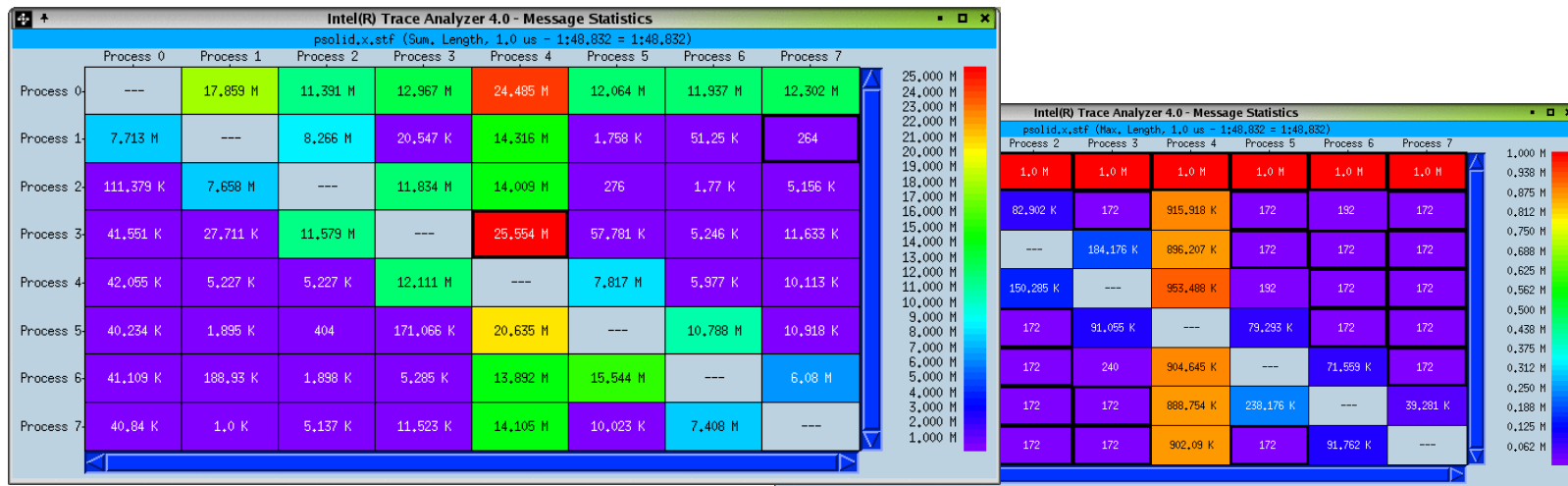
Process 29 Red-Black 216 363 213 228 Red-Black

Process 30 Red-Black 216 363 213 228 Red-Black

Process 31 Red-Black 216 213 228 Red-Black 213 365 MPI_Allreduce Red-Black



Communication Statistics



- Message statistics for each process pair:
 - Byte and message count
 - min/max/avg message length
 - min/max/avg bandwidth
- Filter for message tags or communicators

