Jumpshot-4's User's Guide

Anthony Chan[1], David Ashton[2], Rusty Lusk[3], William Gropp[4]
Mathematics and Computer Science Divsion, Argonne National Laboratory

2nd May 2003

[1]chan@mcs.anl.gov
[2]ashton@mcs.anl.gov
[3]lusk@mcs.anl.gov
[4]gropp@mcs.anl.gov

Acknowledgements

We would like to thank Dave Wootton of IBM Poughkeepsie for his valuable suggestions and comments during the development of this tool. This work has been supported in part through the Center for Astrophysical Thermonuclear Flashes at the University of Chicago by the United States Department of Energy under contract B532820. This work was also supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-ENG-38.

Contents

Chapter 1

Introduction

Jumpshot-4 is the visualization program for the improved scalable logfile format, SLOG-2, which provides a hierarchical structure to store a large number of drawable objects in a very scalable and efficient way for visualization. The new scalable logfile format allows the display program to provide functionalities never made possible before. Level-of-detail support through preview drawables which provides high-level abstraction of the details without reading in huge amount of data into the graphical display engine. New Jumpshot allows seamless scrolling from the begining till the end of logfile at any zoom-level. In addition, new functionlities like dragged-zoom, grasp and scroll, instant zoom in/out, easy vertical expansion of timeline, cut and paste of timelines are available as well. A new search and scan facility is provided to locate the hard-to-find objects in very large logfile. The new legend table makes manipulation of the different category of objects easy. The new viewer also attempts to conform to the standard Look and Feel that is expected by most users.

Chapter 2

Data Model

2.1 Understanding the Drawable

The main visual component in the SLOG-2 visualization program, Jumpshot-4, is the *timeline canvas* which is zoomable and scrollable in both horizontal and vertical axes. The timeline canvas can be thought of as a TIMELINE vs TIME coordinate system. Each point on the canvas is identified by two numbers, a timestamp and a timeline ID. The canvas is where the graphical objects contained in SLOG-2 file are being drawn on. These objects are called *Drawables.* There are 2 kinds of drawable objects. They are *Primitive* and *Composite* drawables. The primitive drawables are the simplest drawables and are considered to be basic elements of slog2 file. They are categoried based on their topological structures. Currently, there are 3 topologies supported in SLOG-2. They are *State*, *Arrow* and *Event*. Both state and arrow are drawables identified by 2 points in the timeline canvas, i.e. a pair of (timestamp, timelineID) coordinates. State's start timeline ID is the same as its final timeline ID, but arrow is different from state in the way that arrow's start and final timeline IDs may be different. Event consists of only 1 point in the timeline canvas, i.e. it has only 1 timestamp and 1 timeline ID. Composite drawable is more complicated and is constructed by a collection of primitive drawables[1]. In order to centralize the properties of drawables, all the displayable attributes of a drawable is stored in its corresponding *Category* object, e.g. color, legend name, topology and other shared description of a drawable. Both the category and drawable definitions are stored in the SLOG-2 file. These definitions are interpreted and displayed by the display program, Jumpshot-4.

2.2 Understanding the Preview Drawable

A preview drawable is created as a result of the renormalization process of the SLOG-2 format. The renormalized object provides a high-level description of what is going on within the (timeline vs time) region where the prevew object spans. Preview drawable is designed to amalgamate real drawables of same topological type, e.g. preview state amalgamates only states. So preview drawable is always a primitive drawable in the renormalization scheme. There are currently 3 different types of preview drawables: *Preview_State, Preview_Arrow, and Preview_Event.* Therefore one preview drawable is

[1] In general, composite drawable can be seen as composed of other simpler composite drawables.

for each supported topology of primitive drawable. The three preview categories will always show up in the Legend window of the display program as shown in Figure 3.2 disregarding of the presence of preview drawables in the slog2 file. The Legend window contains a table of legends. Each legend provides an interface to the user modifiable part of the corresponding category that is relevant to the display program.

Figures 2.1 to 2.5 illustrate the visual transition from preview drawable to its detailed content of the first 5 processes of a 16 processes MPI slog2 file when the timeline canvas is being zooming-in. The sequence of figures is generated by zooming in a marked region in each figure in the sequence. The marked region is shaded and is bounded by a pair of white lines. A magnifying glass with plus sign in the center is the cursor that marks the end of the zoom region. Figure 2.1 is a typical timeline canvas where most of real drawables are still buried by their preview drawables. In the figure, there are preview arrows, preview states and some long running real states.

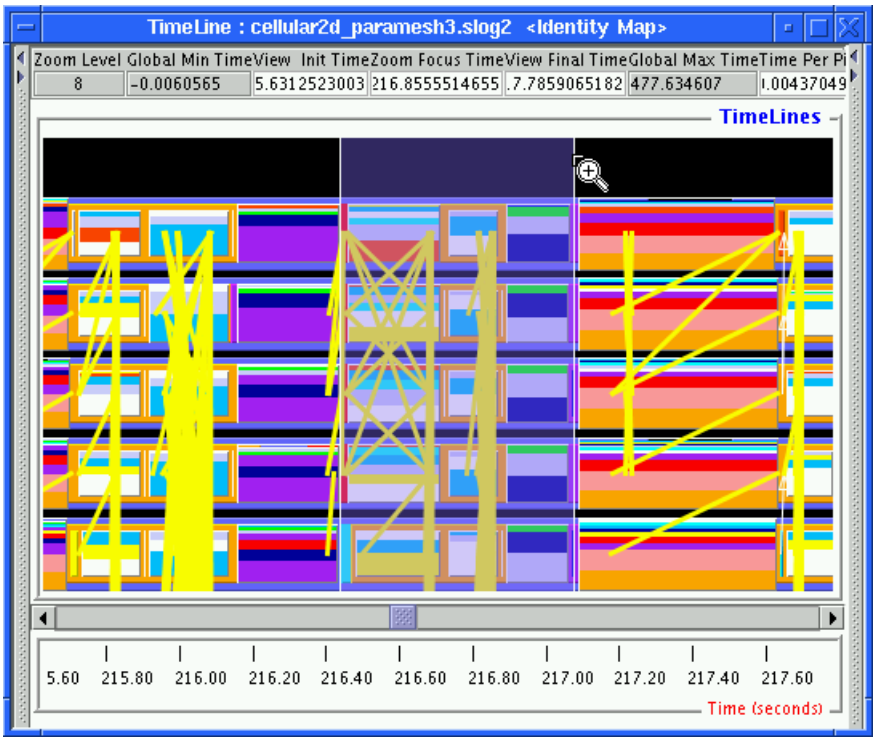


Figure 2.1: A typical zoom-out view of preview states and arrows.

Each thick yellow line is a *preview arrow* which represents a collection of arrows between its 2 ending timelines. The start and final timestamps of preview arrow are the extremes of all real arrows amalgamated inside the preview object. Notice that the beginning or ending timestamp of a preview arrow does not necessarily mean that there is an arrow starting or ending at that time, it just indicates that there are arrows starting and ending within these 2 times and between the 2 marked timelines.

The rectangle that has horizontal strips of colors is *preview state*. The different colors inside a preview state represent the various categories of real states that are amalgamated within the time range of the preview state. Depending on the PREVIEW_STATE_DISPLAY option selected in Preference window as shown in Figure 3.19 and in Table 3.12, the distribution and the heights of the strips can be changed as well. The default display option for preview state is DECRE_WEIGHT_ORDER. In this option,

the strips are arranged in decreasing height order. The tallest strip at the bottom of the preview state corresponds to the category of states that contribute cumulatively the longest duration in specified time range. This visual representation aims to tell what state categories could be within the span of the preview state and which state category contributes the most statistically to the specified time range, so user can decide where to zoom in to find out more details. In a sense, the preview states provide a global coarse-grain summary of what is going on without losing as much details as the preview found in older Jumpshot, i.e. Jumpshot-3. Compared with Jumpshot-3's preview which has averaged out the timeline ID information, the new preview states retain the timeline ID information and that may lead to early detection of load balancing problem before zooming into seeing all the real states.

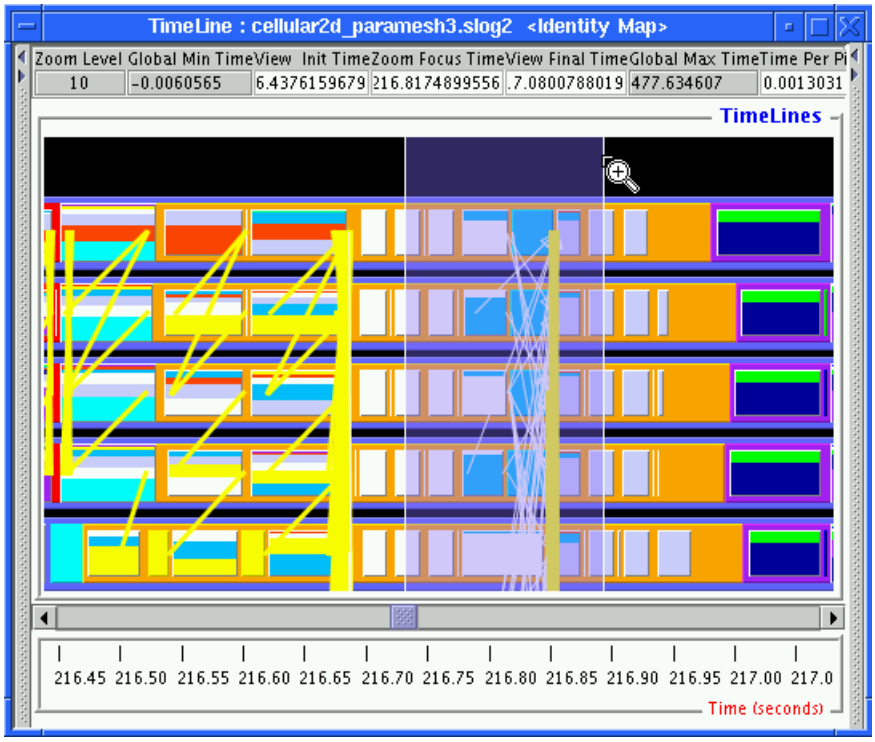


Figure 2.2: The next zoom-in view of Figure 2.1.

