Jumpshot , Vampir and TAU

Rosa Filgueira
Data Intensive Research, School of Informatics, University of Edinburgh

rosa.filgueira@ed.ac.uk

Jumpshot: Basic Information

- Name: MPE/Jumpshot
- Developer: Argonne National Labratory
- Current versions:
  - MPE 1.26
  - Jumpshot-4
- Website:
  http://www-unix.mcs.anl.gov/perfvis

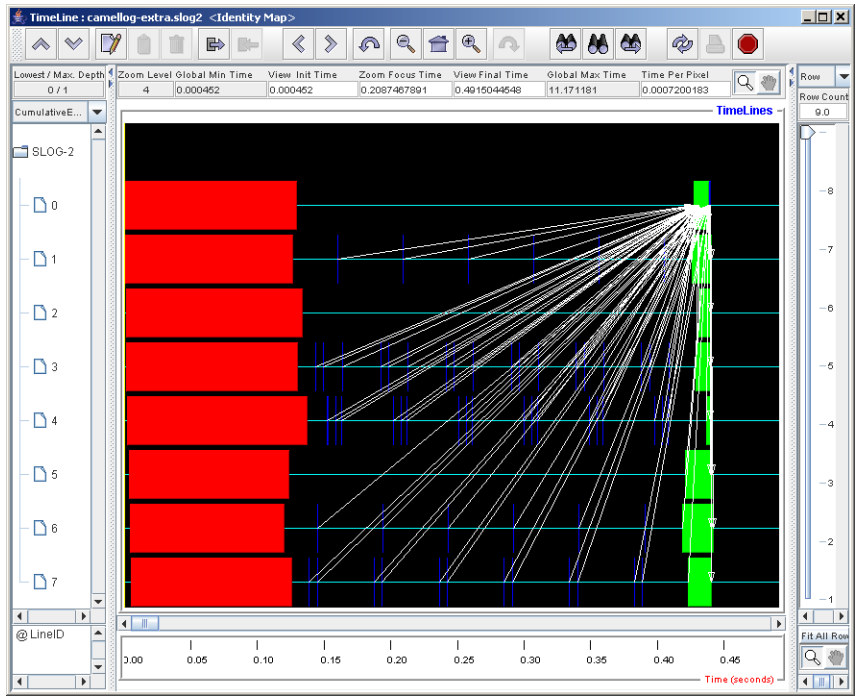
What Is MPE/Jumpshot?

- **MPE** stands for "Multi-Processing Environment". It is a suite of performance analysis tools for MPI programs.
- Creates a log file(s) while the MPI program executes, then provides tools to analyze and view the log files after the execution.
  - clog2 format → we need to convert to slog2 format
- *Jumpshot* is a Java-based GUI tool for viewing one of these log files:
  - A visualization tool for logfiles (slog2) created by the MPE package
  - Written in Java (crossplatform)
  - Provides a “time line” (GANTT) view of MPI and program events
  - Also has basic search and summary (histogram) functionality

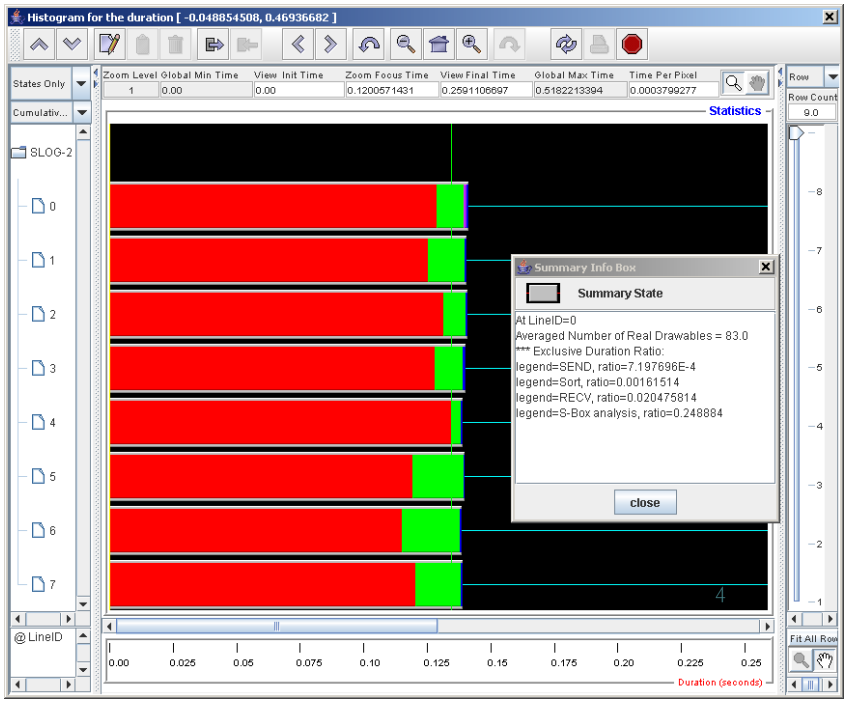
Jumpshot Overview

- Jumpshot-4 supports two types of visualizations for metrics
  - Timeline (left, botto,)
  - Histogram (right, bottom)
- Visualization is dependant on SLOG-2 format and Data model

Timeline view



Histogram view



Hello Word

```
#include "stdio.h"
#include <stdlib.h>
#include <mpi.h>
#include <string.h>
#include <unistd.h>
int main(int argc, char *argv[])
{
int rank, size, i;
int type = 0;
MPI_Status status;
char message[20];
char hostname[1024];
MPI_Init(&argc,&argv);
MPI_Comm_size(MPI_COMM_WORLD,&size);
MPI_Comm_rank(MPI_COMM_WORLD,&rank);
hostname[1023] = '\0';
gethostname(hostname, 1023);
printf("My Hostname proc %d : %s\n",rank,hostname);
if (rank == 0) {
strcpy(message,"Hello, world");
for (i = 1; i < size; i++) MPI_Send(message,
13,MPI_CHAR,i,type,MPI_COMM_WORLD);
}
else
MPI_Recv(message,20,MPI_CHAR,0,type,MPI_COMM_WORLD,&status);
printf("Message from node %d: %13s\n", rank,message);
MPI_Finalize();
}
```

How to use MPE + Jumpshot

- Compile a MPICH program with MPE Logging Library:
  - [rosa@moore mpi_exercise]$ mpicc -g –c hello.c
  - [rosa@moore mpi_exercise]$ mpicc -o hello hello.o -llmpe -lmpe –lm
- Run the application:
  - qsub run.sh
    - It should generate a log file hello.clog2
- Convert log format from CLOG2 to SLOG2:
  - [rosa@moore mpi_exercise]$ /root/rosa_filgueira/slog2sdk-1.2.6/bin/clog2TOslog2 hello.clog2

Convert to SLOG2 format

[rosa@moore mpi_exercise]$ /root/rosa_filgueira/slog2sdk-1.2.6/bin/clog2TOslog2 hello.clog2