Intel Trace Collector

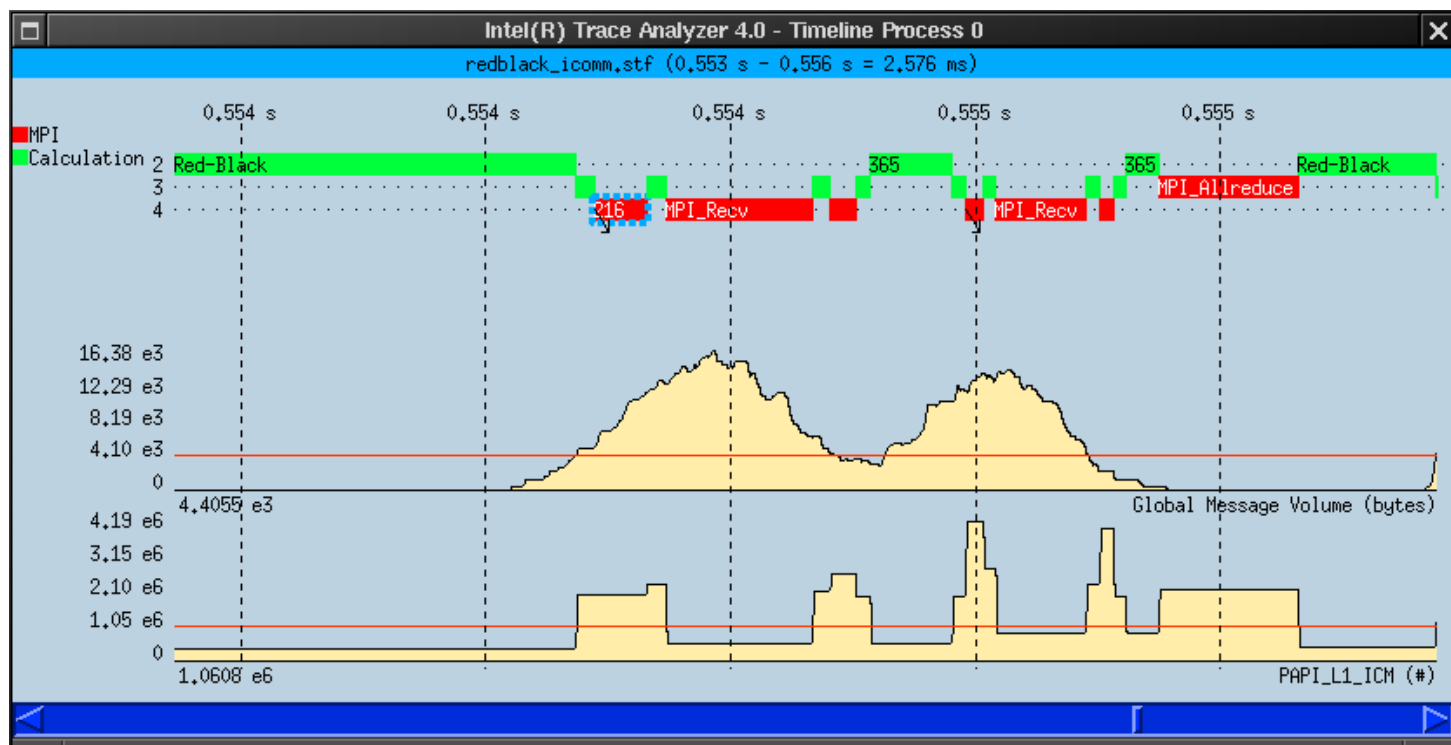
Tracing of
MPI and
Application
Events

Intel Trace Collector is
an event-based tracing
library for program
analysis



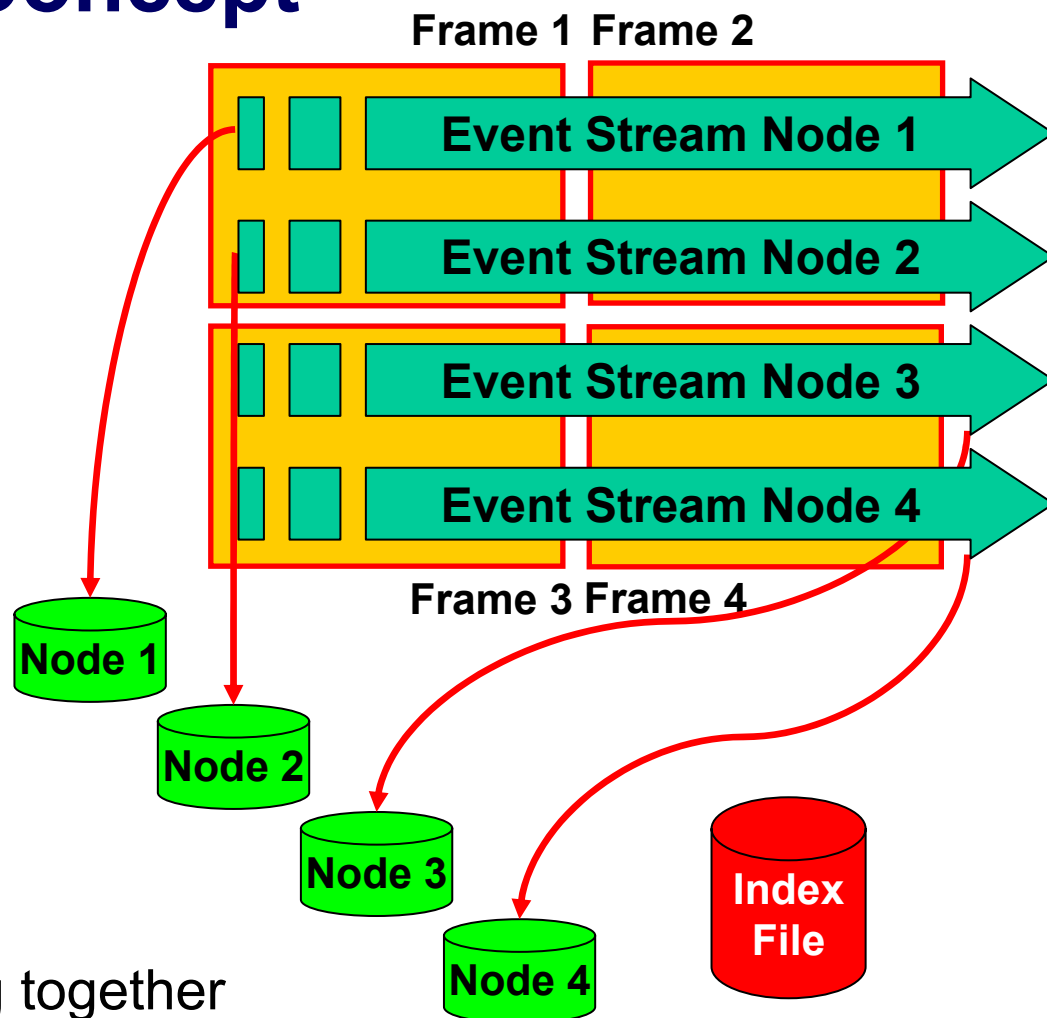
Counter Displays

- Intel Trace Analyzer *Counter Timeline* Display, e.g. using PAPI



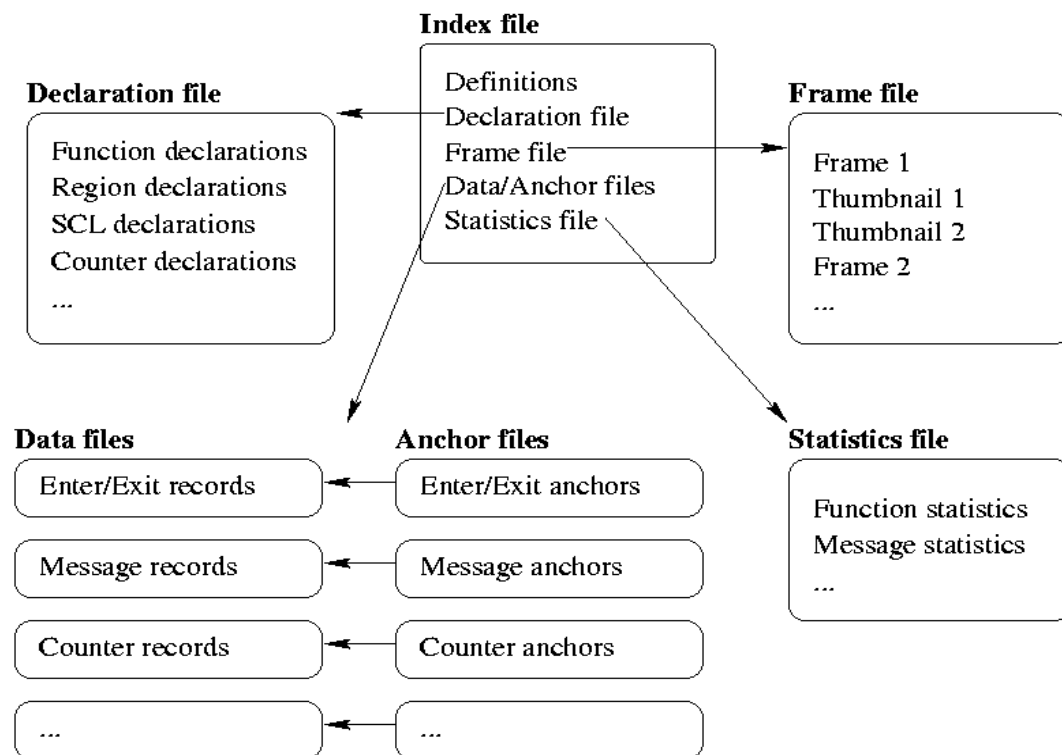
The Structured Tracefile Format – Concept

- Support *realistic* applications and configurations
- Improve trace file loading and navigation
- Subdivide trace into *frames*
- Compute statistics for each frame
- Stripe data across multiple files
- Index file* ties everything together



The Structured Tracefile Format – Implementation

- Components of a structured trace
 - one declaration, frame, index and statistics file
 - one data file per process (group)
 - one anchor file per data file



Intel Trace Analyzer / Collector Tutorial

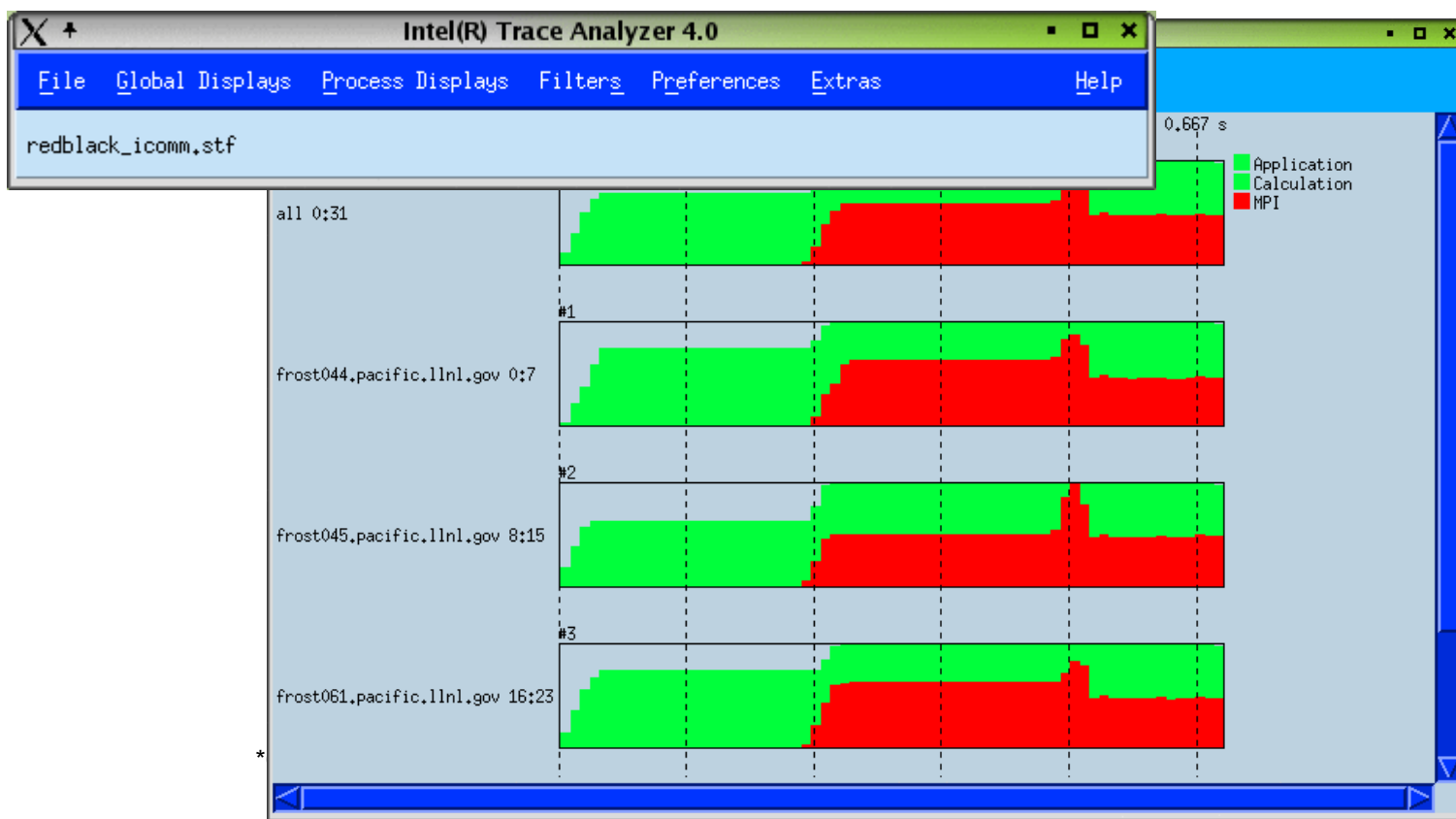


Installing Intel Trace Analyzer/Collector

- create directories 'ITA' and 'ITC'
- unpack tar-files in directories
- run install in both directories
- add '~/ITA/bin' to PATH
- set PAL_LICENSEFILE

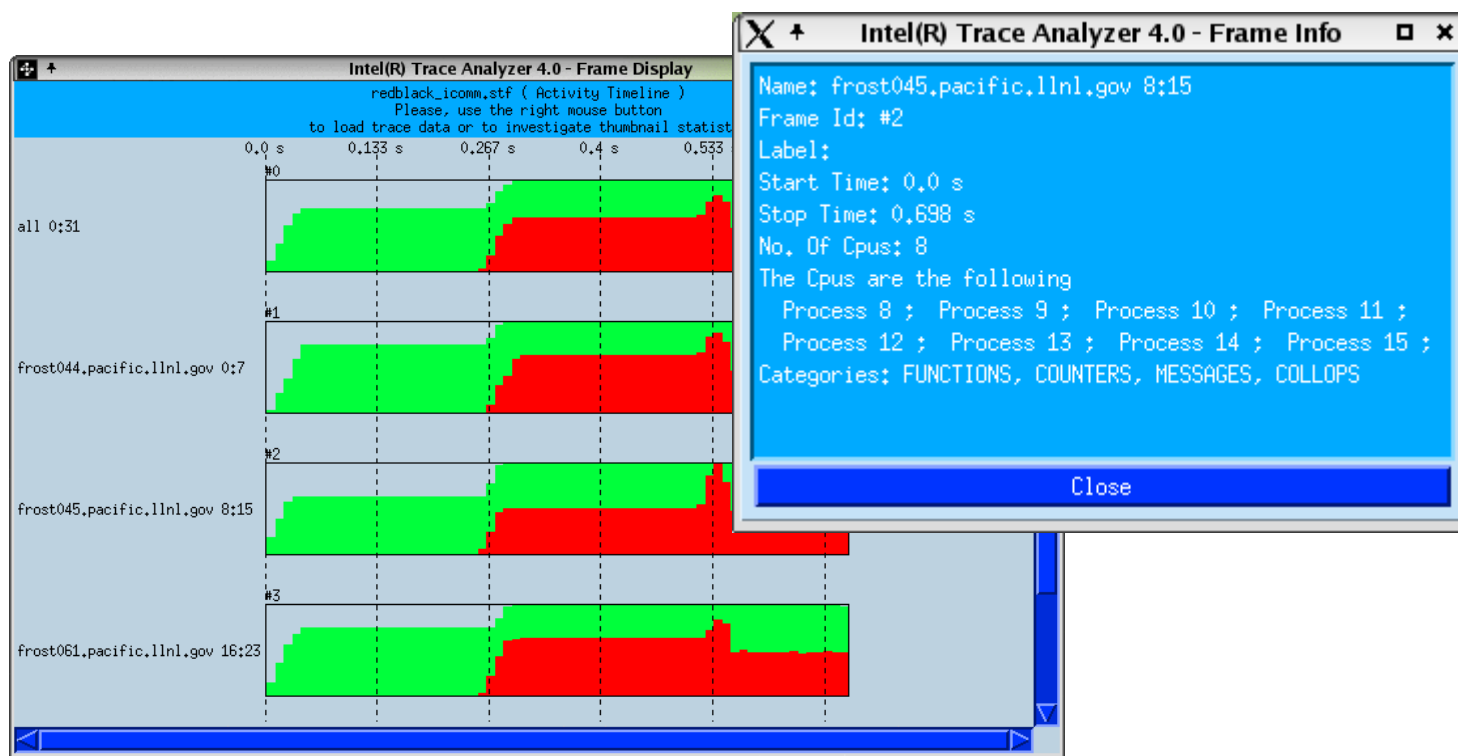
Intel Trace Analyzer – MPI Performance Analysis