Figure 2.2 shows a more zoom-in view of the region marked by the pair of white lines in Figure 2.1. As shown in Figure 2.2, some of the preview arrows have disappeared and are replaced by real arrows, the white arrow. Also, some of the stripped preview states have split into several preview states of one single color, i.e. the white and grey states, to show more detailed distribution. Another important feature of preview state becomes apparent in the figures: Preview states are properly nested within real states. In the most expanded Y-axis label view, preview state is always on top of other nested states[2], i.e. states that enclose the preview state are alway real states. A good visual example is shown in Figure 2.2 where all the white, turquoise and grey preview states[3] are sitting on

[2]Only in slog2 file that has multiple ViewMaps and where timelines can be collapsed, i.e. AIX's UTE generated slog2 file, preview state can be nested with other preview state in collapsed Y-axis label view.

[3]when a preview state contains only real states of one single category, it will appear like a real state in the timeline

top of the long orange and dark royal blue states. This indicates that the white, turquoise and grey real states will be all nested inside the long running orange and dark royal blue states.

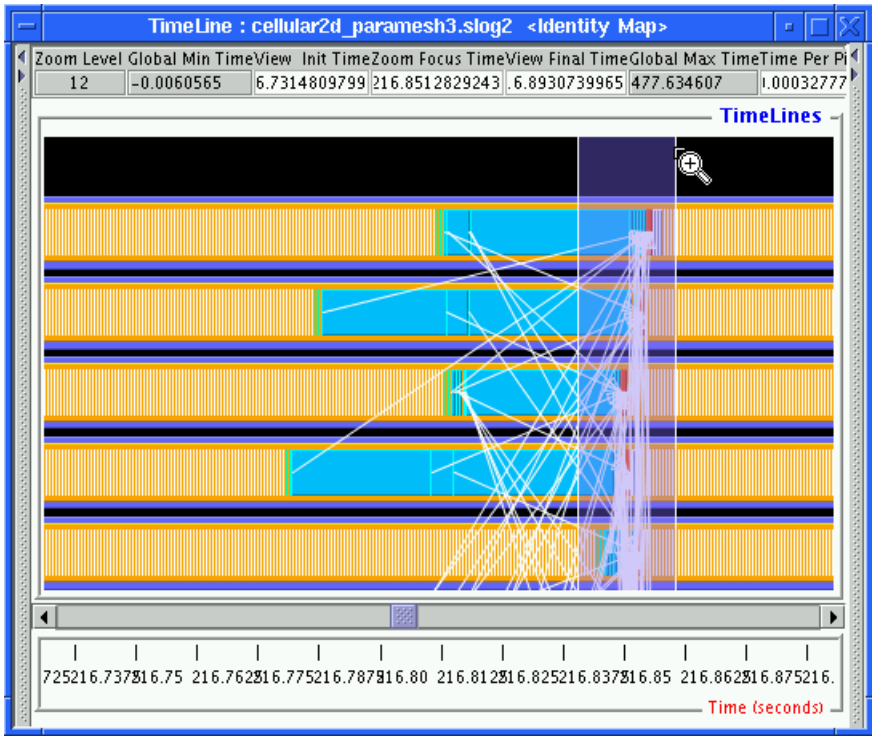


Figure 2.3: The next zoom-in view of Figure 2.2.

Figure 2.3 is the zoom-in view of the region marked by the pair of white lines in Figure 2.2. Comparing these 2 figures, all the preview drawables have disappeared and are replaced by real drawables. Each white preview state are repleaced by hundreds of white real states, the same is also true for the grey preview states that sit to the right of the turquoise states[4]. The preview arrows are all replaced by the real arrows. It becomes apparent that the white lines marked region in Figure 2.2 provides a good description of what is going on in Figure 2.3 but at the same time it reduces the number of drawables drawn on the canvas by a factor of 100. Another way of seeing this benefit is to find out the exact number of real drawables amalgamated by the preview objects within the zoomed region. This can be achieved by right clicking on the preview drawable and the result is shown in Figure 3.13.

Further zooming into the white lines marked region in Figure 2.3 enlarges the real drawables that are displayed in the figure. The enlarged view is shown in Figure 2.4. The densely packed states and arrows become more distinguishable. Another zooming in the whit lines marked region in Figure 2.4 enlarges the real drawables into easily separable objects as shown in Figure 2.5.

canvas. The only way to tell the difference is to bring up the Drawable Info Box by right clicking on the state.

[4]In order to speed up grahics performance of the display program, an aggressive algorithm has been employed to eliminate drawing states that are closely packed together within the nearest neighboring pixels. Together with the fact that the number of pixels available is less than the number of non-overlap states in the region, the number of the real states may sometimes not appear as numerous as the Drawable Info Box of preview state indicates. In that case, a further zoom in will be needed to confirm the case as shown in Fig. 2.4.

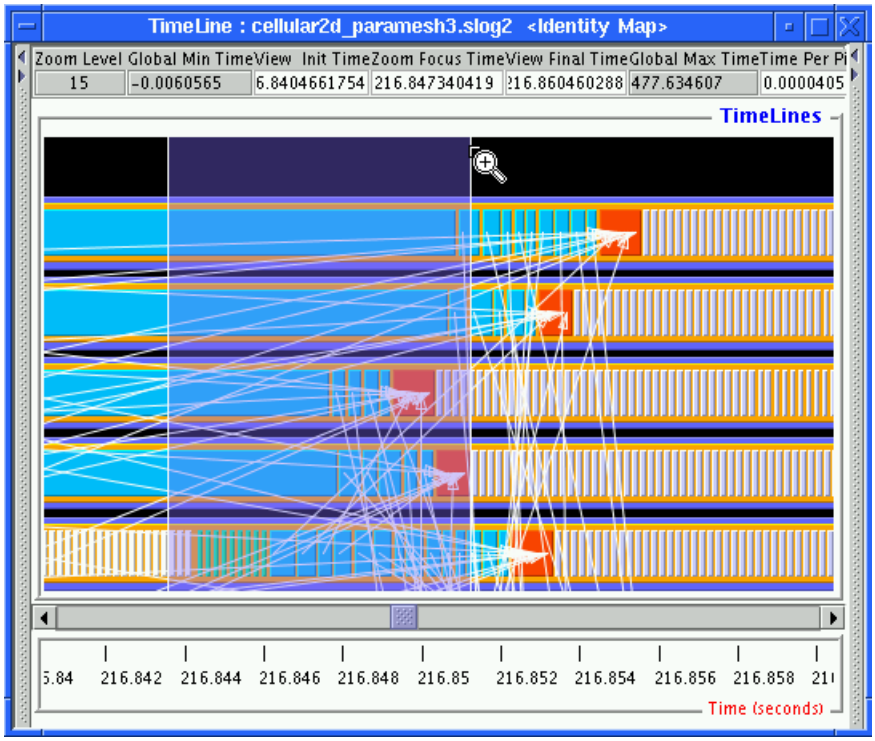


Figure 2.4: The next zoom-in view of Figure 2.3.

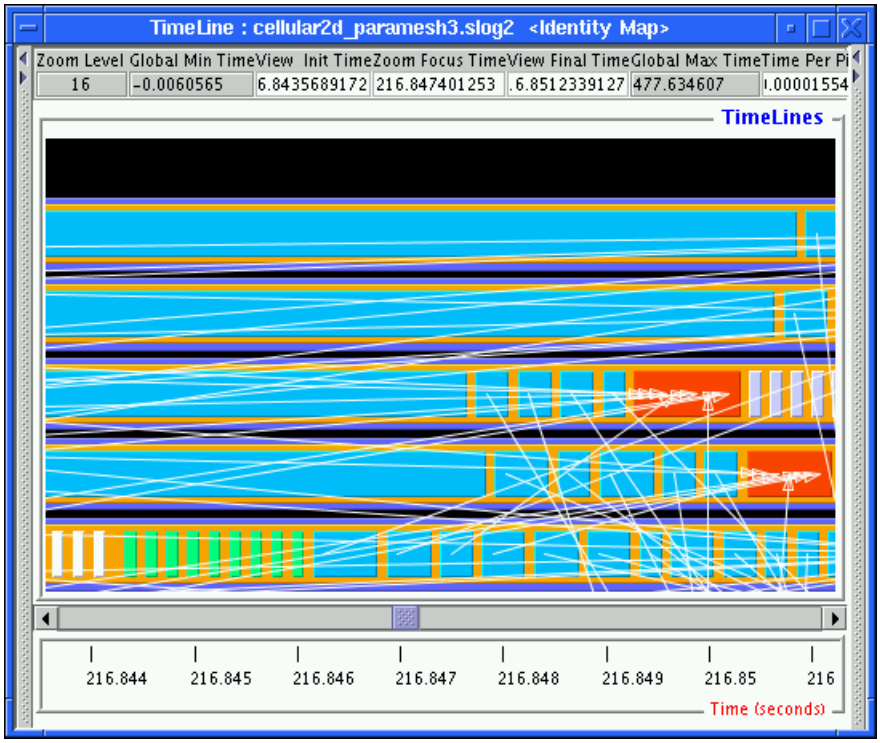


Figure 2.5: The next zoom-in view of Figure 2.4.