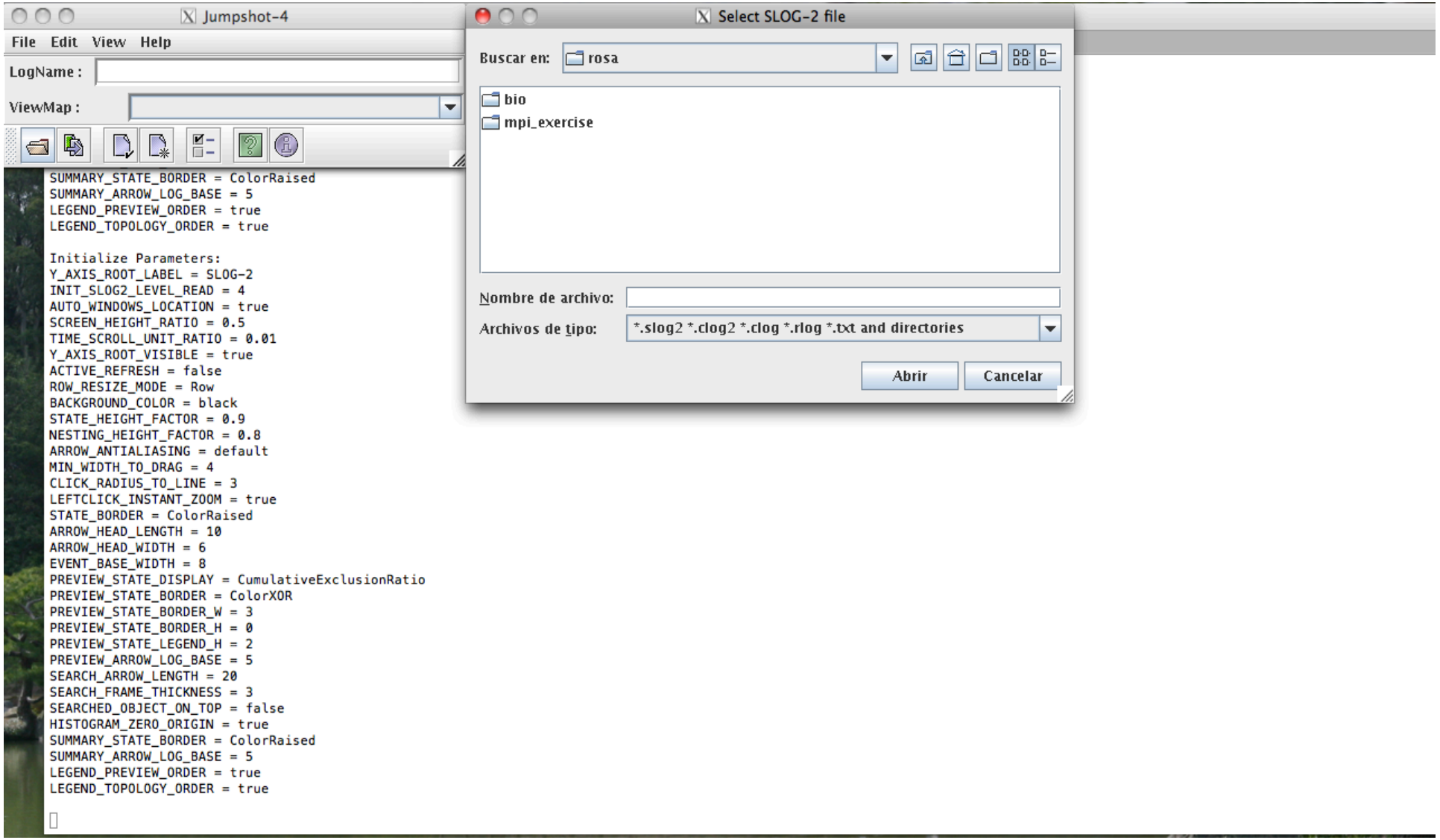
```
GUI_LIBDIR is set. GUI_LIBDIR = /root/rosa_filgueira/slog2sdk-1.2.6/lib
        SLOG-2 Header:
version = SLOG 2.0.6
NumOfChildrenPerNode = 2
TreeLeafByteSize = 65536
MaxTreeDepth = 0
MaxBufferByteSize = 715
Categories  is FBinfo(478 @ 823)
MethodDefs  is FBinfo(0 @ 0)
LineIDMaps  is FBinfo(232 @ 1301)
TreeRoot    is FBinfo(715 @ 108)
TreeDir     is FBinfo(38 @ 1533)
Annotations is FBinfo(0 @ 0)
Postamble   is FBinfo(0 @ 0)

Number of Drawables = 21
Number of Unmatched Events = 0
Total ByteSize of the logfile = 3040
timeElapsed between 1 & 2 = 13 msec
timeElapsed between 2 & 3 = 48 msec
[rosa@moore mpi_exercise]$
```

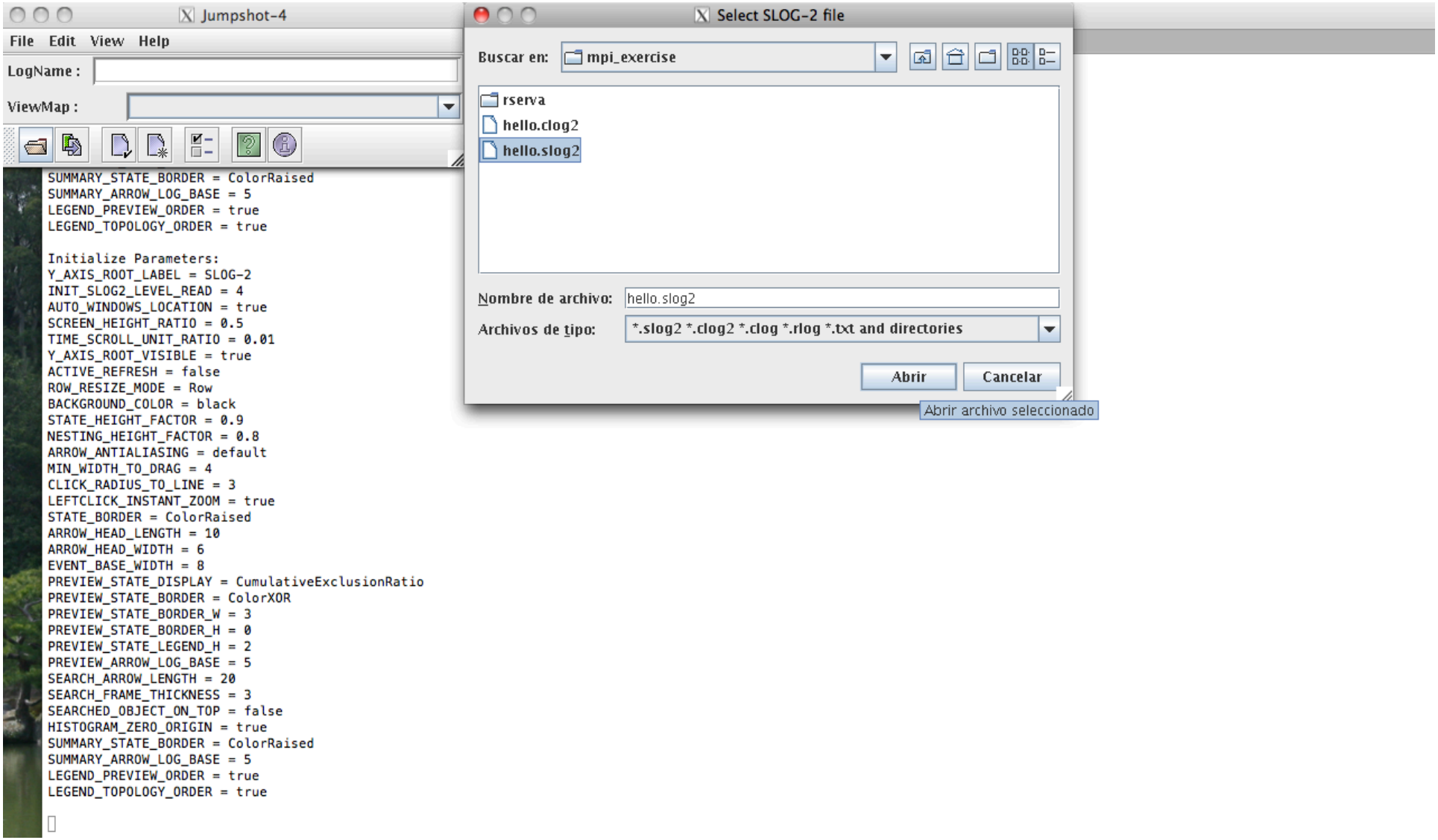
MPE + Jumpshot

- Start Jumpshot-4:
  - New terminal: ssh -X user@moore.udl.net
  - **/share/apps/jumpshot/bin/jumpshot**
    - First time that the user starts jumpshot, the application generates a configuration file and asks to save it. → Yes.