- Invoke Intel Trace Analyzer: 'traceanalyzer'
- Select File → Open tracefile 'redblack_icomm.stf'



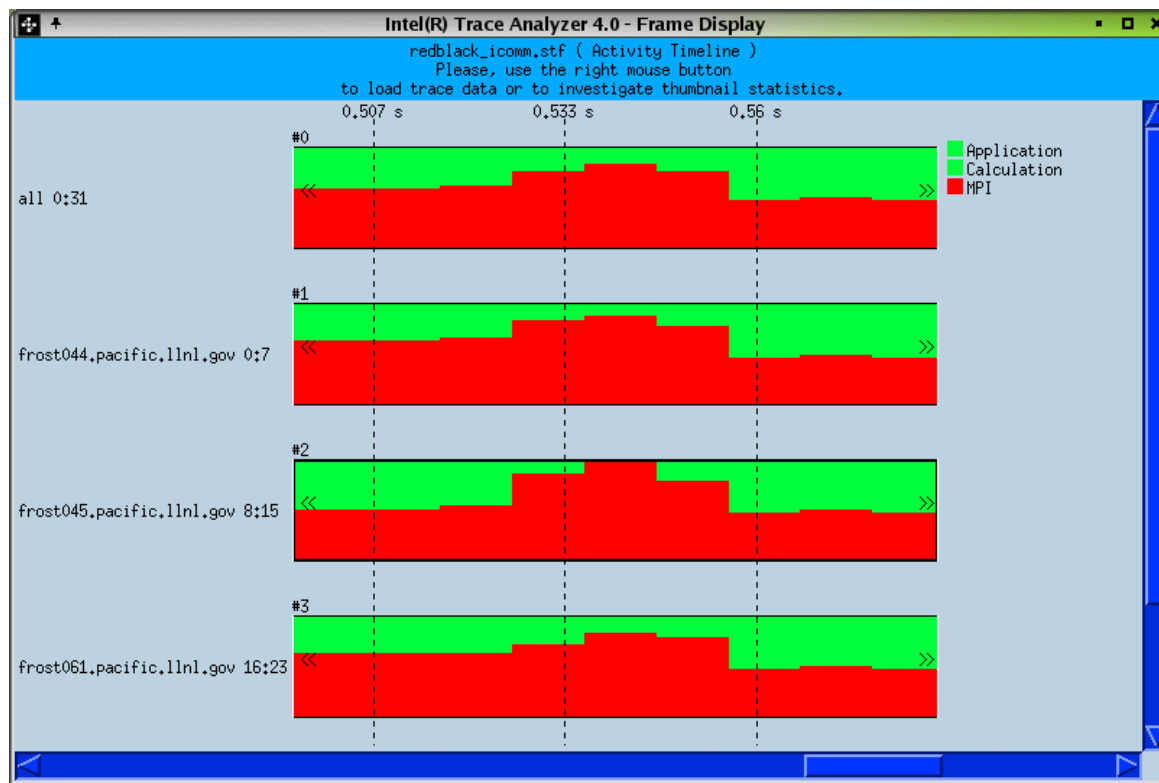
Intel Trace Analyzer – Looking at Frame Information

- e.g. Click on Frame and select Context Menu ⇒ Frame Info



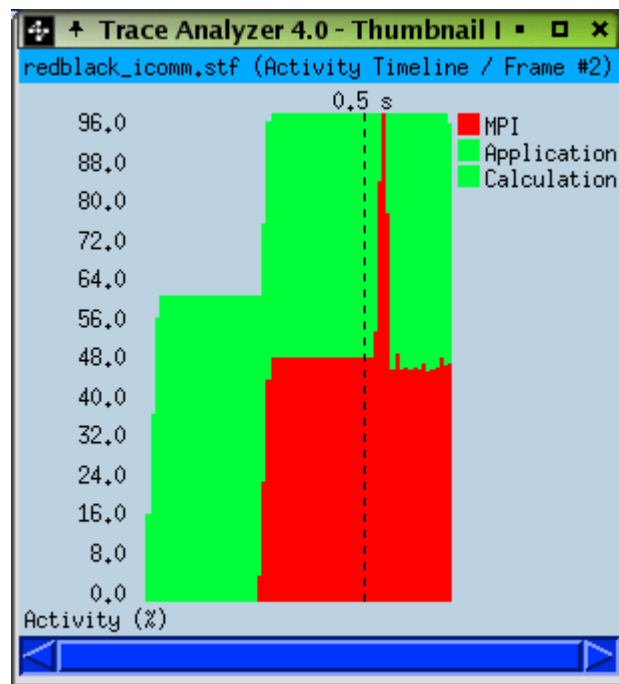
Intel Trace Analyzer – Looking at Frame Information

- Zoom to see more detail



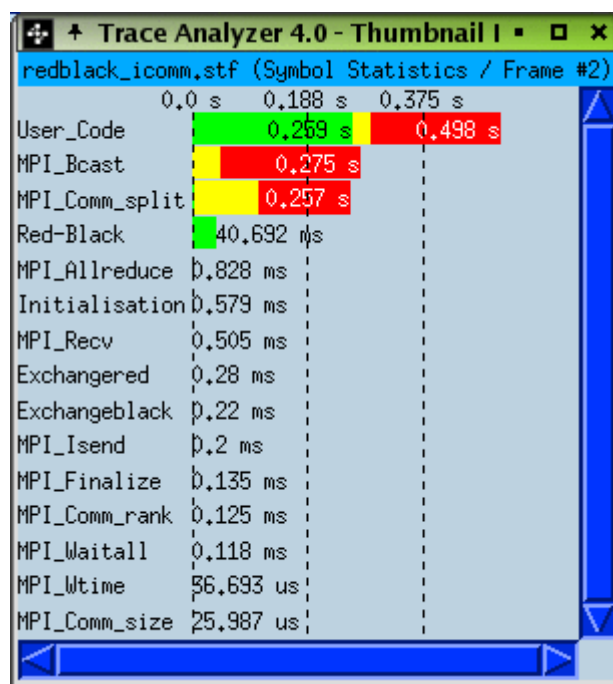
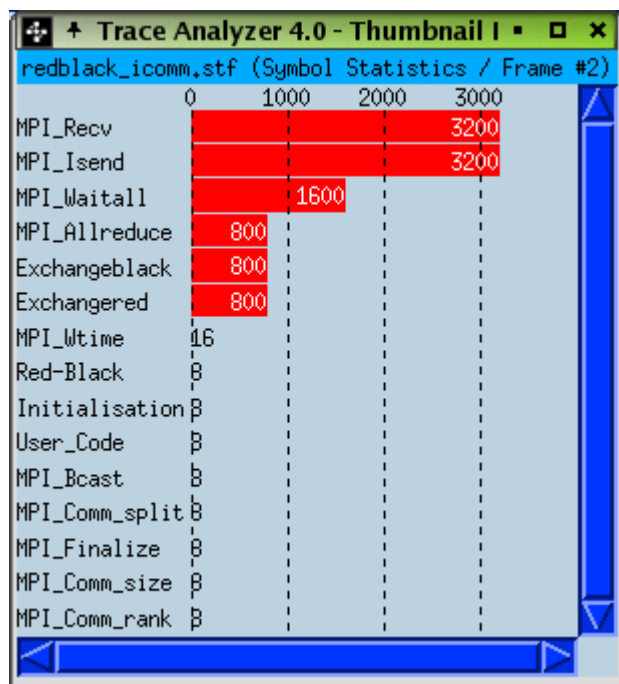
Intel Trace Analyzer – Precomputed STF Statistics

- Context Menu⇒Open Thumbnail
- Click on Frame



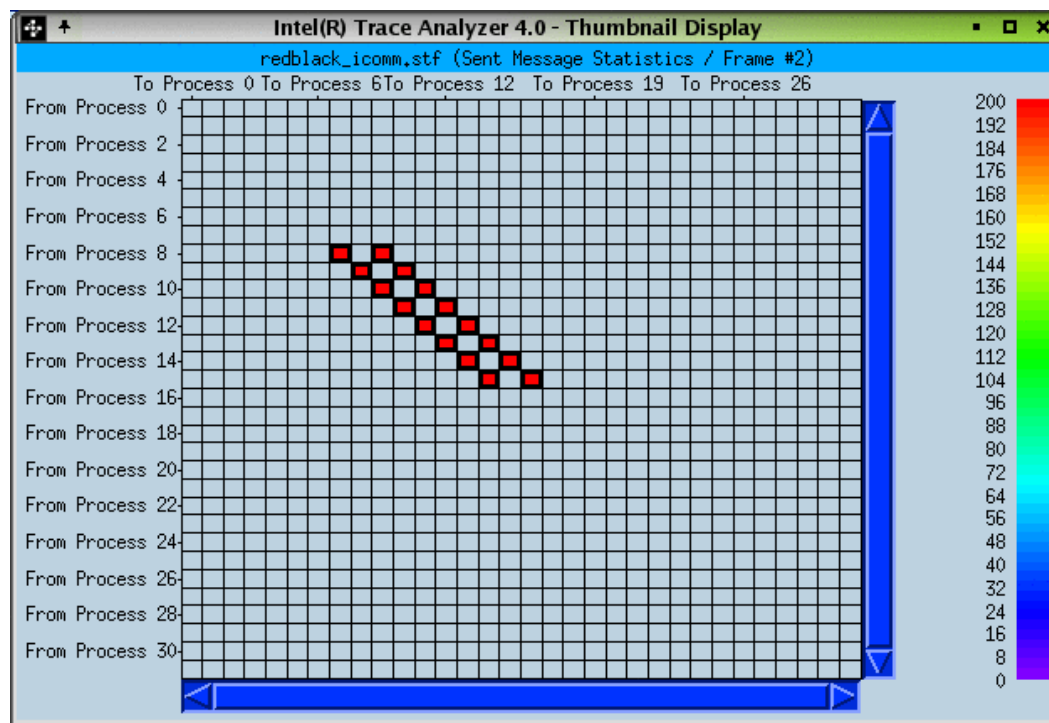
Intel Trace Analyzer – Precomputed STF Statistics