Chapter 3

Graphical User Interface

3.1 Main Window

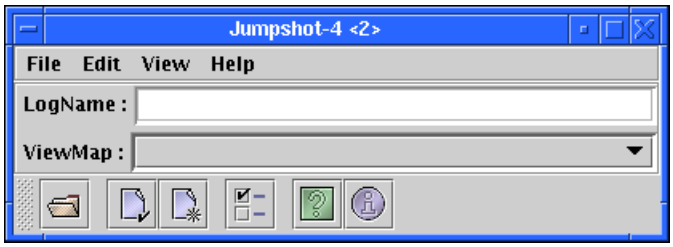


Figure 3.1: The main control window of Jumpshot-4.

The first window that pops up when inovking Jumpshot-4 is called Main window as shown in Figure 3.1. The buttons shown in toolbar are shortcuts to the sub menu items in the top menubar. The function of each of these buttons is listed in the Table 3.2. There are 2 textfields that display crucial information about the logfile being processed. The textfield which is titled *LogName* displays the pathname of the SLOG-2 file being processed. The combobox which is titled *ViewMap* lists all the available ViewMaps in the logfile. Currently, both CLOG[1] and RLOG[2] converted SLOG-2 file contains one ViewMaps, it is called the Identity Map. Only IBM's UTE trace converted SLOG-2 file contains multiple ViewMaps.

3.2 Legend Window

As soon as a SLOG-2 file is selected in Main window, the Legend window like the one shown in Figure 3.2 will be displayed.

The Legend window contains mainly the 4 columns legend table. The 4 columns are labeled as *Topo, Name, V* and *S* as in Table 3.4.

[1] a low-overhead native trace format from MPE.
[2] an internal MPICH2 profiling format

Icon	Description	Function
	File Selection	display a File Chooser dialog to select logfile to be processed
	Show Legend Window	display the Legend window of the selected logfile if it is hidden
	Show Timeline Window	display the Timeline window of the selected logfile if it is hidden
	Edit Preferences	display the Preference window that adjusts Jumpshot's properties
	Show User's Manual	show the User's Manual of this program
	Show FAQs	show the FAQs of this program

Table 3.2: Functions of the toolbar buttons

Icon	Description	Left Mouse Click on Column Cell	Right Mouse Click on Column Cell or Left Mouse Click on Column Title
	Topology	Pick new Color (Figure 3.3)	None
	Name	Edit Name	Sort Order menu (Figure 3.4)
	Visibility	Check or Uncheck	Checkbox Operations Menu (Figure 3.5)
	Searchability	Check or Uncheck	Checkbox Operations Menu (Figure 3.5)

Table 3.4: Operations on the Legend window's columns.

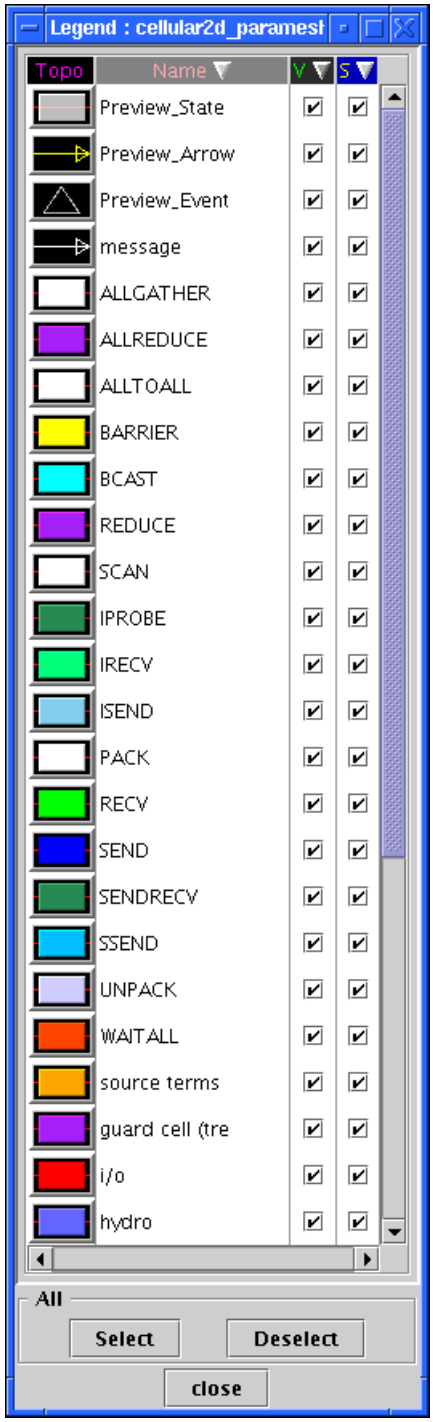


Figure 3.2: A typical Legend window when slog2 file is first loaded into Jumpshot-4.

Table 3.4 also lists out all defined mouse operations that are provided in each column. The operations are left mouse clicking on the column title icon and on the column cell as well as right mouse clicking in any column cell.

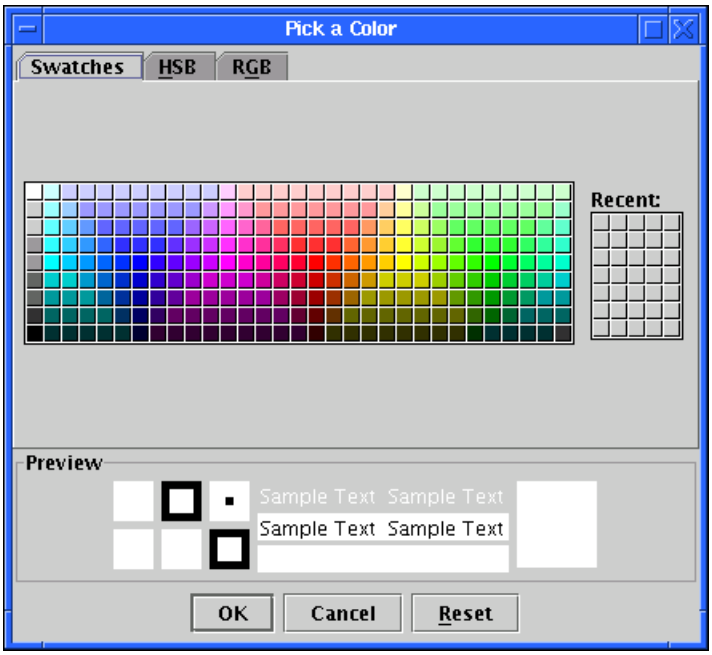


Figure 3.3: Color Chooser Dialog for column Category Topology

Figure 3.3 is the Color Chooser dialog that will pop up when one of the icon buttons in column Topo is pressed. The color editor provides 3 different ways of choosing a new color. After selecting a new color from the dialog, the new color will be used to update the icon button. The update won't be carried out in the timeline canvas automatically, explicit screen redraw is needed.

Figure 3.4 shows the popup dialog box either when the title icon of column Name is pressed or when right mouse button is clicked somewhere in the column. There are altogether 6 different sort orders. The first order in the list is called *Creation Order* which refers to the category order is stored in slog2 file when it is being created. The 4 alphabetical ordering has a hidden sort order that is not mentioned in their names. This hidden order is called *Preview Order* which puts the preview drawable category before all the real drawable categories of the same topology. The word *Topo Order* refers to topologyical order. Taking altogether, "Case Sensitive Topo Order" means first topological order, preview order, then case sensitive alphabetical order. The various orderings serve to facilitate the continuous selection of the category rows in the legend table.

Figure 3.5 shows a popup dialog box when the title icon of column V (Visibililty) or S (Searchability) is pressed or when right mouse button is clicked somewhere in either columns. The rule of selection in the legend table follows the standard practice of other graphical user interfaces as in the Table 3.6. Together with this standard selection rules, the operations provided in checkbox operation menu allow easy enabling and disabling of visibility as well as searchability checkboxes.

NOTE: Any changes done in the Legend window that alters the appearance of drawables won't be automatically updated in the timeline canvas until the CanvasReDraw button in the Timeline window is pressed.