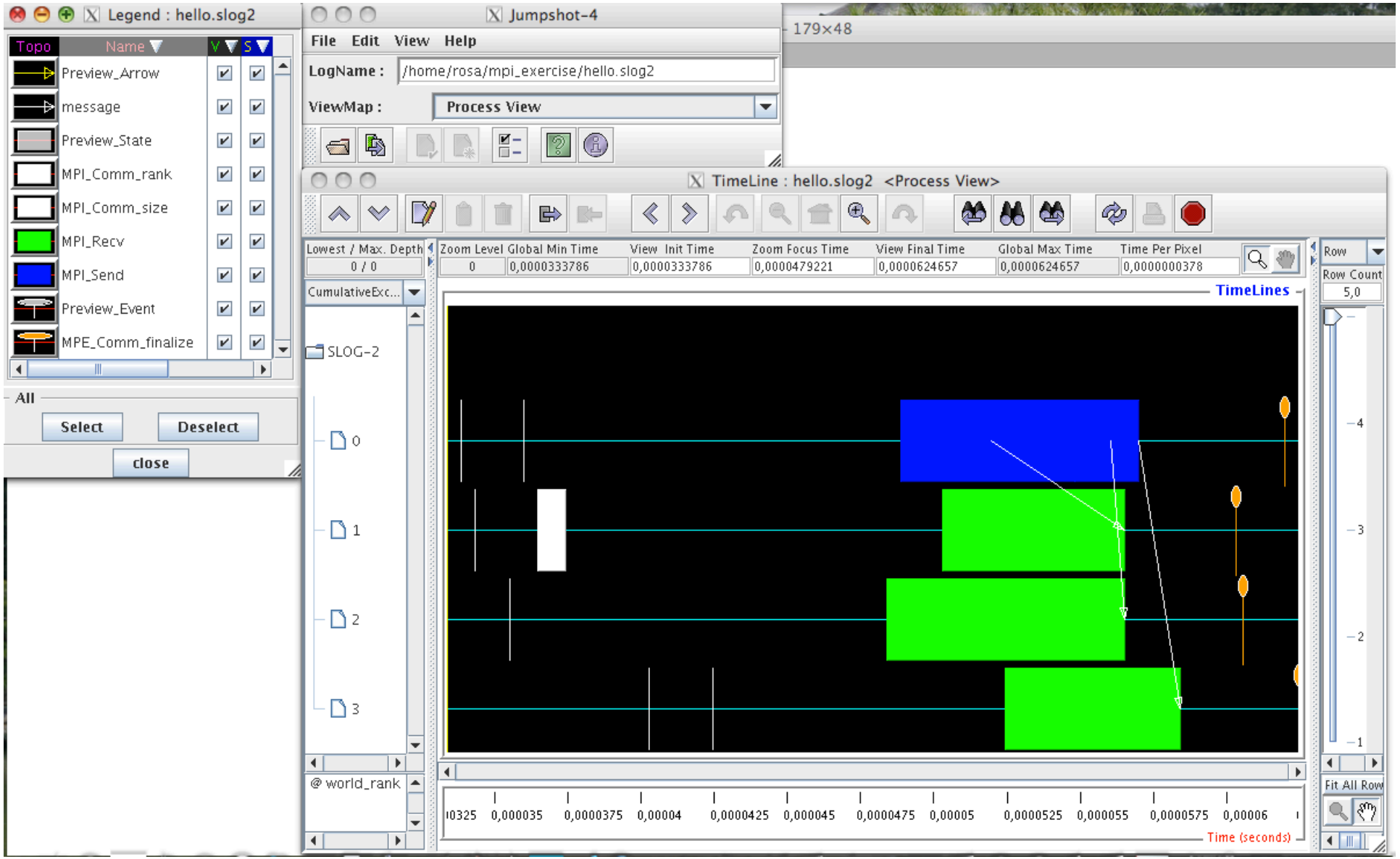
JumpShot-4


Jumpshot-4
File Edit View Help
LogName :
ViewMap :
SUMMARY_STATE_BORDER = ColorRaised
SUMMARY_ARROW_LOG_BASE = 5
LEGEND_PREVIEW_ORDER = true
LEGEND_TOPOLOGY_ORDER = true
Initialize Parameters:
Y_AXIS_ROOT_LABEL = SLOG-2
INIT_SLOG2_LEVEL_READ = 4
AUTO_WINDOWS_LOCATION = true
SCREEN_HEIGHT_RATIO = 0.5
TIME_SCROLL_UNIT_RATIO = 0.01
Y_AXIS_ROOT_VISIBLE = true
ACTIVE_REFRESH = false
ROW_RESIZE_MODE = Row
BACKGROUND_COLOR = black
STATE_HEIGHT_FACTOR = 0.9
NESTING_HEIGHT_FACTOR = 0.8
ARROW_ANTIALIASING = default
MIN_WIDTH_TO_DRAG = 4
CLICK_RADIUS_TO_LINE = 3
LEFTCLICK_INSTANT_ZOOM = true
STATE_BORDER = ColorRaised
ARROW_HEAD_LENGTH = 10
ARROW_HEAD_WIDTH = 6
EVENT_BASE_WIDTH = 8
PREVIEW_STATE_DISPLAY = CumulativeExclusionRatio
PREVIEW_STATE_BORDER = ColorXOR
PREVIEW_STATE_BORDER_W = 3
PREVIEW_STATE_BORDER_H = 0
PREVIEW_STATE_LEGEND_H = 2
PREVIEW_ARROW_LOG_BASE = 5
SEARCH_ARROW_LENGTH = 20
SEARCH_FRAME_THICKNESS = 3
SEARCHED_OBJECT_ON_TOP = false
HISTOGRAM_ZERO_ORIGIN = true
SUMMARY_STATE_BORDER = ColorRaised
SUMMARY_ARROW_LOG_BASE = 5
LEGEND_PREVIEW_ORDER = true
LEGEND_TOPOLOGY_ORDER = true
Select SLOG-2 file
Buscar en:
rosa
bio
mpi_exercise
Nombre de archivo:
Archivos de tipo:
*.slog2 *.clog2 *.clog *.rlog *.txt and directories
Abrir
Cancelar