- Context Menu ⇒ Display ⇒ Symbol Statistics ⇒ Min + Avg + Max



Intel Trace Analyzer – Precomputed STF Statistics

- Context Menu ⇒ Display ⇒ Sent Message Statistics ⇒ Count



Intel Trace Analyzer – Loading Event Data

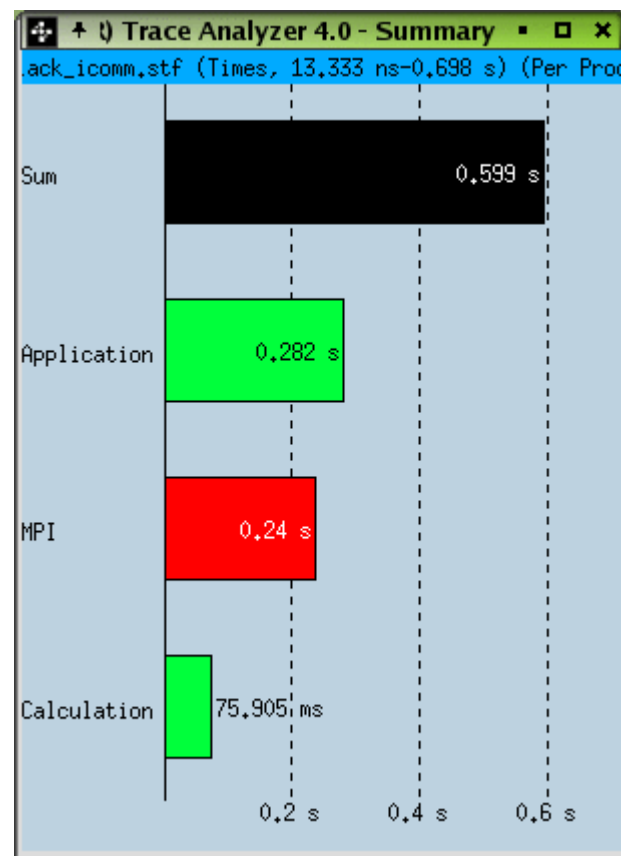
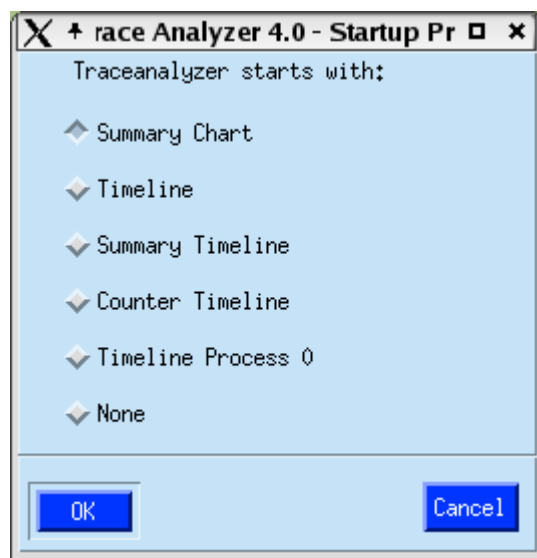
- Select frames by
 - Context Menu⇒Select/Deselect Frames and draw rectangle (toggles selection status)
 - Context Menu⇒Select All Frames

 - Load by Context Menu⇒Load Selected Frames

 - First "regular" Intel Trace Analyzer display pops up
 - Configure with Preferences⇒Displays⇒Startup with

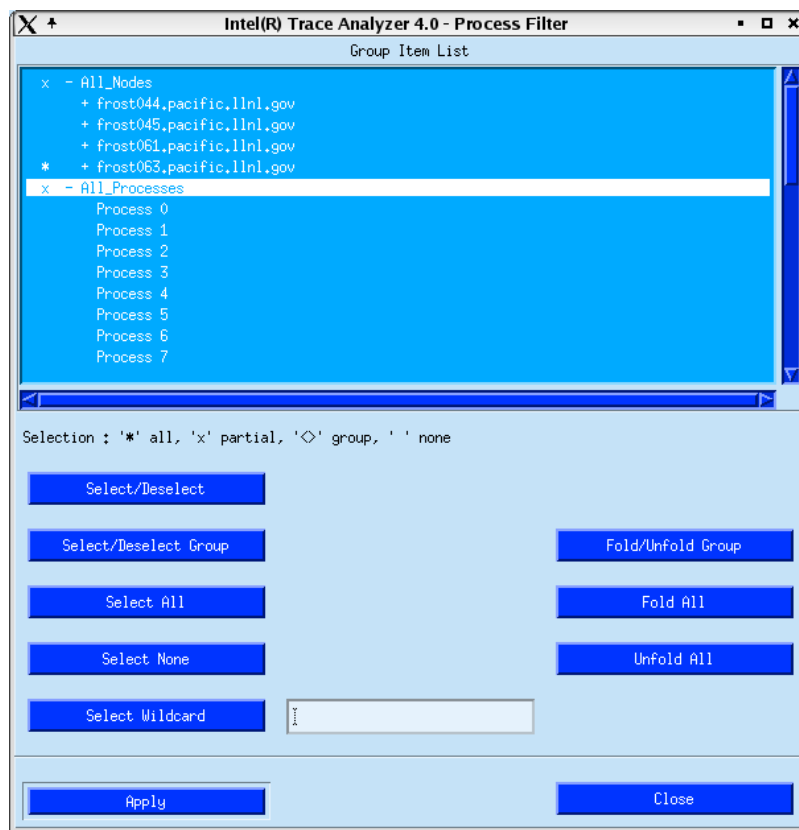
Intel Trace Analyzer – Loading Event Data

- Configure the Summary Chart



Intel Trace Analyzer – Process Grouping and Filtering

- Select Filters ⇒ Processes
- Look at list of defined groups

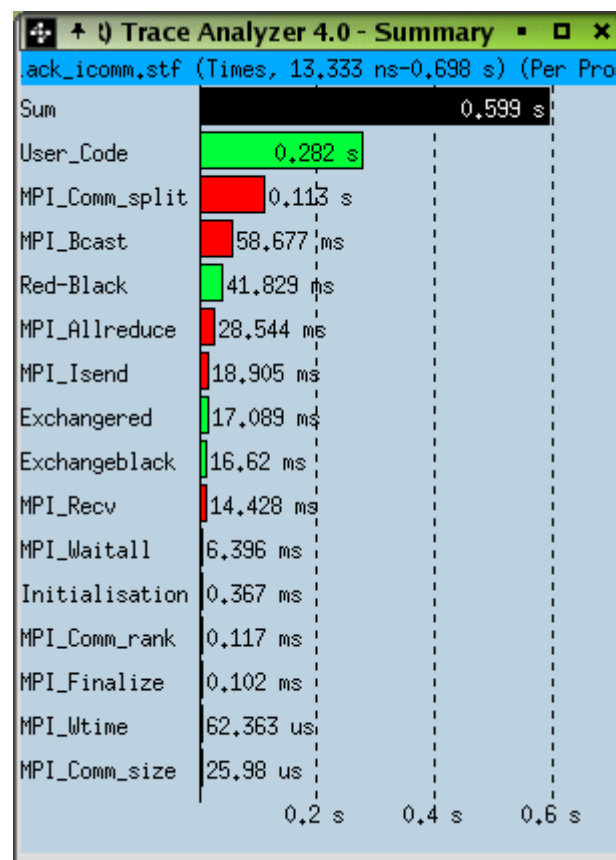


Intel Trace Analyzer – Routine Profiles and Statistics

- Setup Summary Chart

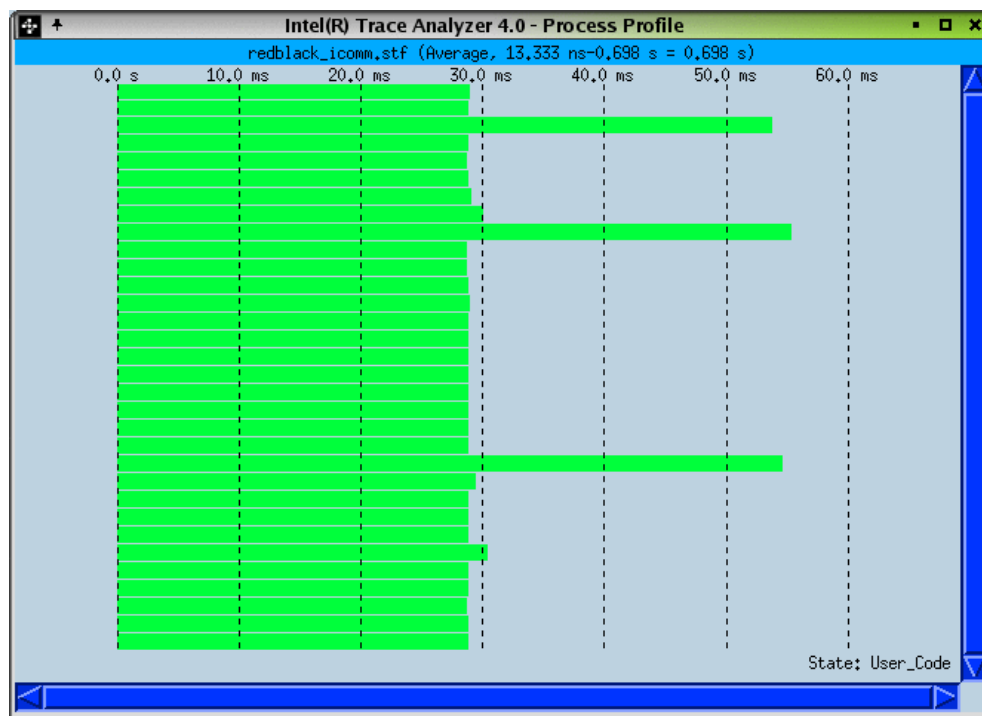
- Global Displays ⇒ Summary Chart
- Context Menu ⇒ Select ⇒ All Symbols
- Context Menu ⇒ Options ⇒ Per Process