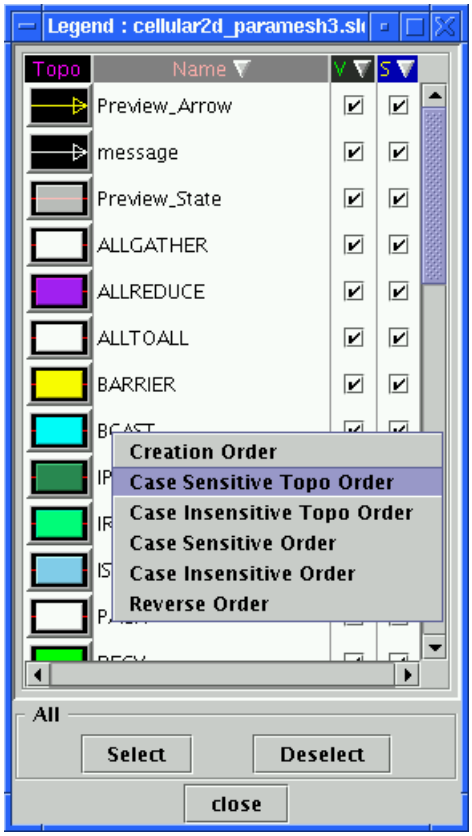


Figure 3.4: Sort Order operation menu for the column Category Name in the Legend window.

Left Mouse Operation	Action
CLICK	*Click* on an object deselects any existing selection and selects the object.
CONTROL-CLICK	*Control-click* on an object toggles its selection without affecting the selection of any other objects
SHIFT-CLICK	*Shift-click* on an object extends the selection from the most recently selected object to the current object.
DRAGGING	*Dragging* (that is, moving the mouse while holding down left mouse button) through a range of TEXT deselects any existing selection and selects the range of text.

Table 3.6: Standard Selection Rules.

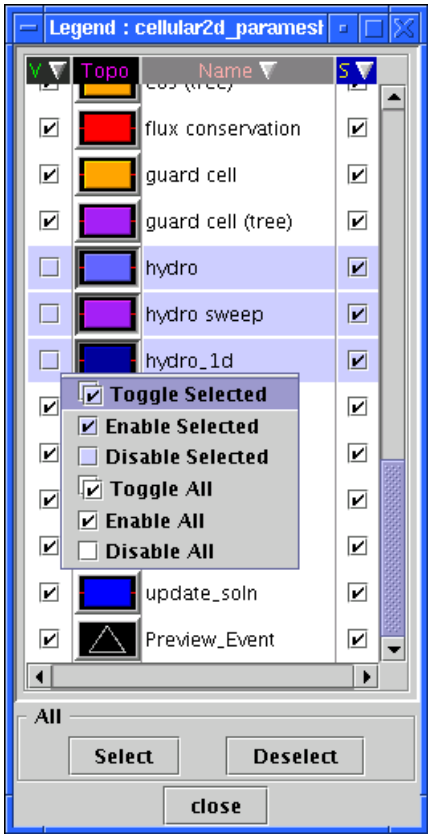


Figure 3.5: Checkbox Operation menu for column Category Visibility and Searchability

3.3 Timeline Window

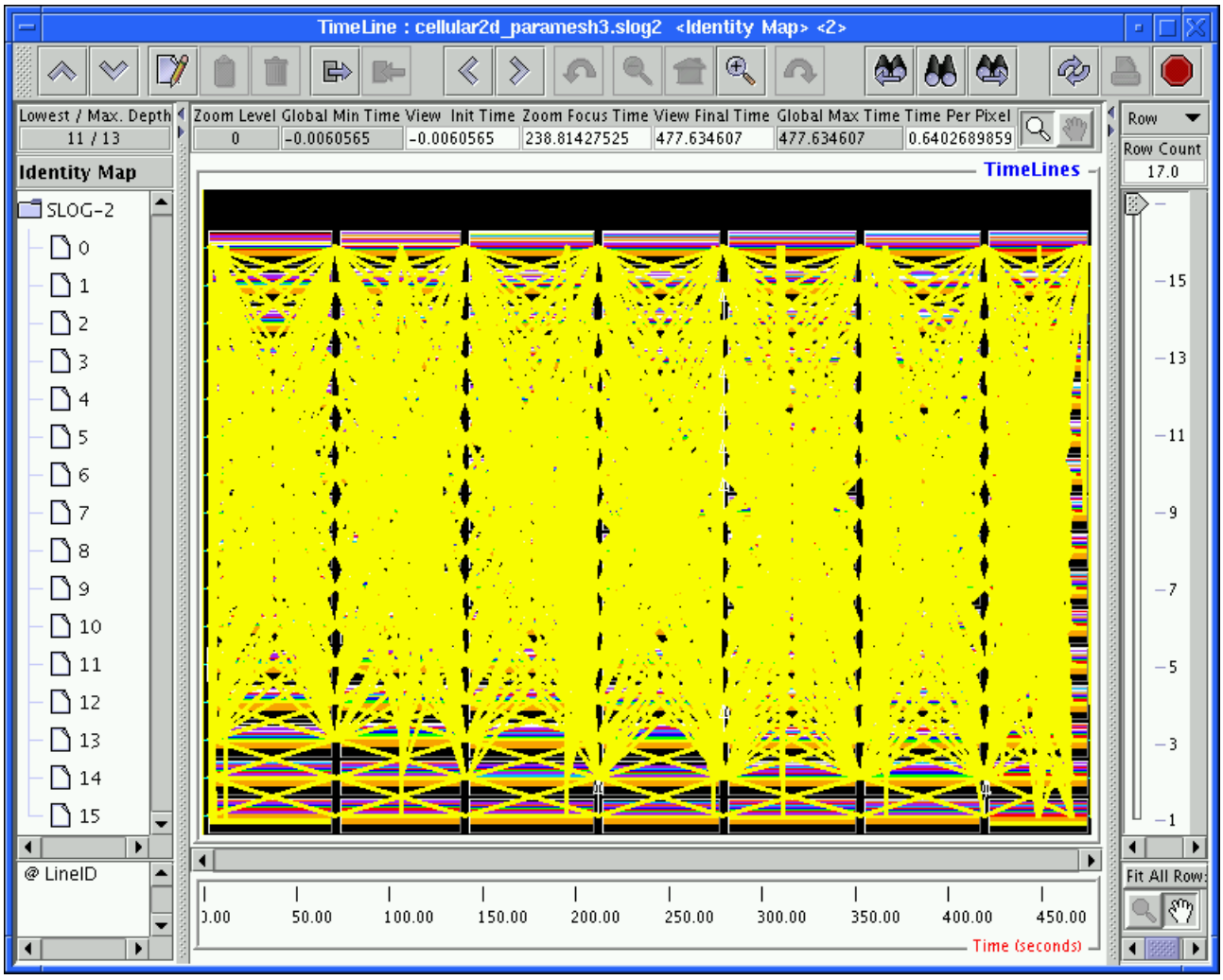


Figure 3.6: The initial display of the Timeline window of a 506 MB 16 processes slog2 file with default preview resolution.

Figure 3.6 is the initial display of the Timeline window of a big 16 timelines slog2 file. Several concealable and removable components are used to create the Timeline window. In the center of the window, there is the *timeline canvas*. Directly on top of the timeline canvas is the *time display panel*. On top of the display panel, there is the removable *toolbar*. To the left of the canvas is the concealable *Y-axis label panel*. To the right of the canvas is the concealable *row adjustment panel*. At the bottom of the canvas is the *time ruler canvas*. Both Y-axis label and the row adjustment panels can be put out of sight by clicking the tabs in the dividers or dragging the dividers. The top toolbar can be dragged out of the window or be repositioned in the other 3 sides of the window. After removal of the toolbar and hidding of the left and right panels, the bare minimal Timeline window looks like the one shown in Figure 2.1.

3.3.1 Zoomable and Scrollable Canvas

When viewing a big slog2 file like the one shown in Figure 3.6, the whole timeline canvas is filled up with preview drawables. Though it provides a reasonable description at high level, i.e. one still gets a vague sense where the long and/or frequent drawables are. Nevertheless, it is pretty obscure to know the details. Hence, a well-designed zoomable and scrollable user interface (ZSUI) of the timeline canvas becomes an absolute necessity to facilitate the location of events of interest. The ZSUI of the timeline canvas includes many parts and operations. But the most handy ones are *dragged zoom*, *grasp and scroll* and *instant zoom in and out.*All these features are supported by the Zoomable and Scrollable canvas. There are 2 such canvases in the Timeline window. They are *Timeline Canvas* and *Time Ruler Canvas*. In these canvases, left mouse clicking can be alternated in 2 different modes by a pair of toggled buttons as shown in Figures 3.7 and 3.8. They are called *Zoom* and *Hand* modes. Each canvas in the Timeline window has its own set of toggled buttons that determine its left mouse click behavior. The timeline canvas's toggled buttons are located above the canvas and sit at the end of the time display panel. The time ruler's toggled buttons are located at the bottom of row adjustment panel, i.e. sit right next to the end of the ruler. By default, the timeline canvas is in zoom mode and the time ruler canvas is in hand mode, so user can do zooming when the cursor is in the timeline canvas and can scroll easily by simply moving the cursor over the ruler canvas. Also, the scrolling can be done by simply dragging on scrollbar's knob, clicking the end buttons and in the space between the knob and scrollbar's end buttons.

Figure 3.7: Canvas's left mouse click is in zoom mode.

Figure 3.8: Canvas's left mouse click is in hand mode.