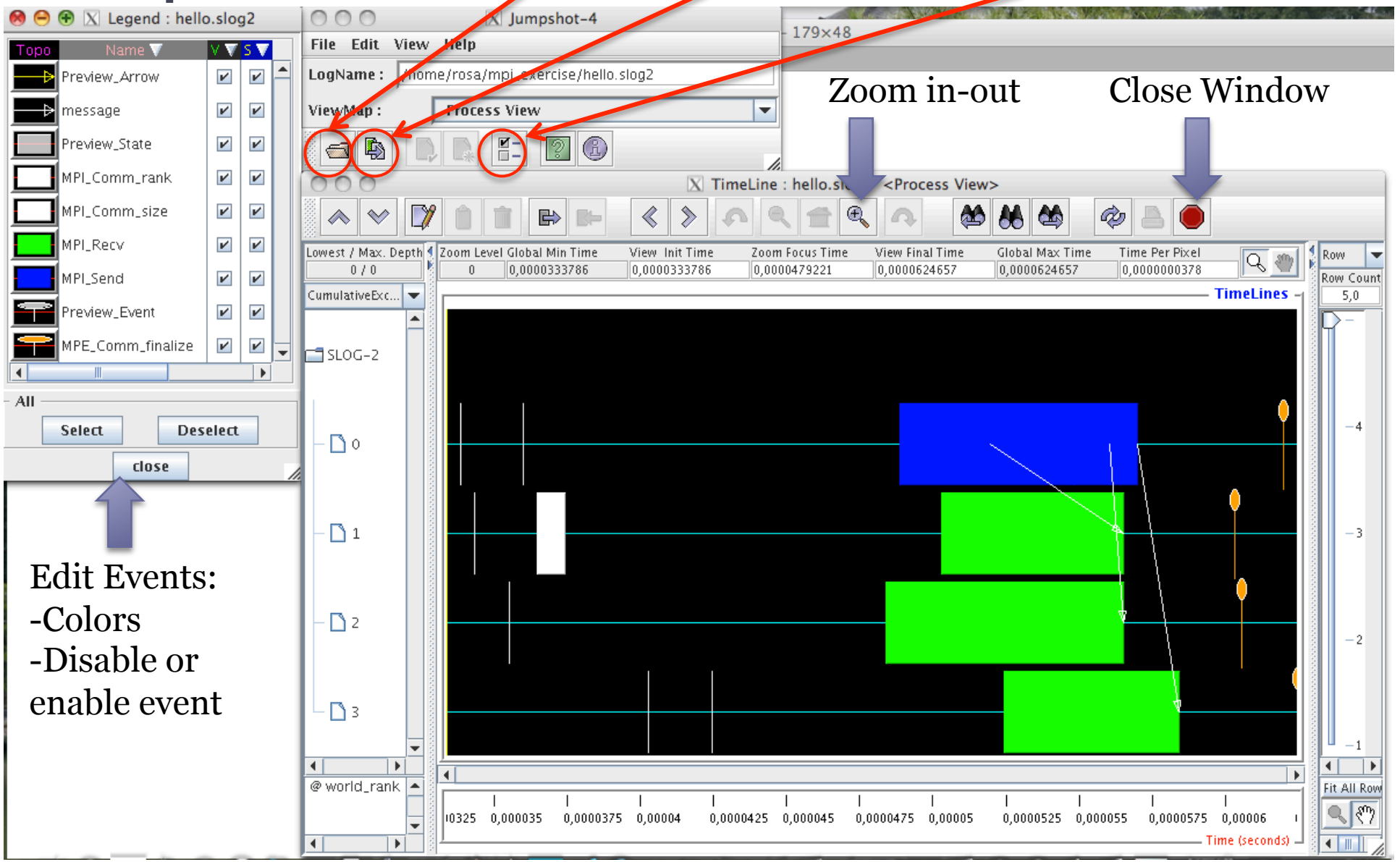
JumpShot-4


Jumpshot-4
File Edit View Help
LogName :
ViewMap :
SUMMARY_STATE_BORDER = ColorRaised
SUMMARY_ARROW_LOG_BASE = 5
LEGEND_PREVIEW_ORDER = true
LEGEND_TOPOLOGY_ORDER = true
Initialize Parameters:
Y_AXIS_ROOT_LABEL = SLOG-2
INIT_SLOG2_LEVEL_READ = 4
AUTO_WINDOWS_LOCATION = true
SCREEN_HEIGHT_RATIO = 0.5
TIME_SCROLL_UNIT_RATIO = 0.01
Y_AXIS_ROOT_VISIBLE = true
ACTIVE_REFRESH = false
ROW_RESIZE_MODE = Row
BACKGROUND_COLOR = black
STATE_HEIGHT_FACTOR = 0.9
NESTING_HEIGHT_FACTOR = 0.8
ARROW_ANTIALIASING = default
MIN_WIDTH_TO_DRAG = 4
CLICK_RADIUS_TO_LINE = 3
LEFTCLICK_INSTANT_ZOOM = true
STATE_BORDER = ColorRaised
ARROW_HEAD_LENGTH = 10
ARROW_HEAD_WIDTH = 6
EVENT_BASE_WIDTH = 8
PREVIEW_STATE_DISPLAY = CumulativeExclusionRatio
PREVIEW_STATE_BORDER = ColorXOR
PREVIEW_STATE_BORDER_W = 3
PREVIEW_STATE_BORDER_H = 0
PREVIEW_STATE_LEGEND_H = 2
PREVIEW_ARROW_LOG_BASE = 5
SEARCH_ARROW_LENGTH = 20
SEARCH_FRAME_THICKNESS = 3
SEARCHED_OBJECT_ON_TOP = false
HISTOGRAM_ZERO_ORIGIN = true
SUMMARY_STATE_BORDER = ColorRaised
SUMMARY_ARROW_LOG_BASE = 5
LEGEND_PREVIEW_ORDER = true
LEGEND_TOPOLOGY_ORDER = true
Select SLOG-2 file
Buscar en:
mpi_exercise
rserva
hello.clog2
hello.slog2
Nombre de archivo:
hello.slog2
Archivos de tipo:
*.slog2 *.clog2 *.clog *.rlog *.txt and directories
Abrir
Cancelar
Abrir archivo seleccionado

JumpShot-4


Legend : hello.slog2
Topo
Name
V
S
Preview_Arrow
message
Preview_State
MPI_Comm_rank
MPI_Comm_size
MPI_Recv
MPI_Send
Preview_Event
MPE_Comm_finalize
All
Select
Deselect
close
Jumpshot-4
File Edit View Help
LogName : /home/rosa/mpi_exercise/hello.slog2
ViewMap : Process View
179×48
TimeLine : hello.slog2 <Process View>
Lowest / Max. Depth
0 / 0
Zoom Level
Global Min Time
View Init Time
Zoom Focus Time
View Final Time
Global Max Time
Time Per Pixel
0
0,0000333786
0,0000333786
0,0000479221
0,0000624657
0,0000624657
0,0000000378
CumulativeExc...
SLOG-2
0
1
2
3
@ world_rank
TimeLines
Row
Row Count
5,0
Fit All Rows
0,000035
0,0000375
0,00004
0,0000425
0,000045
0,0000475
0,00005
0,0000525
0,000055
0,0000575
0,00006
Time (seconds)