- Selected group: All Master Threads



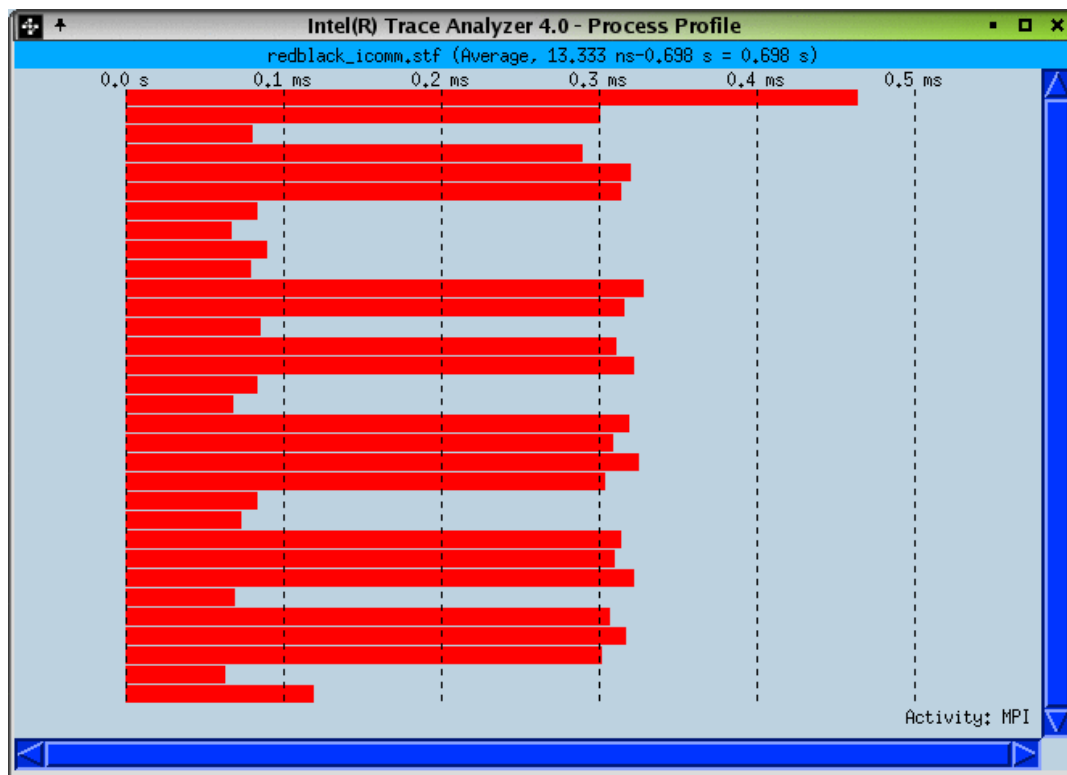
Intel Trace Analyzer – Load Balance Analysis

- Setup Process Profile
 - Global Displays ⇒ Process Profile



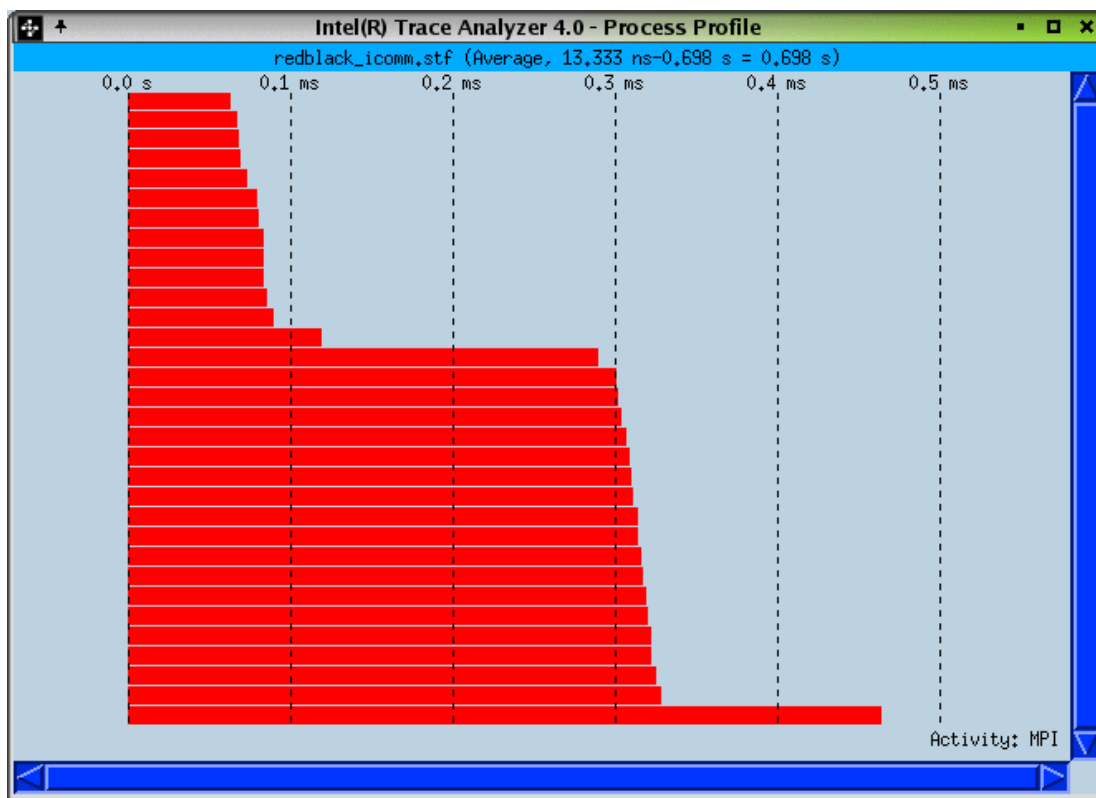
Intel Trace Analyzer – Load Balance Analysis

- Context Menu ⇒
Select ⇒ MPI



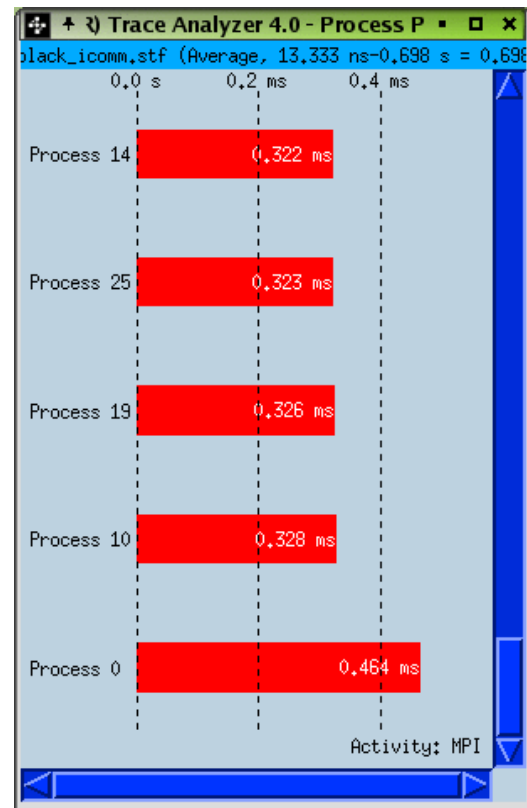
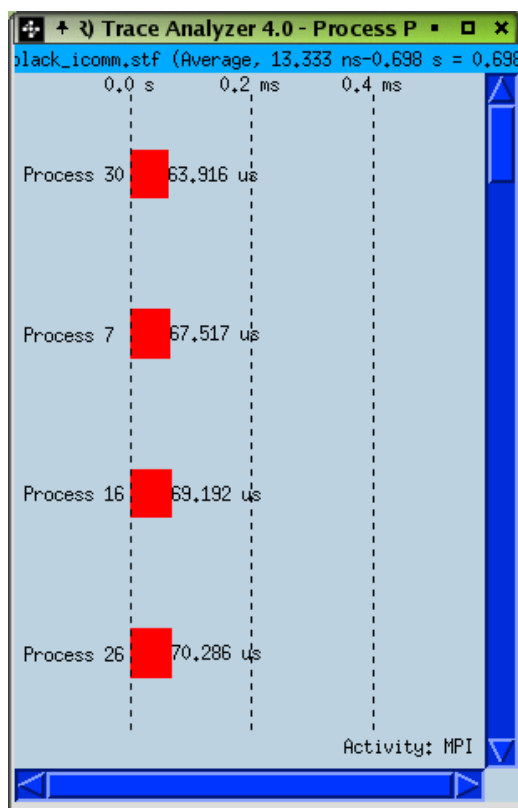
Intel Trace Analyzer – Load Balance Analysis

- Context Menu ⇒
Sort by ⇒ Value Up



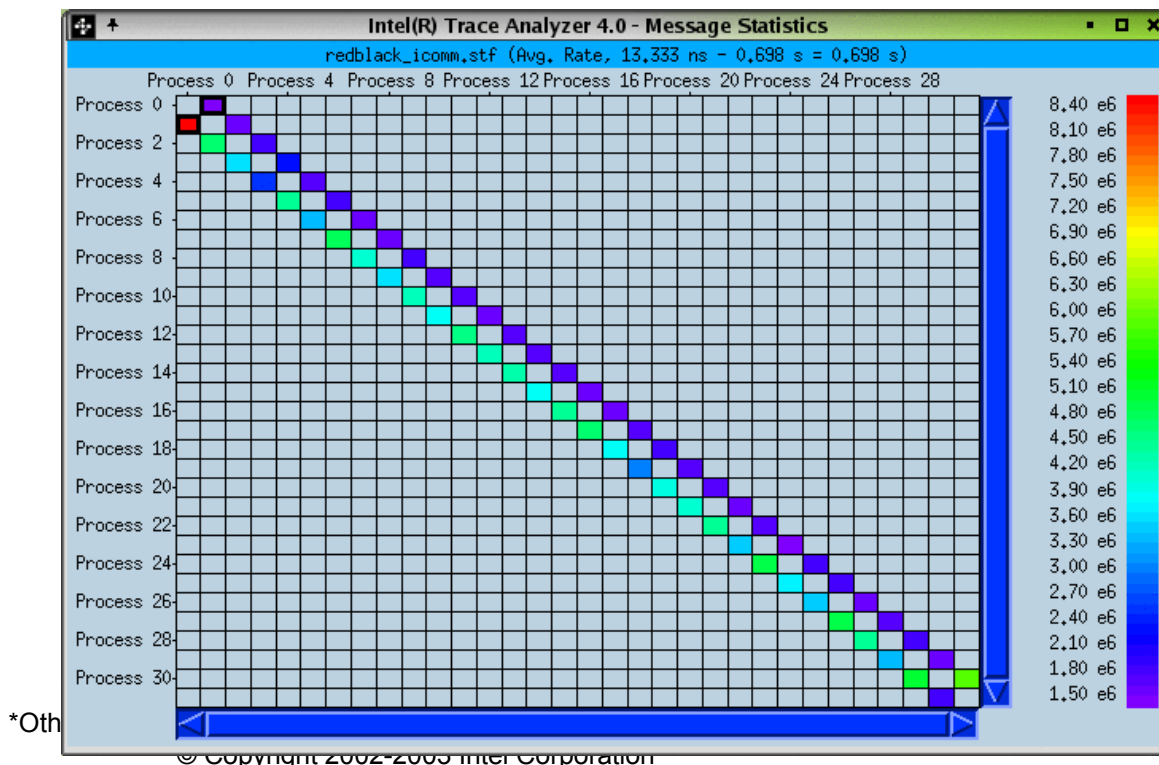
Intel Trace Analyzer – Load Balance Analysis