3.3.1.1 Dragged Zoom

Figure 3.9: Zoom-plus cursor that indicates the left mouse clicking is ready for zooming in.

Dragged zoom is active only when the left mouse click is in zoom mode, i.e. when the the magnifying glass button is pressed in the toggled buttons as in Figure 3.7. In zoom mode, the cursor within the canvas will appear like a magnifying glass with plus sign in the center as in Figure 3.9. It is called zoom-plus cursor. The dragged zoom operation is initialized by pressing the left mouse button at the beginning of the zoom-in region, a white line will then appear. As soon as dragging is detected, another white line will appear to mark the current ending of the zoom-in region. The region that is marked by pair of white lines is lightly shaded as shown in Figure 2.4. The process can be cancelled

anytime by hitting the ESC key during dragging. Once the left mouse button is released, zooming will be carried out and the Timeline window will then be updated as in Figure 2.5. The time display panel is updated with the latest time related information of the zoom-in region. Notice that the zooming as well as scrolling can be achieved by explicitly editing the text fields in the time display panel.

3.3.1.2 Instant Zoom

Figure 3.10: Zoom-minus cursor that indicates the left mouse clicking is ready for zooming out.

While the canvas is still in zoom mode, *instant zoom* is enabled by default. Instant zoom allows zooming in at the point of left mouse clicking by a factor of 1/2, i.e. the region centered at the point of left clicking will be magnified by a factor or 2. Also, the *Zoom Focus Time* in the time display panel will be updated with the time where left clicking on the canvas is detected. In the process, the cursor remains zoom-plus cursor. *Shift-click*, on the other hand, will do the opposite. While holding down Shift key, the cursor will be changed to a zoom-minus cursor as in Figure 3.10 to indicate zooming out is the action associated with left clicking. The zoom factor is 2 in this case.

3.3.1.3 Grasp and Scroll

Figure 3.11: Open hand cursor indicates that left mouse clicking is ready to grasp and scroll.

Figure 3.12: Close hand cursor indicates that left mouse clicking is scrolling.

Grasp and Scroll is active only when the left mouse click is in hand mode, i.e. when the open hand button is pressed as in Figure 3.8. The cursor in hand mode is an open hand as in Figure 3.11. As soon as left mouse button is pressed down, the cursor turns to a close hand as in Figure 3.12. It indicates the canvas will move in the same direction that the cursor moves as long as the left mouse button remain pressed. The grasp and scroll mode in time ruler canvas can only move horizontally, but the grasp and scroll mode in timeline canvas allows movement in both vertical and horizontal axes.

3.3.1.4 Information Dialog Box

Jumpshot-4 wouldn't be complete if it cannot provide a way to tell user what exactly are being displayed. It is particularly important when there are many preview drawables. Following standard

user interface practice, Jumpshot-4 uses *right mouse clicking* as an interface for user to tell Jumpshot-4 what object that more information is needed. In general, anywhere on the canvas, both timeline and time ruler canvases, can be inquired with right mouse clicks. Information dialog box will pop up accordingly to tell user more about object that is being clicked. There are 3 different types of information dialogs: Drawable Info Box, Duration Info Box and Time Info Box. All these info boxes remain in memory as long as they are not closed even if the canvas has been scrolled or zoomed. One of the usages of the info boxes is to serve as time markers in between zooming and scrolling.

Drawable Info Box Drawable Info Box is a popup dialog box that provides detailed information about the drawable object that is being clicked. There are 2 different kind of Drawable Info Box, one for preview drawable, one for real drawable.

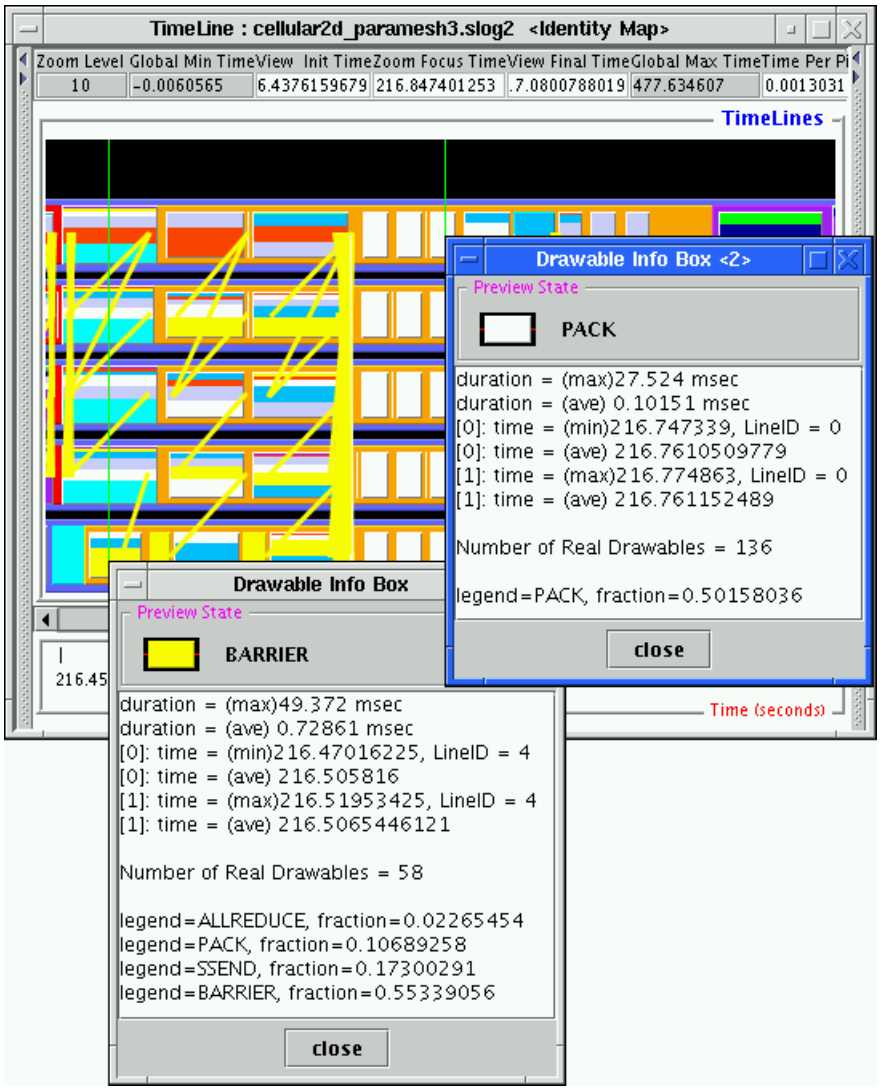


Figure 3.13: Drawable Info Box for Preview State

Drawable Info Box for Preview Drawable Right mouse clicking on 2 of the preview states in the timeline canvas shown in Figure 2.2 will pop up 2 Drawable Info Boxes for the preview states.

They are displayed as in Figure 3.13. The popup Info Box's upper left hand corner will be positioned at exactly where right mouse click is detected and a green line marker is drawn on the canvas to indicate what time has been clicked in case the dialog box is moved from its original popup location. In order to best illustrate what information is presented by the Drawable Info Box, let's take the highlighted Drawable Info Box in Figure 3.13 as an example. The dialog box which contains a pink label "Preview State" is the Drawable Info Box for preview state, and the icon inside the dialog box shows the color and shape that are used to draw the drawable. Below the icon, there is a big text area that prints all the detailed statistical information about this preview state. There are 6 timestamps in the text area: maximum duration, average duration, minimum starttime, average starttime, maximum endtime and average endtime. Here "[0]" refers to starting point, and "[1]" refers to the ending point. The 3 "average" timestamps are averaged over all the real drawables represented by this preview drawable. Besides timestamps, the info box also tells "Number of Real Drawables" represented by the preview object. In this case, 136 real states are amalgamated by the pure white preview state. Also, the text area lists all the different categories of drawables amalgamated and their ratios of the cumulative duration of all real drawables amalgamated to the duration the preview states. In this case, there is only 1 category of real states in this preview state, so all 136 states are all PACKs. The sum of the durations of all PACKs is about half of the duration of the preview state as it is indicated by "fraction=0.50158036" which is called category weight.

Another Drawable Info Box is shown at the lower portion of Figure 3.13. Here the preview state that is pointed by the upper left-hand corner of the Info Box has 3 different strips of colors: yellow, royal blue and white. Right mouse clicking at the yellow strip pops up a Drawable Info Box with a yellow state icon with label BARRIER. As shown in the figure, this preview state amalgmated 4 different categories of real states: ALLREDUCE, PACK, SSEND, and BARRIER. The statistically most significant one is BARRIER which proportionally occupies 0.55339056 of the length of the preview state. Hence BARRIER strip has the tallest height among all the color strips shown in the preview state. Clicking on the different color strip in the same preview state will pop up a Drawable Info Box that has a different icon label of with corresponding color, but the content of the text area should remain the same. Out of the 4 categories mentioned in the text area, only 3 are graphically displayed in the figure given the limited pixel height availabel to the preview state. The least significant category ALLREDUCE is missing visually. But the limitation can be improved by selecting another display option for preview state in Preference window that does not rely on the category weight[3]. As indicated, there are 58 real drawables in the preview state, but no information is provided about how many real drawables are in each real categories.