JumpShot-4

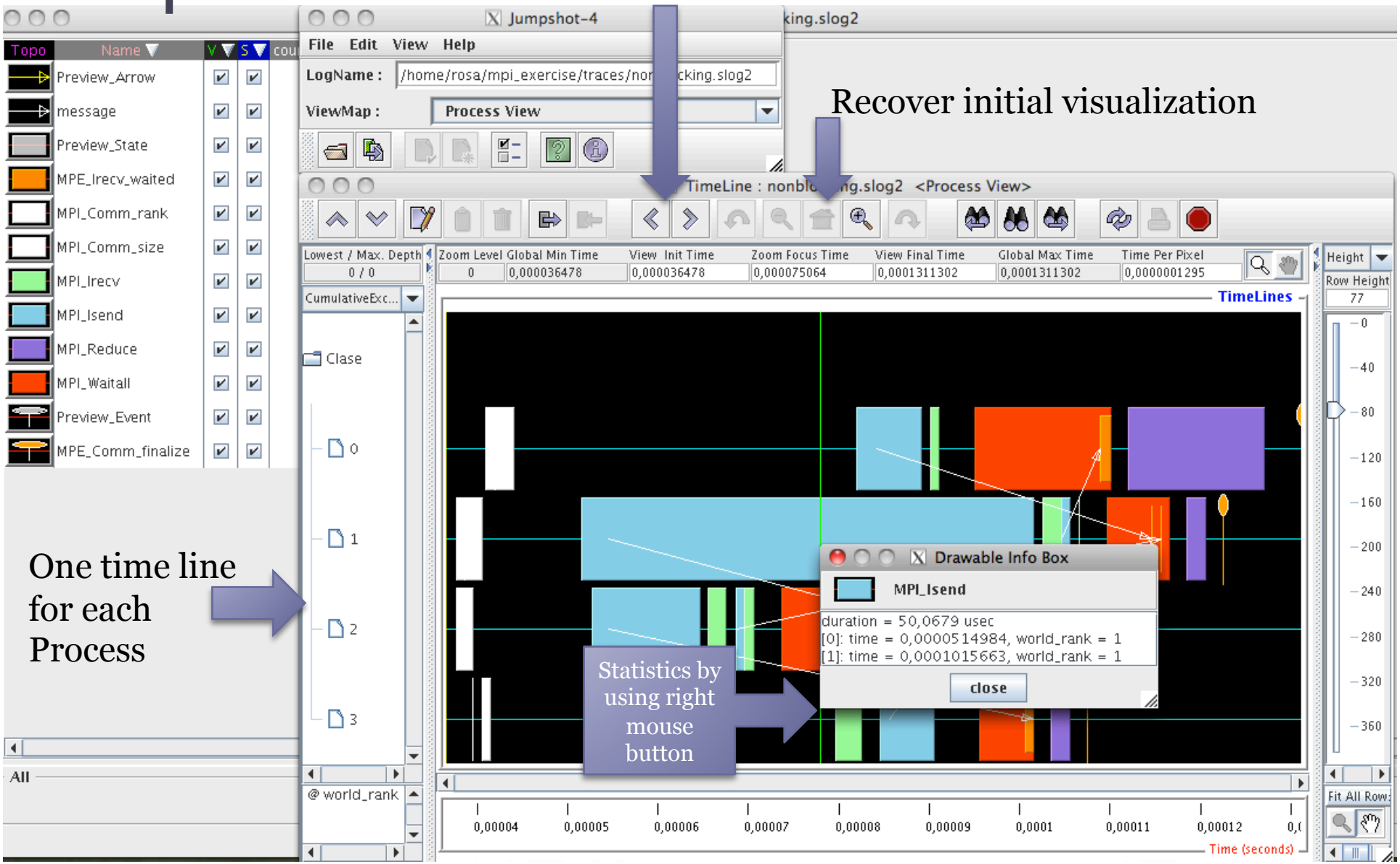


Edit Events:
-Colors
-Disable or enable event

JumpShot-4

Move through the time line

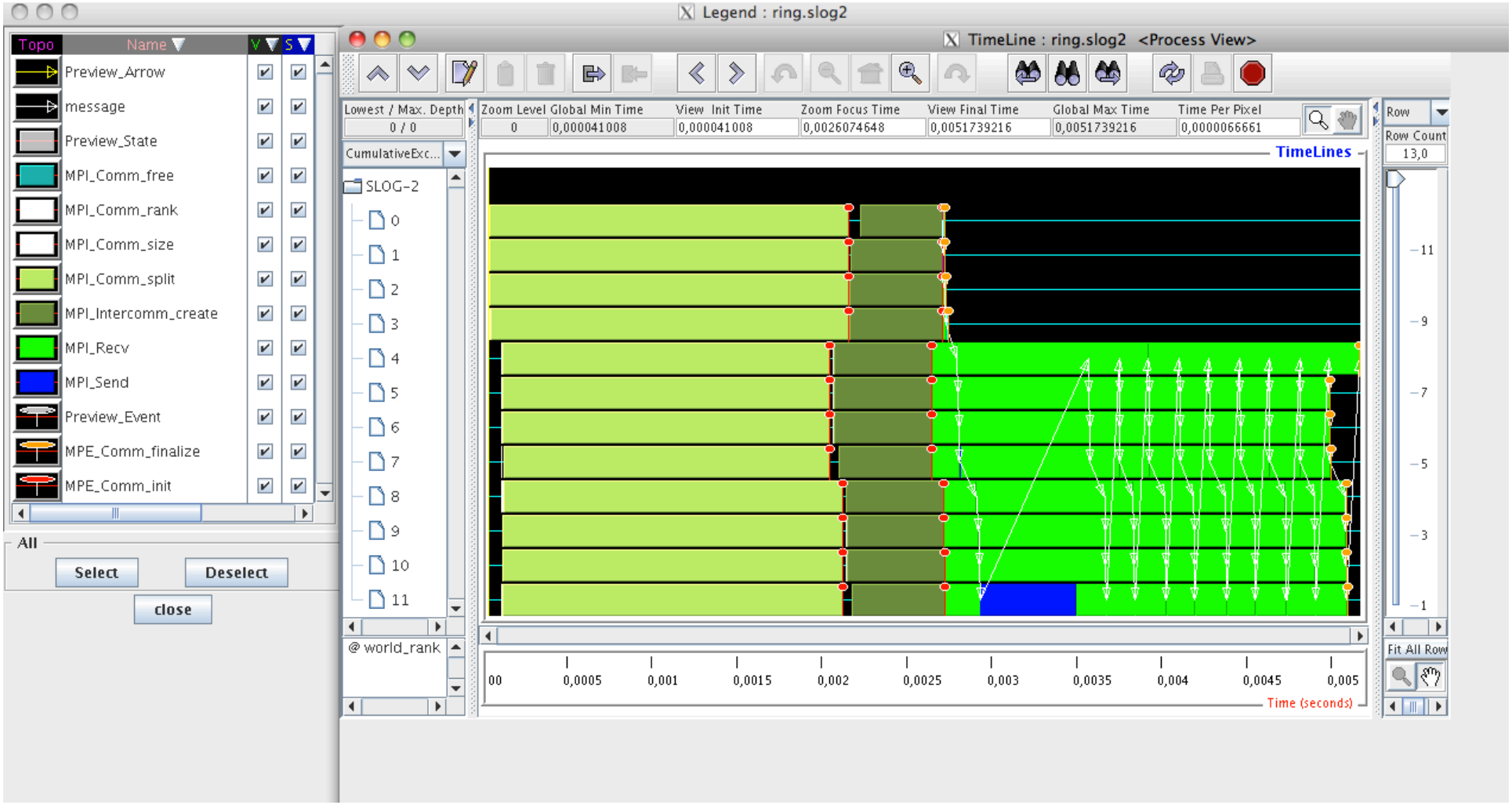
Recover initial visualization



One time line for each Process

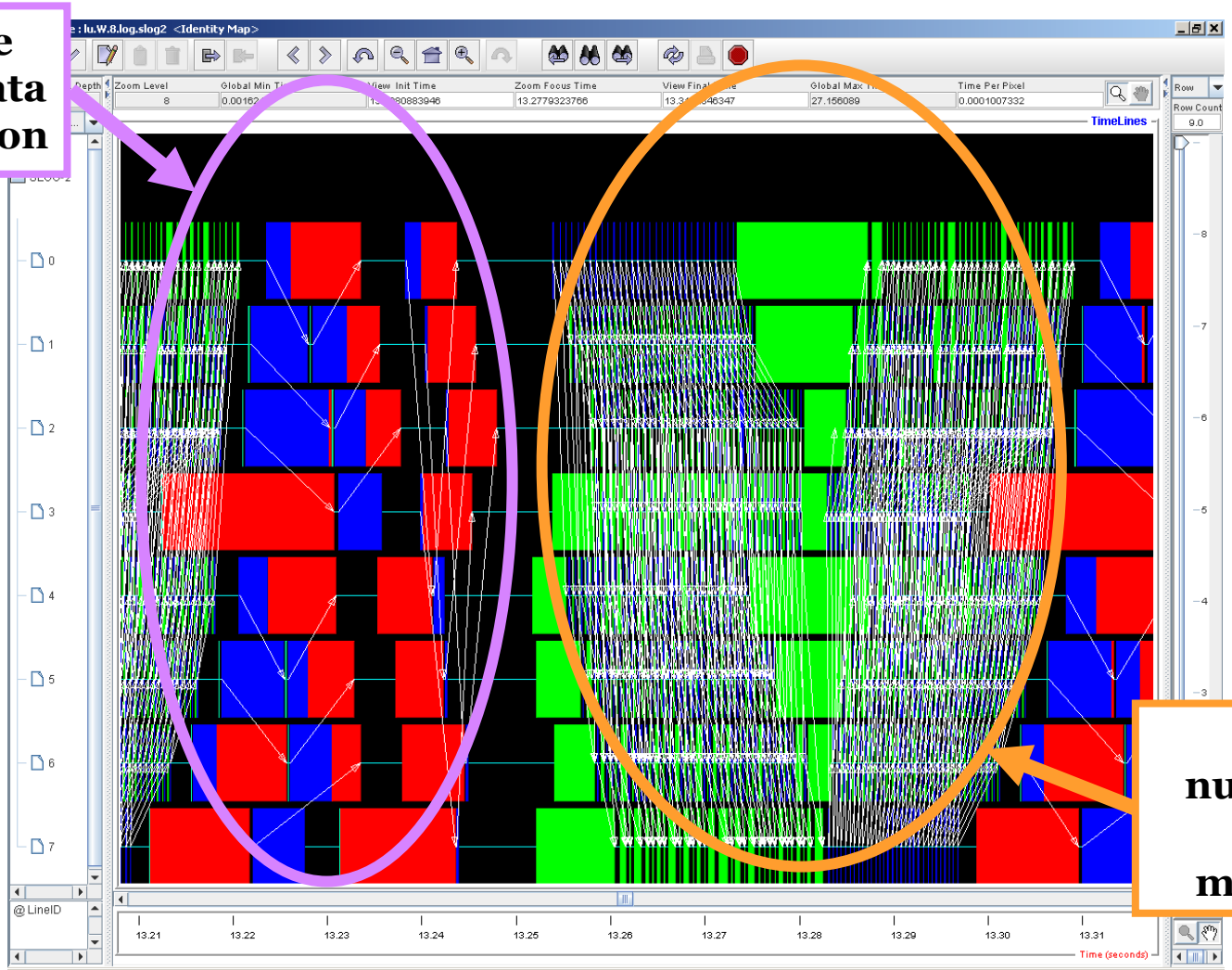
Statistics by using right mouse button

Ring.c


Legend : ring.slog2
Topo
Name
V
S
Preview_Arrow
message
Preview_State
MPI_Comm_free
MPI_Comm_rank
MPI_Comm_size
MPI_Comm_split
MPI_Intercomm_create
MPI_Recv
MPI_Send
Preview_Event
MPE_Comm_finalize
MPE_Comm_init
All
Select
Deselect
close
TimeLine : ring.slog2 <Process View>
Lowest / Max. Depth
0 / 0
Zoom Level
Global Min Time
View Init Time
Zoom Focus Time
View Final Time
Global Max Time
Time Per Pixel
0
0,000041008
0,000041008
0,0026074648
0,0051739216