- Zoom to identify
- min/max load



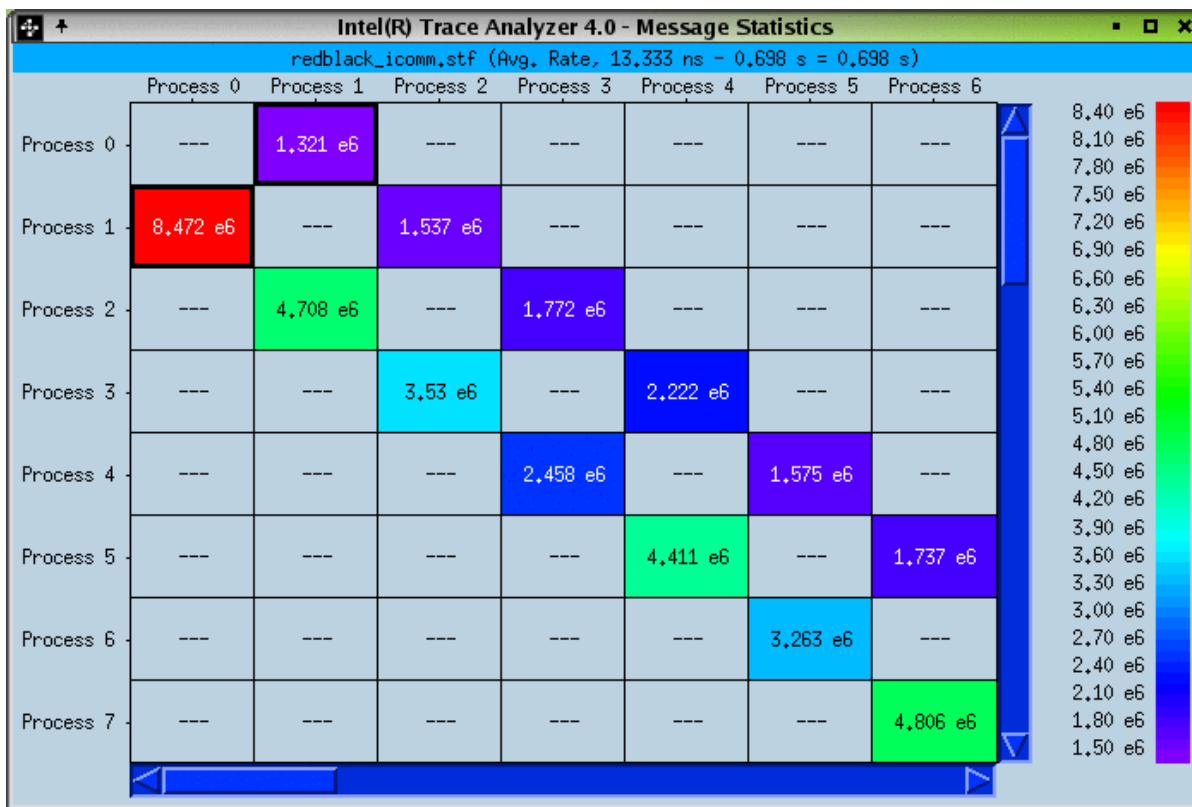
Intel Trace Analyzer – Message Statistics

- Setup Message Statistics
 - Global Displays ⇒ Message Statistics
 - Context Menu ⇒ Display ⇒ Avg. Rate



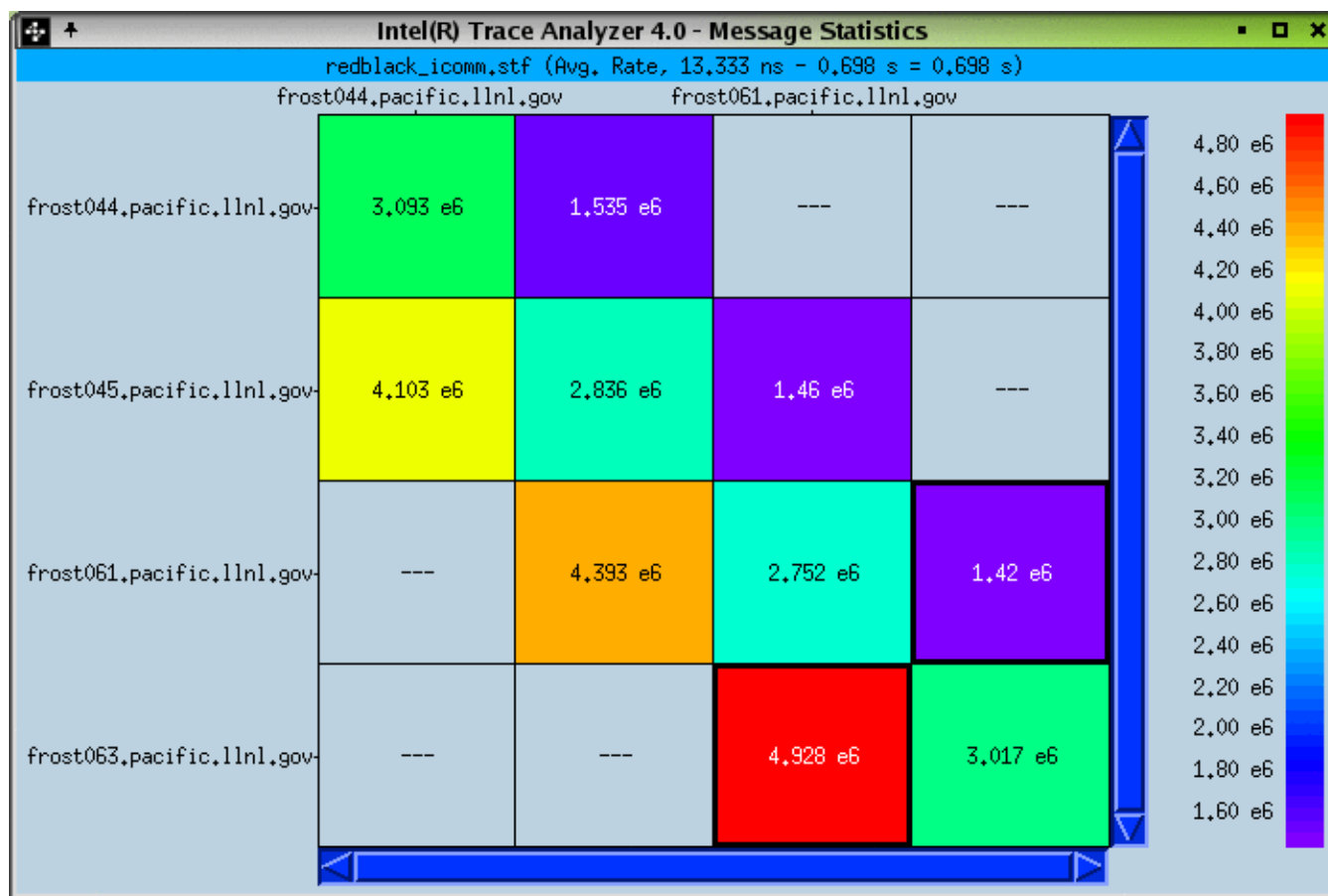
Intel Trace Analyzer – Message Statistics