Drawable Info Box for Real Drawable Similarly for real drawables, Drawable Info Box can be brought up by right mouse clicking on the real drawables. In Figure 3.14, Drawable Info Boxes for a real arrow and a real state are shown. The Drawable Info Box for the arrow is invoked by clicking anywhere within the vicinity of the arrow body[4], and the info box shows the starttime, start timeline ID, endtime, and ending timeline ID and some extra information implemented by the native format . In this example, the message size carried by the specific arrow is 1600 byte.

[3] i.e. by setting the PREVIEW_STATE_DISPLAY in Preference window to MOST_LEGENDS_ORDER as listed in Table 3.12

[4] The vincinity width can be adjusted by modifying the parameter CLICK_RADIUS_TO_LINE in Preference window as listed in Table 3.10. The default is 3 pixels.

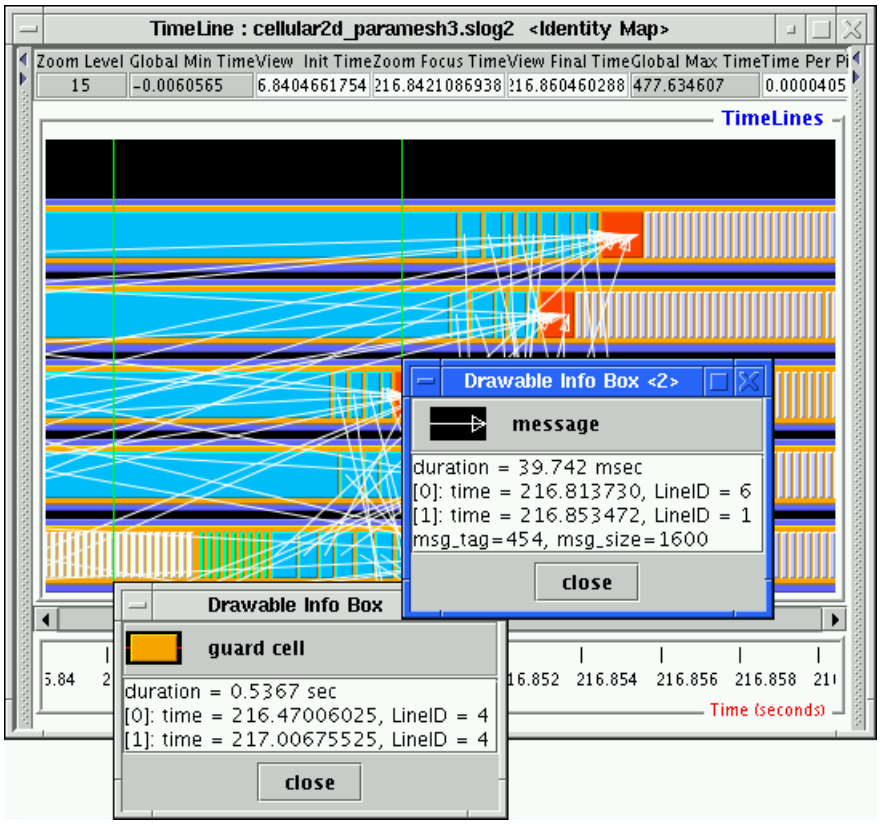


Figure 3.14: Drawable Info Box for real state and arrow. The Drawable Info Box for the arrow shows the messsage size, 1600 byte, and tag ID, 454.

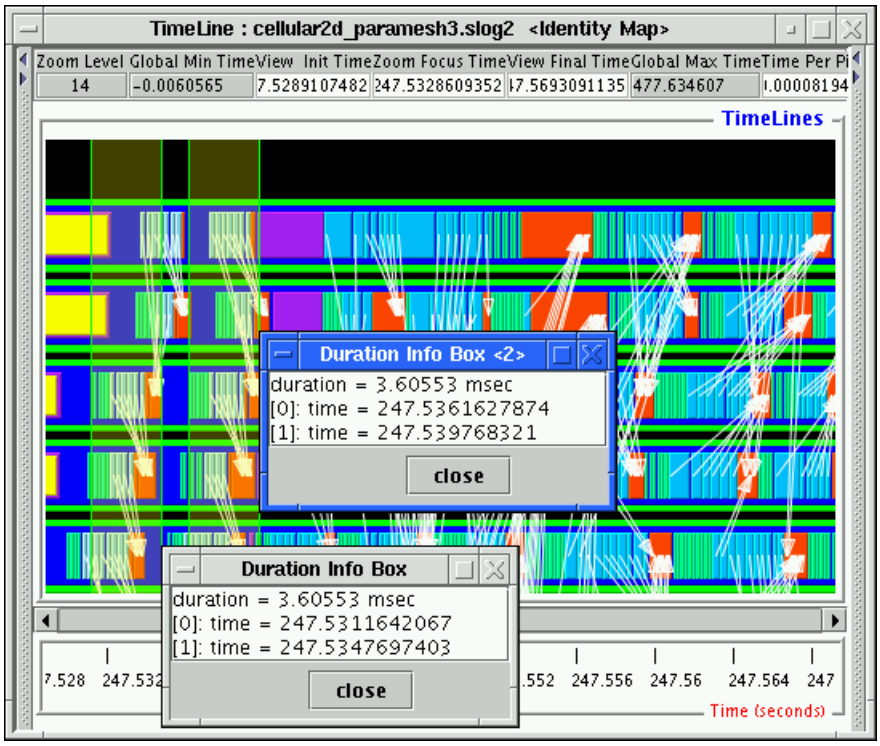


Figure 3.15: Duration Info Box shows the duration, starttime, and endtime of a time region marked by a pair of green lines.

Duration Info Box Duration Info Box is created by right dragging a region of empty space in the timeline canvas or the time ruler canvas. The dragged region will be marked by a pair of green lines and is lightly shaded as well. One of the usages of Duration Info Box is to compare different durations to see if they are the same. For instance in Figure 3.15, the 2 Duration Info Boxes mark 2 non-overlap regions on the 3rd timeline to check if the total duration taken up by the all consecutive small states within the 2 regions are the same. As shown in the Duration Info Box, it appears the total durations are the same.

Time Info Box Time Info Box is created by right clicking in the empty space in either timeline or tthe time ruler canvas as in Figure 3.16. This Info Box is usually used as a marker for a single event in time.

3.3.2 Toolbar

The buttons in the toolbar of Timeline window provides various basic services to the Timeline window. Table 3.8 contains the list of functionalities of the buttons found in the toolbar.

3.3.3 Y-axis Label Panel

The concealable left panel in Timeline window is called Y-axis label panel which contains a tree-like representation for Y-axis label for the timelines. For a single viewmaps slog2 file from CLOG or

Icon	Description	Shortcut	Function
	Up	Alt-UP	Scroll upward by half a screen
	Down	Alt-DOWN	Scroll downward by half of a screen
	LabelMark	none	Mark the timeline(s)
	LabelMove	none	Move the marked timeline(s)
	LabelDelete	none	Delete the marked timeline(s)
	LabelExpand	Alt-E	Expand the Y-axis tree label by 1 level
	LabelCollapse	Alt-C	Collapse the Y-axis tree label by 1 level
	Backward	Alt-LEFT	Scroll Backward by half a screen
	Forward	Alt-RIGHT	Scroll Forward by half a screen
	ZoomUndo	Alt-U	Undo the previous zoom operation
	ZoomOut	Alt-O	Zoom Out by 1 level in time
	ZoomHome	Alt-H	Reset zoom to the initial resolution in time
	ZoomIn	Alt-I	Zoom In by 1 level in time
	ZoomRedo	Alt-R	Redo the previous zoom operation
	SearchBackward	Alt-B	Search backward in time
	SeachInitialize	Alt-S	Search Initialization from last popup InfoBox's time
	SearchForward	Alt-F	Search forward in time
	CanvasReDraw	Alt-D	Redraw canvas to synchronize changes from Preference/Legend window or Yaxis label panel.
	Print	none	Print the Timeline window
	Exit	none	Exit the Timeline window

Table 3.8: Table of toolbar's functionalities.

RLOG, the typical Y axis label panel looks like that is shown in Figure 3.17. Together with toolbar's label buttons, e.g. LabelMark and LabelMove, and standard selection methods by mouse click listed in Table 3.6, labels can be rearranged easily to create a more easily understood timeline canvas. For multiple viewmaps slog2 from IBM's UTE trace enviroment, LabelExpand and LabelCollapse buttons will come in handy to expand and collapse the label tree by one whole level. In order to minimize unnecessary redraw of the timeline canvas, the sychronization between the label panel and the timeline canvas is carried out passively, i.e. user needs to press the CanvasReDraw button in the toolbar to update the Timeline window with the changes from the label panel.

3.3.4 Row Adjustment Panel

The concealable right panel in Timeline window contains the row adjustment panel which is used to determine the row adjustment scheme. There are 2 different modes in row adjustment panel: row count mode and row height mode. These 2 modes can be selected by the combox at the top of the panel. The row count mode attemps to keep the number of timelines constant as indicated in the Row Count text field when the Timeline window resizes. On the other hand, the row height mode fixes the height of each timeline as indicated by the Row Height text field. Currently, the height of the timeline can be adjusted up to the height of the timeline canvas, in that case the Row Count text field shows a number 1[5]. The maximum number of timelines that can be displayed is set to the total number of rows represented by the whole Y-axis label tree[6]. For multiple viewmaps slog2 file, the Y-axis label tree can be expanded or collapsed. This could change the maximum number of rows in the row count slider after user hits the CanvasReDraw button. Coupling with window resize, the row adjustment panel allows user to magnify or shrink the height of the timeline as one desires.