0,0051739216
0,0000066661
CumulativeExc...
SLOG-2
0
1
2
3
4
5
6
7
8
9
10
11
@ world_rank
TimeLines
Row
Row Count
13,0
11
9
7
5
3
1
Fit All Rows
00
0,0005
0,001
0,0015
0,002
0,0025
0,003
0,0035
0,004
0,0045
0,005
Time (seconds)

JumpShot-4

Much time taken for data redistribution



Large number of small messages

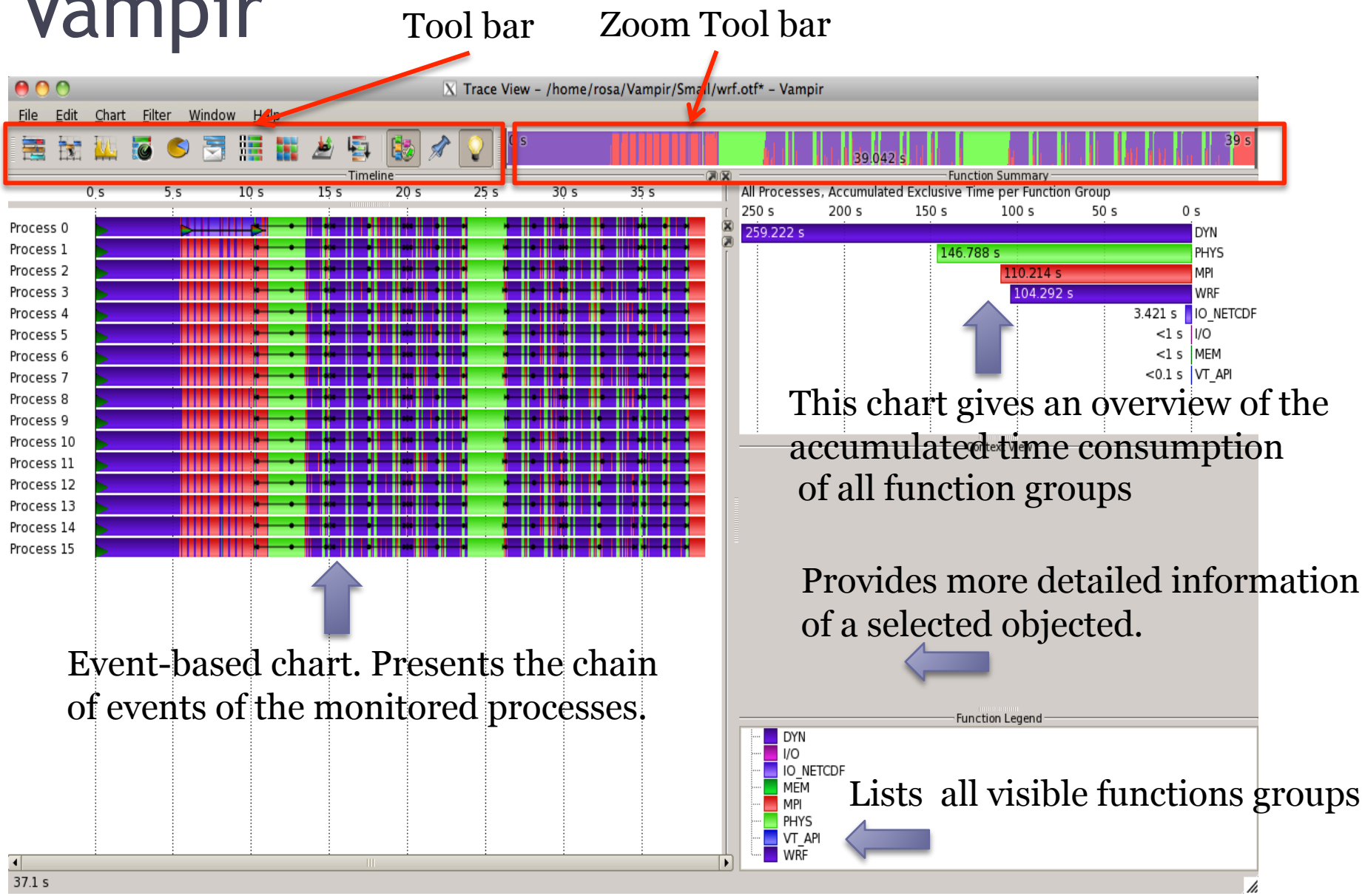
Vampir

- Performance Analysis Framework
  - Easy to use performance analysis framework for parallel programs
  - Graphical data representation enables detailed understanding of dynamic processes on massively parallel systems
  - In-depth event based analysis of parallel run-time behavior and interprocess communication
  - Identification of performance problems and bottlenecks
  - Linux-based PCs and Clusters, SGI, IBM, SUN, NEC, HP, Apple, ..