- Zoom to read min/max values



Intel Trace Analyzer – Message Statistics

- Select All Nodes



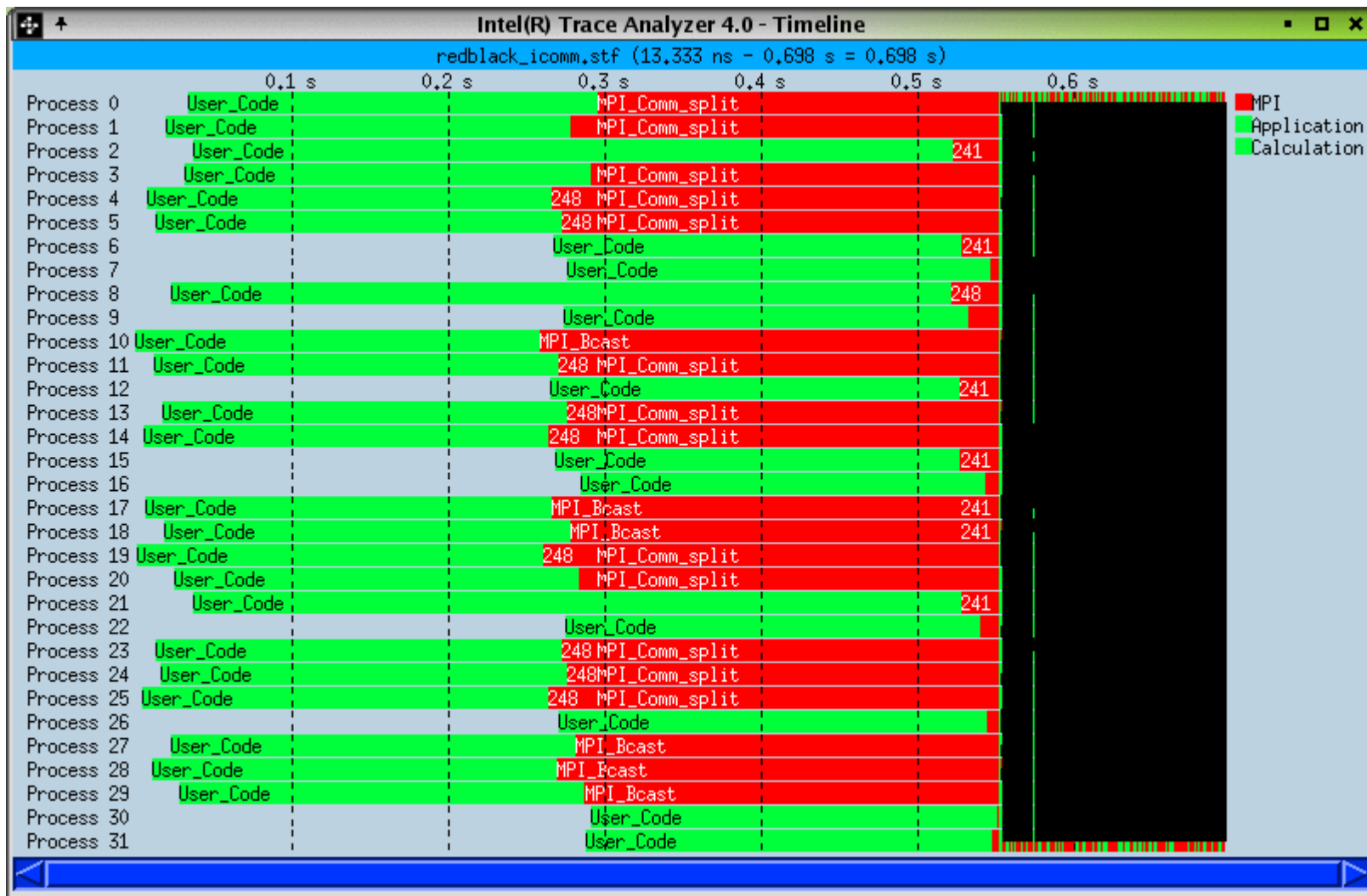
*Other brands and names are the property of their respective owners

© Copyright 2002-2003 Intel Corporation

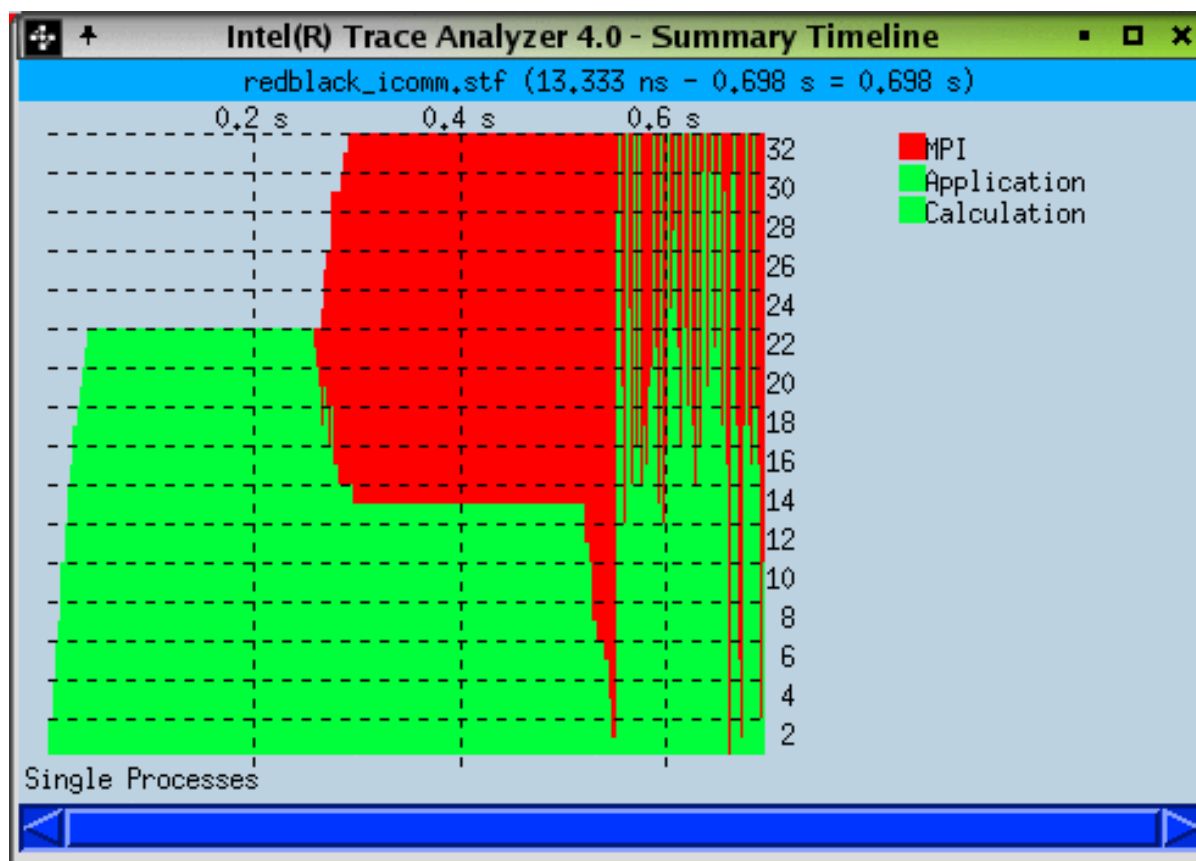
Intel Trace Analyzer – Timeline Displays

- Open two Timelines by
 - Global Displays⇒Timeline
 - Global Displays⇒Summary Timeline
 - Zoom