3.4 Preference Window

As shown in Figure 3.19 is the Preference window that adjusts the various display properties of the visualization program. A list of the parameters and their definitions are listed in Tables 3.10 and 3.12.

[5]If the slog2 file contains numerous timelines, increasing the Row Height will increase the size of the images managed by Jumpshot-4. This may cause the Java Virtual Machine to exhaust all its memory if the virtual machine is not set to have enough memory when Jumpshot-4 is started or there isn't enough physical memory in machine that Jumpshot-4 runs on.

[6]Hence the row height cannot be adjusted all the way to zero.

Parameter	Values	Description
Y_AXIS_ROOT_LABEL	any text	Label for the root node of the Y-axis tree label in the left panel.
INIT_SLOG2_LEVEL_READ	+ve integer	The number of slog2 levels being read into memory when the Timeline window is initialized, the integer affects the zooming and scrolling performance exponentially (in a asymptotical sense).
AUTO_WINDOWS_LOCATION	true, false	Whelther to let Jumpshot-4 automatically set windows placement
SCREEN_HEIGHT_RATIO	0.0 ... 1.0	Ratio of the initial timeline canvas height to the screen height
TIME_SCROLL_UNIT_RATIO	0.0 ... 1.0	Unit increment of the horizontal scrollbar in the fraction of timeline canvas's wdith.
Y_AXIS_ROOT_VISIBLE	true, false	Whelther to show the top of the Y-axis tree-styled directory label.
BACKGROUND_COLOR	Black, DarkGray, Gray, LightGray, White	Background color of the timeline canvas
ACTIVE_REFRESH	false	Whelther to let Jumpshot-4 actively update the timeline canvas.
STATE_BORDER	ColorRaised, ColorLowered, WhiteRaised, WhiteLowered, WhitePlain, Empty	Border style of real states.
STATE_HEIGHT_FACTOR	0.0 ... 1.0	Ratio of the outermost rectangle height to row height. The larger the factor is, the larger the outermost rectangle will be with respect to the row height.
NESTING_HEIGHT_FACTOR	0.0 ... 1.0	The gap ratio between successive nesting rectangles. The larger the factor is, the smaller the gap will be.
ARROW_ANTIALIASING	default, on, off	Whelther to draw arrow with anti-aliasing lines. Turning this on will slow down the canvas drawing by a factor of 3.
ARROW_HEAD_LENGTH	+ve integer	Length of arrow head in pixel.
ARROW_HEAD_HALF_WIDTH	+ve integer	Half width of arrow head's base in pixel.
CLICK_RADIUS_TO_LINE	+ve integer	Radius in pixel for a click to be considered on the arrow.

Table 3.10: Modifiable parameters in Preference window.

Parameter	Values	Description
PREVIEW_STATE_DISPLAY	DecreLegendOrder, DecreWeightOrder, MostLegndsOrder	Display option of Preview state.
PREVIEW_STATE_BORDER	ColorRaised, ColorLowered, WhiteRaised, WhiteLowered, WhitePlain, Empty	Border style of Preview state.
PREVIEW_STATE_LEGEND_H	integer	Minimum height of the legend divison in pixel for the Preview state
PREVIEW_STATE_BORDER_W	integer	The empty border insets' width in pixel for the Preview state.
PREVIEW_STATE_BORDER_H	integer	The empty border insets' height in pixel for the Preview state.
PREVIEW_ARROW_LINE_W	float	The line thickness in pixel for the Preview arrow.
MIN_WIDTH_TO_DRAG	integer	Minimum width in pixel to be considered a dragged operation.
SEARCH_ARROW_LENGTH	integer	Length of the search marker's arrow in pixel
SEARCH_FRAME_THICKNES	integer	Thickness in pixel of the popup frame that highlights the searched drawable
SEARCHED_OBJECT_ON_TOP	true, false	Whelther to display the searched object on top of the search frame.
LEFTCLICK_INSTANT_ZOOM	true, false	Whelther to zoom in immediately after left mouse click on canvas.

Table 3.12: Modifiable parameters in Preference window

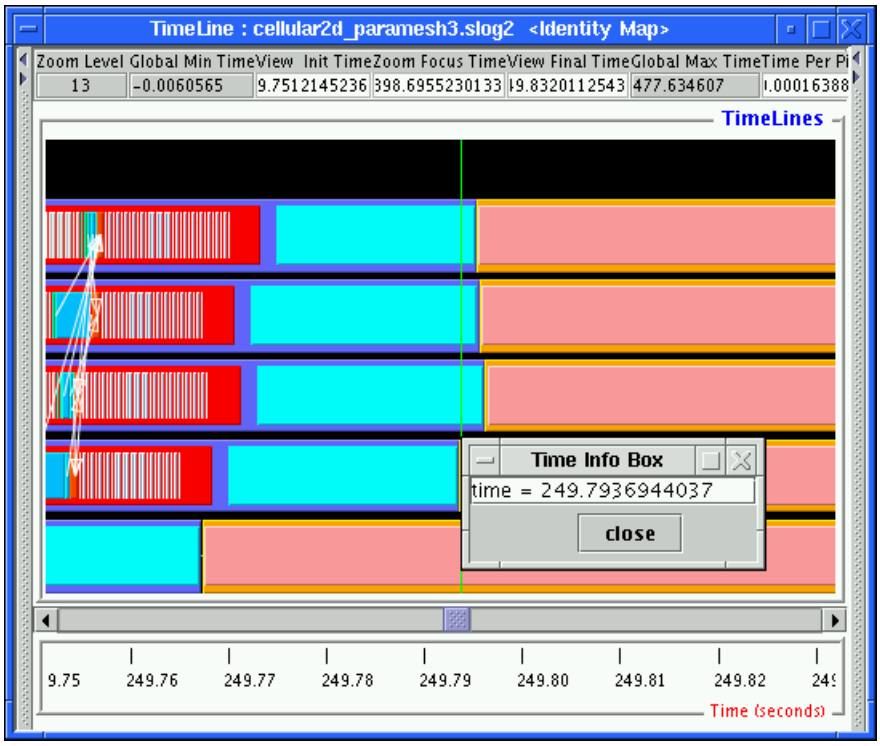


Figure 3.16: Time Info Box displays the time of where it pops up.

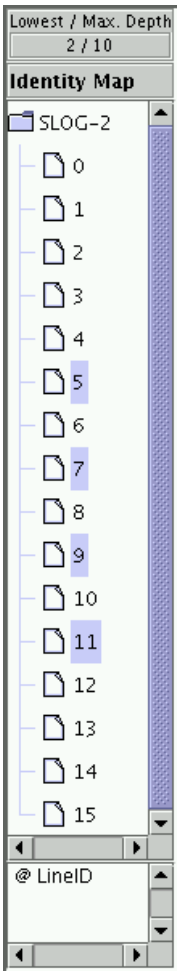
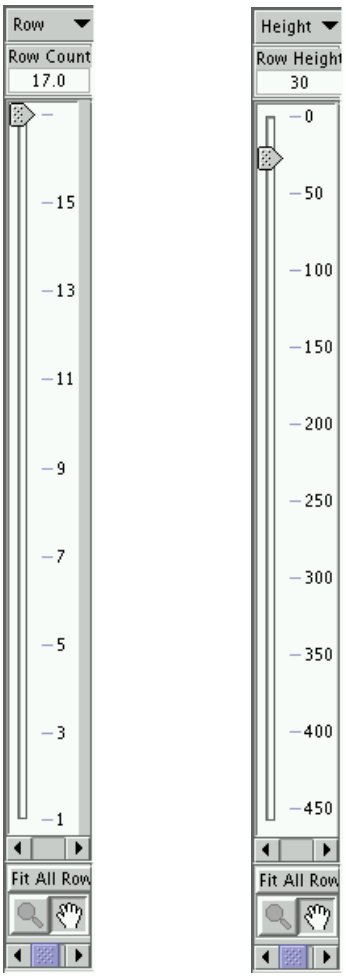


Figure 3.17: A simple 1 level Yaxis label tree. The highlighted labels are those that have been selected.



(a) Row Count mode

(b) Row Height mode

Figure 3.18: Row Adjustmemt Panel determines the Timeline window's resize scheme. When one of the mode sliders or textfields is adjusted, the other 3 components will be adjusted simultaneously.