Vampir

- **Vampir Features**
  - Powerful zooming and scrolling in all displays
  - Adaptive statistics for user selected time ranges
  - Filtering of processes, functions, messages, collective operations
  - Hierarchical grouping of threads, processes, and nodes
  - Support of source code locations
  - Integrated snapshot and printing for publishing
  - Customizable displays

Vampir



This chart gives an overview of the accumulated time consumption of all function groups

Provides more detailed information of a selected objected.

Event-based chart. Presents the chain of events of the monitored processes.

Lists all visible functions groups

TAU

- TAU Performance Systems is a portable profiling and tracing toolkit for performance analysis of parallel programs written in Fortran, C, C++, Java, and Python.
- TAU (Tuning and Analysis Utilities) is capable of gathering performance information through instrumentation of functions, methods, basic blocks, and statements.
- TAU's profile visualization tool, paraprof, provides graphical displays of the performance analysis results, to help the user visualize the collected data.

TAU instrumentation

- **Dynamic Instrumentation:** No change to your application's source or binary code is necessary. Using the tau_exec command before your executable generates both profile and trace data.
- **Compiler-Based Instrumentation:** TAU can also be used to compile your application using wrapper scripts for mpicc, mpicxx, and mpif90. Tau instruments the binary code without changing the source. Using the tau_exec command before the TAU-compiled executable generates profile and trace data. Data is collected on both MPI calls and your application's function calls.

Visualize Information

- Jumpshot: To visualize traces
- Paraprof: To visualize profiles
  - Tool provided by TAU
  - GUI (Java)
  - Text and graphics
  - Options
    - --pack <file>
    - Pack the data in a unique file (no GUI)
      - paraprof -pack app.ppk

Dynamic Instrumentation

- Compile
  - mpicc –o hello hello.c
- Dynamic instrumentation
  - Modify the “run.sh” script to insert
    - export TAU_TRACE=1 # This line is needed only for tracing.
    - export TAU_PROFILE=1 # This line is needed only for profiling.
    - mpiexec -f $MPICH_MACHINES -n $NSLOTS **tau_exec -T PROFILE** ./hello

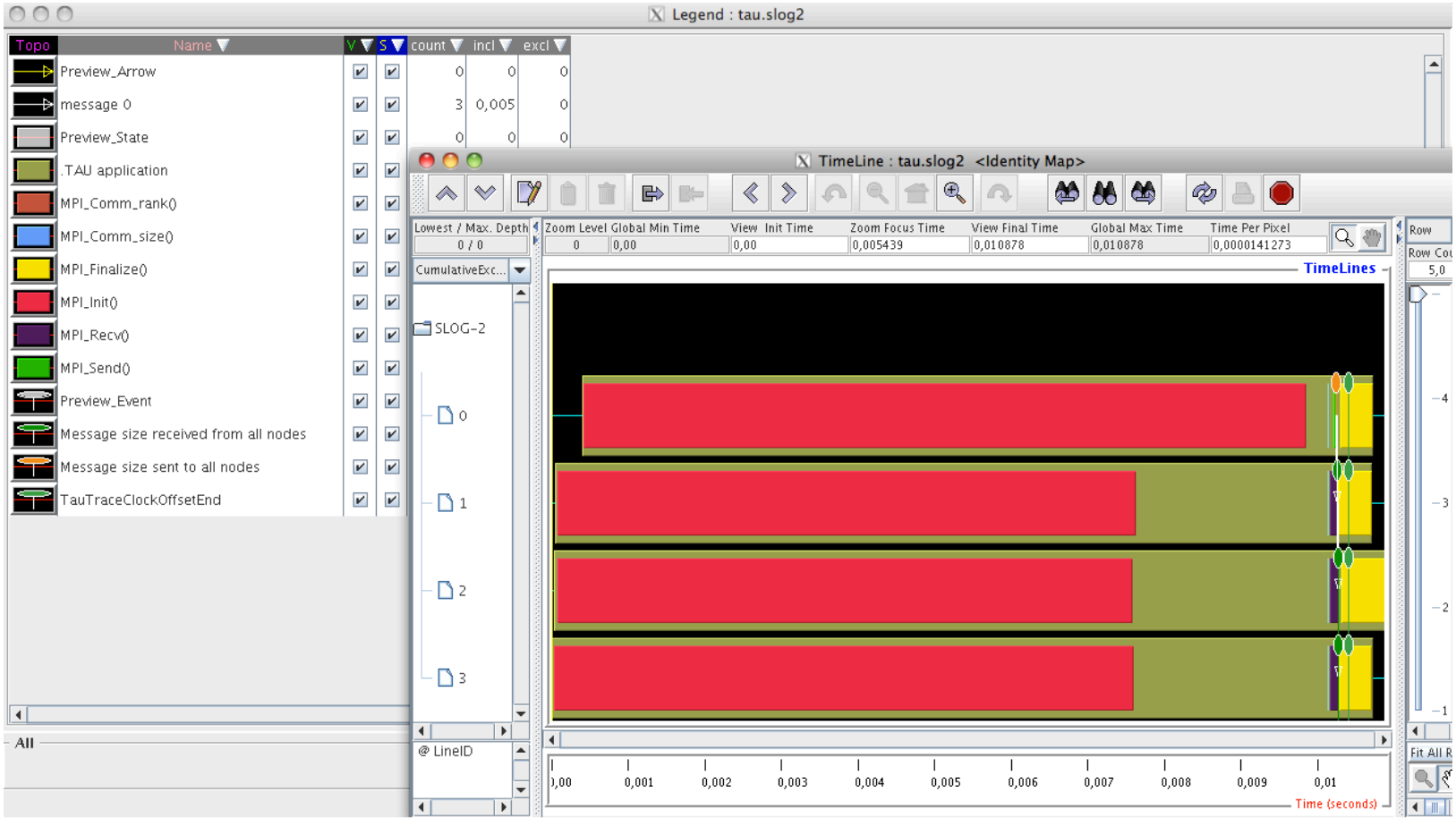
Generating traces and profiles

- Run the application
  - qsub run.sh
  - The script will generate two types of data:
    - Profile data:
      - profile.node.0.0 (One per node)
    - Trace data:
      - tautrace.node.trc (One per node)
      - events.node.edf (One per node)

Visualization of traces

- We need to pack all the tautraces.* in one file, and all the events.* in other file:
  - tau_treemerge.pl
    - Generate a unique tautrace file and unique events file
- Generating slog2 file:
  - tau2slog2 tau.trc tau.edf -o tau.slog2
- Finally, the tau.slog2 can be visualized by using Jumpshot application