Intel Trace Analyzer – Activity Timeline

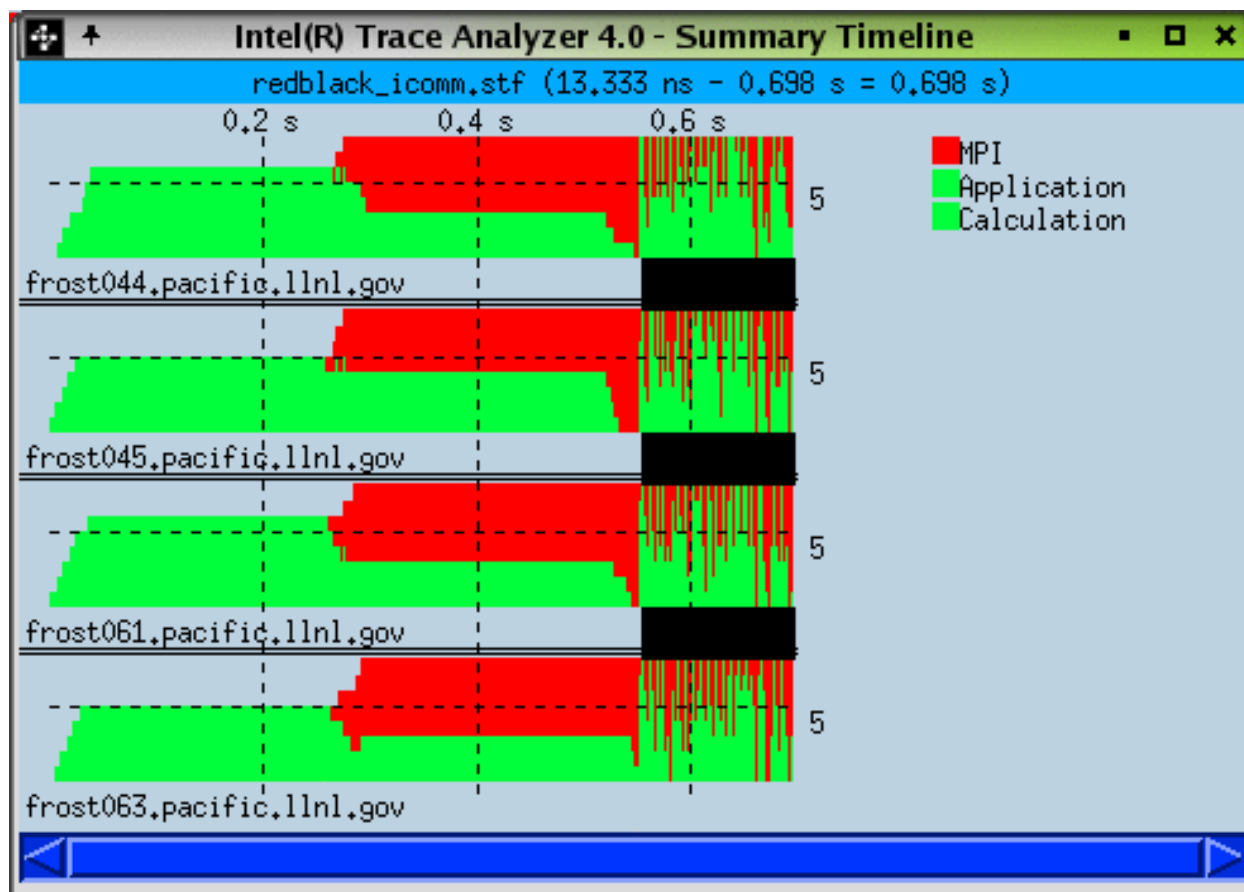


Intel Trace Analyzer – Summary Timeline



Intel Trace Analyzer – Summary Timeline

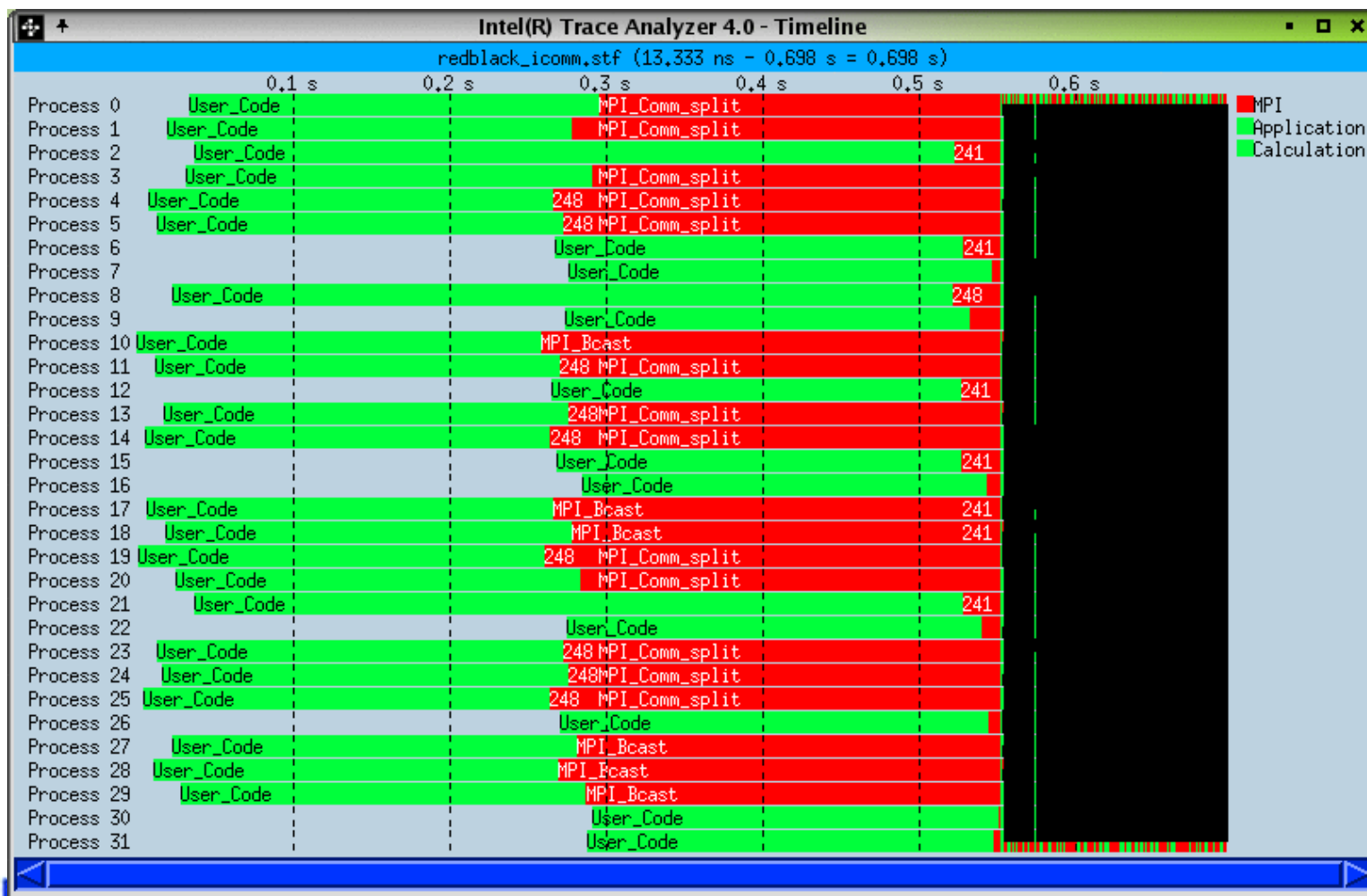
- Select All Nodes



*Other brands and names are the property of their respective owners

© Copyright 2002-2003 Intel Corporation

Intel Trace Analyzer – Zooming and Linked Displays



*Other brands and names are the property of their respective owners

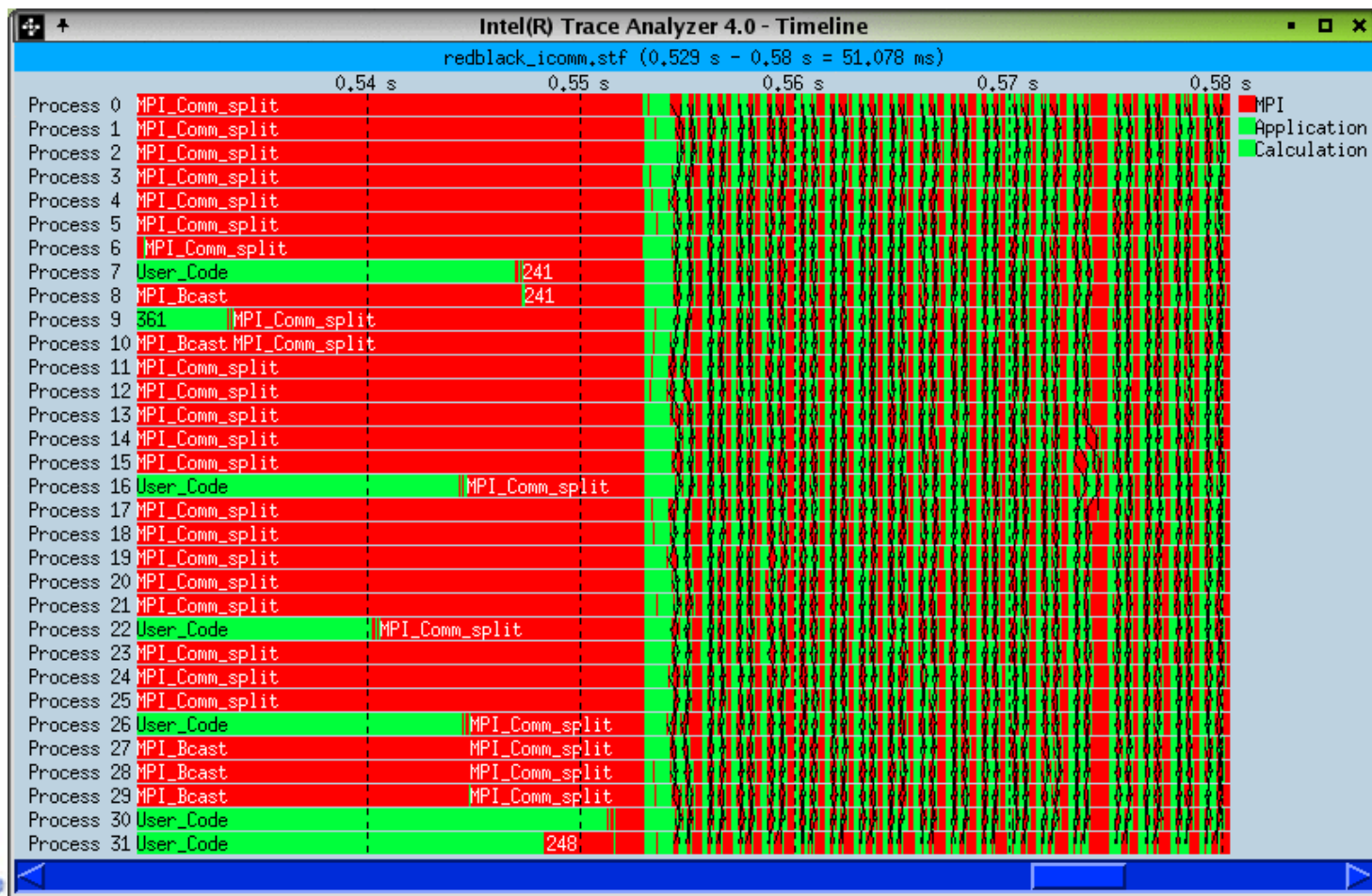
© Copyright 2002-2003 Intel Corporation

Intel Trace Analyzer – Zooming and Linked Displays

- To zoom on a display, press the left mouse button at the start of the region to be magnified
- While holding the mouse button, drag the mouse to the end of the desired magnification region, which will be shown by Intel Trace Analyzer with a rectangular rubber-band, and then release the left mouse button
- Zooming can be done in an unlimited depth
- The reverse action, Undo Zoom, works hierarchical and can be invoked with the hotkey U (case-insensitive) or from the context menu

Intel Trace Analyzer – Zooming and Linked Displays

- The result of zooming from the previous panel display



Product Key Features and Benefits

Event-based tracing tool

Accurate and detailed information of serial and parallel program runs

Low overhead structured trace file format (STF)

Designed from the ground up for scalability and compact data representation. It can be written in parallel and allows random access to parts of a trace.

Provides a convenient way to graphically analyze runtime event traces produced by MPI and other applications Enables the user to quickly focus at the appropriate level of detail.



Product Key Features and Benefits

Variety of graphical displays

Presents important aspects of the application runtime behavior in detailed and aggregate views.

Timeline views and parallelism display

The timeline display visualizes the concurrent behavior of parallel applications and statistics can be calculated on demand for certain time intervals and specific processes.

Communication statistics

Communication metrics for an arbitrary time interval and the message-length distribution.



Product Key Features and Benefits

Execution statistics

Display subroutine execution metrics for an arbitrary time interval and shows call-tree information.

Easy to use API

Provide the user with a flexible and easy to use interface to instrument the user code.

Thread-safe

Allows tracing of multithreaded applications.

Tutorial – Task0: setup environment

- objective: getting started
- places and paths for compile, link, execute, license keys, doc, etc.

Tutorial – Task1: start Intel Trace Analyzer

- objective: basic use of Intel Trace Analyzer
- this directory contains
`jacobic.stf.single.gz`
which is an example trace file.

Tutorial – Task2: basic tracefile generation

objective: basic use of Intel Trace Collector,
how to
generate a tracefile.

this directory contains

```
mpi.c
```

which is a simple example program from the
mpich distribution.

Tutorial – Task3: basic API use to instrument function

objective: basic use of Intel Trace Collector API,
how to instrument a function

this directory contains

```
api.c
```

which is a simple example program from the
mpich distribution.

Tutorial – Task4: instrument function with source code location

objective: basic use of Intel Trace Collector API,
how to instrument a function with source code
location

this directory contains

```
api.c
```

which is a simple example program from the
mpich distribution.

Tutorial – Task5: record custom performance counter

objective: basic use of Intel Trace Collector API,
how to instrument a custom performance counter

this directory contains

```
cpic.c
```

which is a simple example program from the mpich distribution.

Tutorial – Task6: selective tracing and filtering

objective: basic use of Intel Trace Collector API, switch recording off and on, basic use of a VT_CONFIG configuration file

this directory contains

```
cpic.c
```

which is a simple example program from the mpich distribution.

Tutorial – Task7: convert data to single trace file

objective: learn about stftool, use single file trace archives to simplify transfer of storage