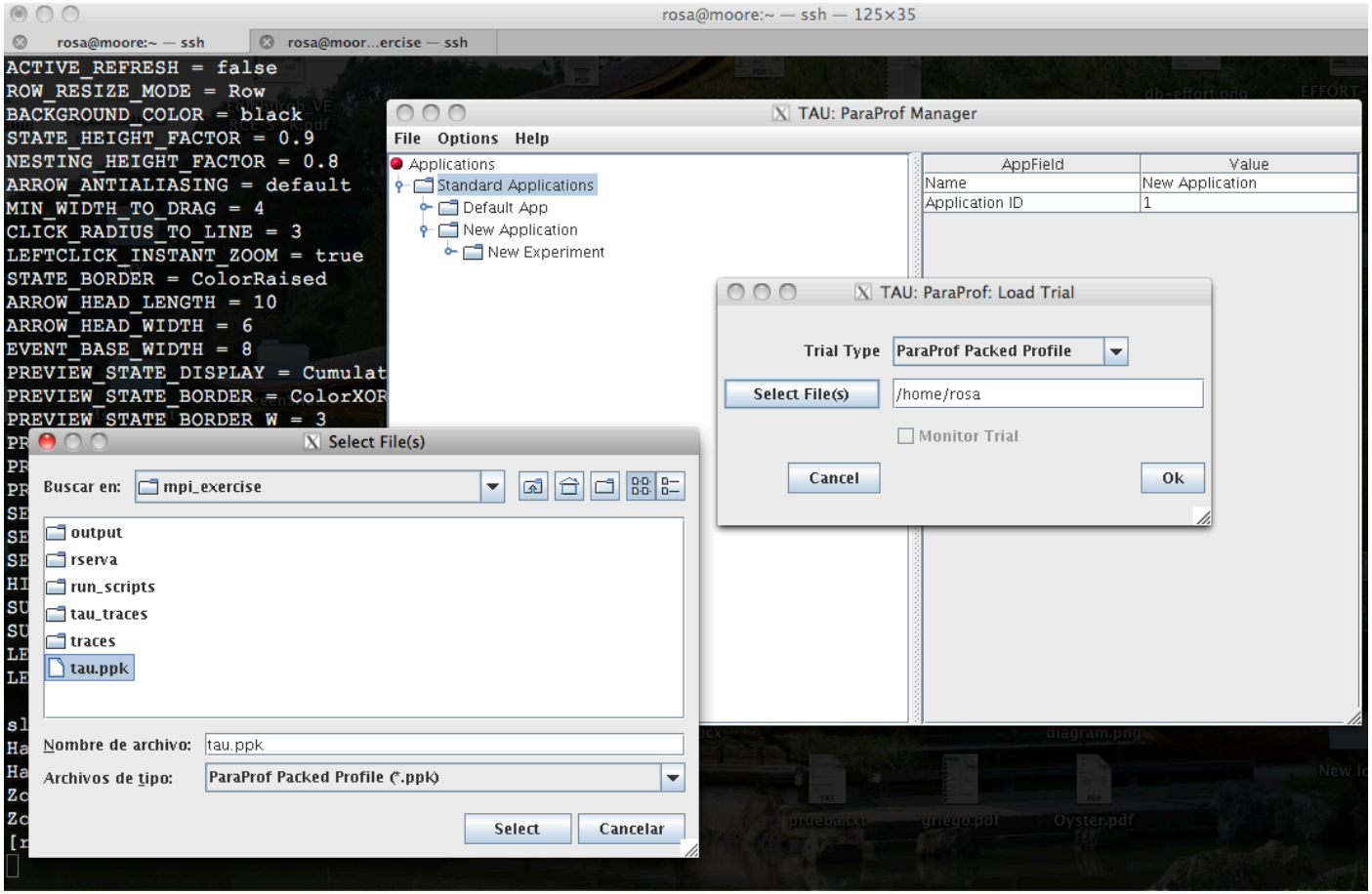
Visualization of traces



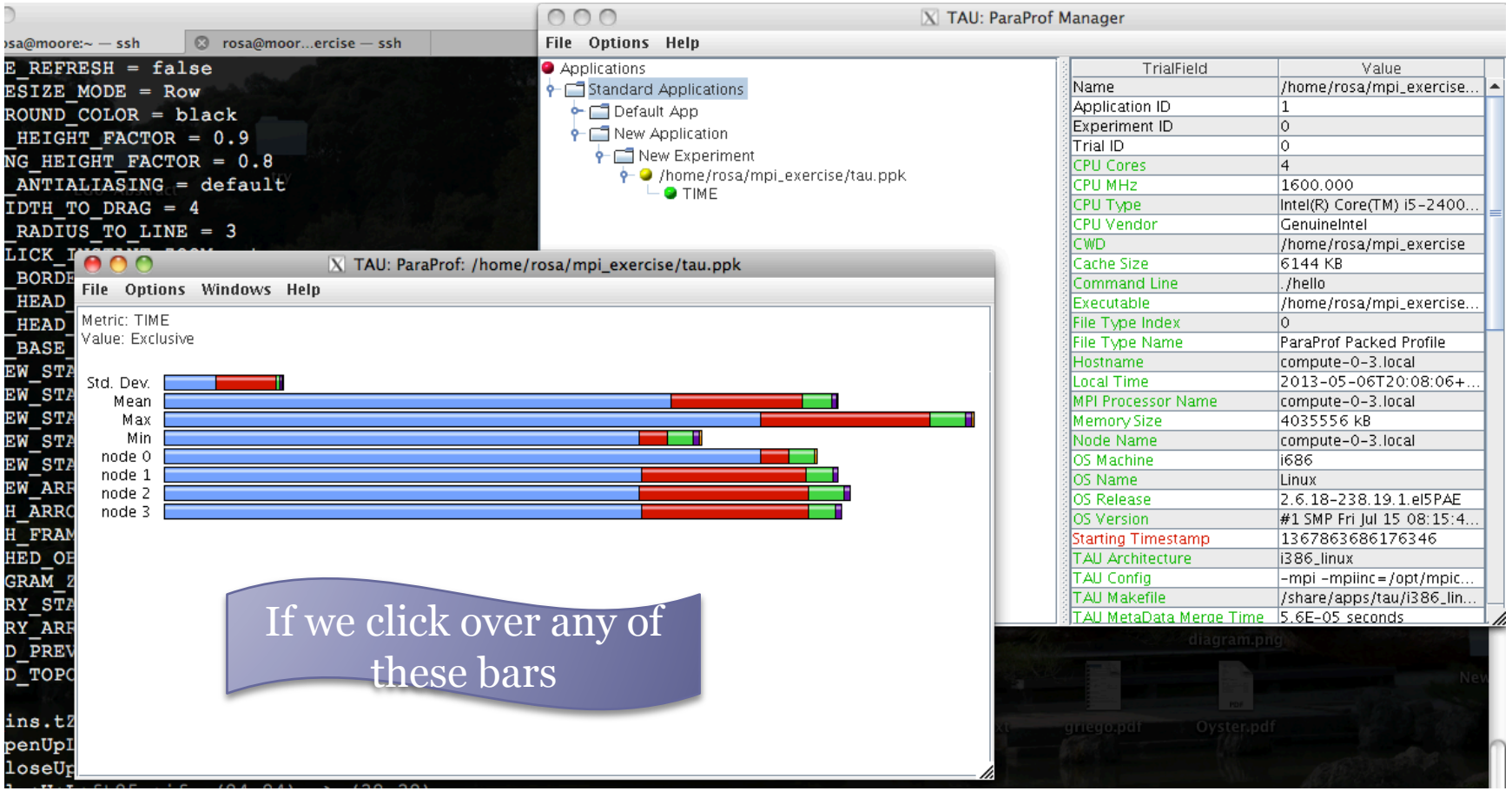
Visualization of profiles

- We need to pack all the profiles.* files in a unique one:
  - paraprof --pack tau.ppk
- Now, we can visualize the tau.ppk file by using Paraprof
  - Open a new terminal by using –X
  - /share/apps/tau/i386_linux/bin/paraprof

Visualization of profiles



Visualization of profiles



Visualization profiles