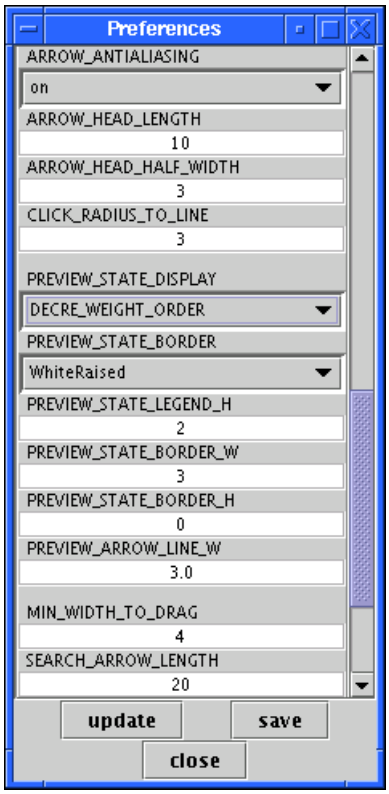


Figure 3.19: The Preference window that shows PREVIEW_STATE_DISPLAY

Chapter 4

Special Features

4.1 Search and Scan Facility

The Level-of-detail support provided in SLOG-2 and Jumpshot-4 tends to help locate states which are either longer in time or occur very frequently. States that are short and occur rarely in a big logfile are very difficult to locate without any special tool. In Jumpshot-4, a search and scan facility is provided to facilitate this goal. There are 3 search criteria: search time, searchable timeline IDs and searchable categories.

1. *Search Time* is the time that search starts. It is marked by a yellow line calld search cursor. There are 2 differnt ways of setting the search cursor. When the timeline canvas is in Hand mode as described in Figure 3.8 of section 3.3.1, left mouse clicking will set the search cursor. The other way can be done in either Hand or Zoom mode. First popup an information dialog box of any kind using right mouse clicking, then press the SearchInitialize button in the toolbar to replace the green line by the yellow search cursor. When there are more than one information dialog box, the information dialog box that is shown up last will have its green line used to initialize the search cursor. When Timeline window first starts up, the search cursor is set at the starttime of the logfile.

2. *Searchable Timeline IDs* are the timlines that search will operate on, i.e only states on the marked timelines will be returned by the search facility. These marked timelines can be selected by clicking on their timeline IDs on Yaxis label panel with rules described in Table 3.6. When nothing is selected, all timelines are searchable.

3. *Searchable Categories* are categories that have their searchable checkboxes enabled as in Figure 3.5. Only drawable with searchable category can be returned by the search facility. By default, all categories in Legend window are searchable.

After setting any needed search criteria, the search operation can be carried out by pressing either the SearchForeward or SearchBackward buttons shown in the Table 3.8. As shown in Figure 4.1, the search facility returns a searched state which is bounded by a 3D raised bordered and transparent

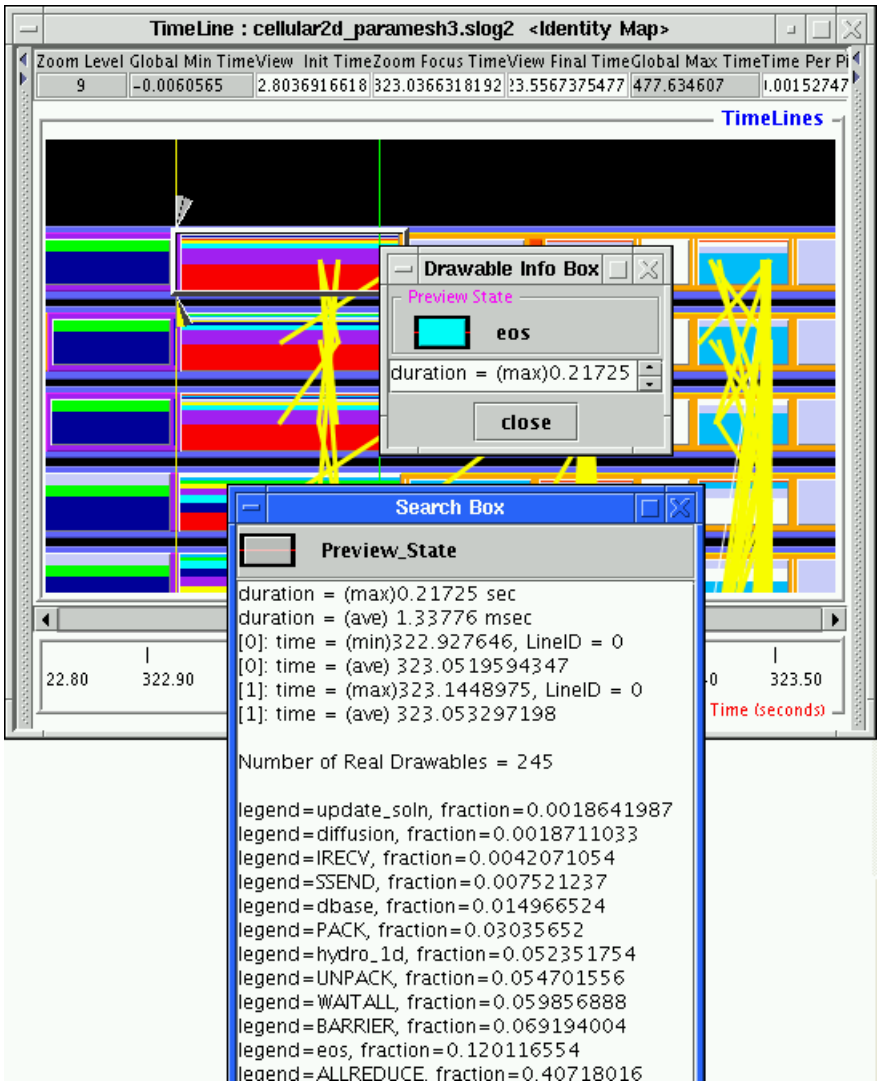


Figure 4.1: Search of state *eos* in preview stage. The returned state is a preview state containing state *eos* as shown in Search Box and Drawable Info Box.

[1]box whose starttime is marked by a yellow search cursor and 2 3D arrowheads. The upper 3D arrow's color matches that of the returned state. In the figure, since the returned state is a preview state, so the upper 3D arrow is grey in color as it is shown in Legend window. Accompanied with the 3D raised bordered box is a popup Search Box that shows the detailed of the preview state like the Drawable Info Box in Figure 3.13. Since the search in the figure is looking for state *eos*, a Drawable Info Box is shown to indicate the returned 3D bordered box does contain category *eos* graphically. In order to locate the real state *eos*, a dragged zoom is performed around the 3D raised bordered box and the result is shown in Figure 4.2. In the figure, the real *eos* is located at the end of the original 3D bordered box and it is marked by the Drawable Info Box.

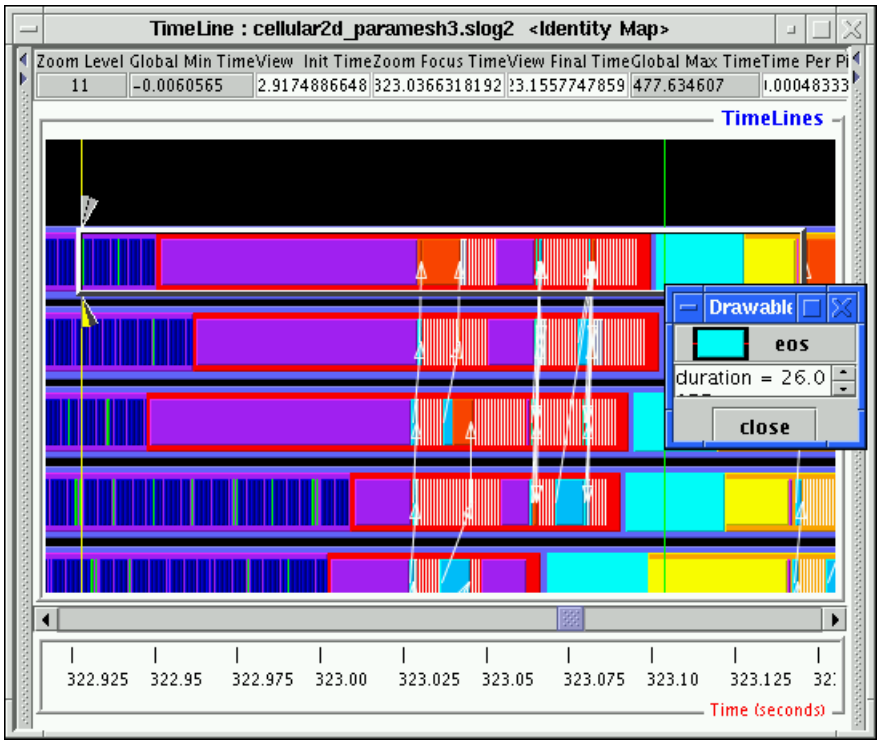


Figure 4.2: Dragged zoom the region around the 3D raised bordered box in Figure 4.1 shows the real state *eos*.

In general, when searching on big slog2 file, all preview categories should be set searchable, otherwise searching for real drawables may not return anything. Because at lower zoom level, there may not be any real drawable belong to the categories of interest, only preview drawable contains the categories of interest. Also, the search facility is carried out for the drawables that are in the physical memory. In some rare occasions, drawables in the memory may have been exhausted for searching but end of the logfile have not been reached, user may need to advance the search by scroll forward or backward to read in more drawables and to restart the search again. For a very big logfile, the search process of a real state may require repeated operations of search and dragged zoom before the real state can be found. This process will be automated in the later version of Jumpshot-4.

[1]the transparency of the 3D raised box can be made opaque by selecting the SEARCHED_OBJECT_ON_TOP true.

4.2 Tuning of the Timeline Window

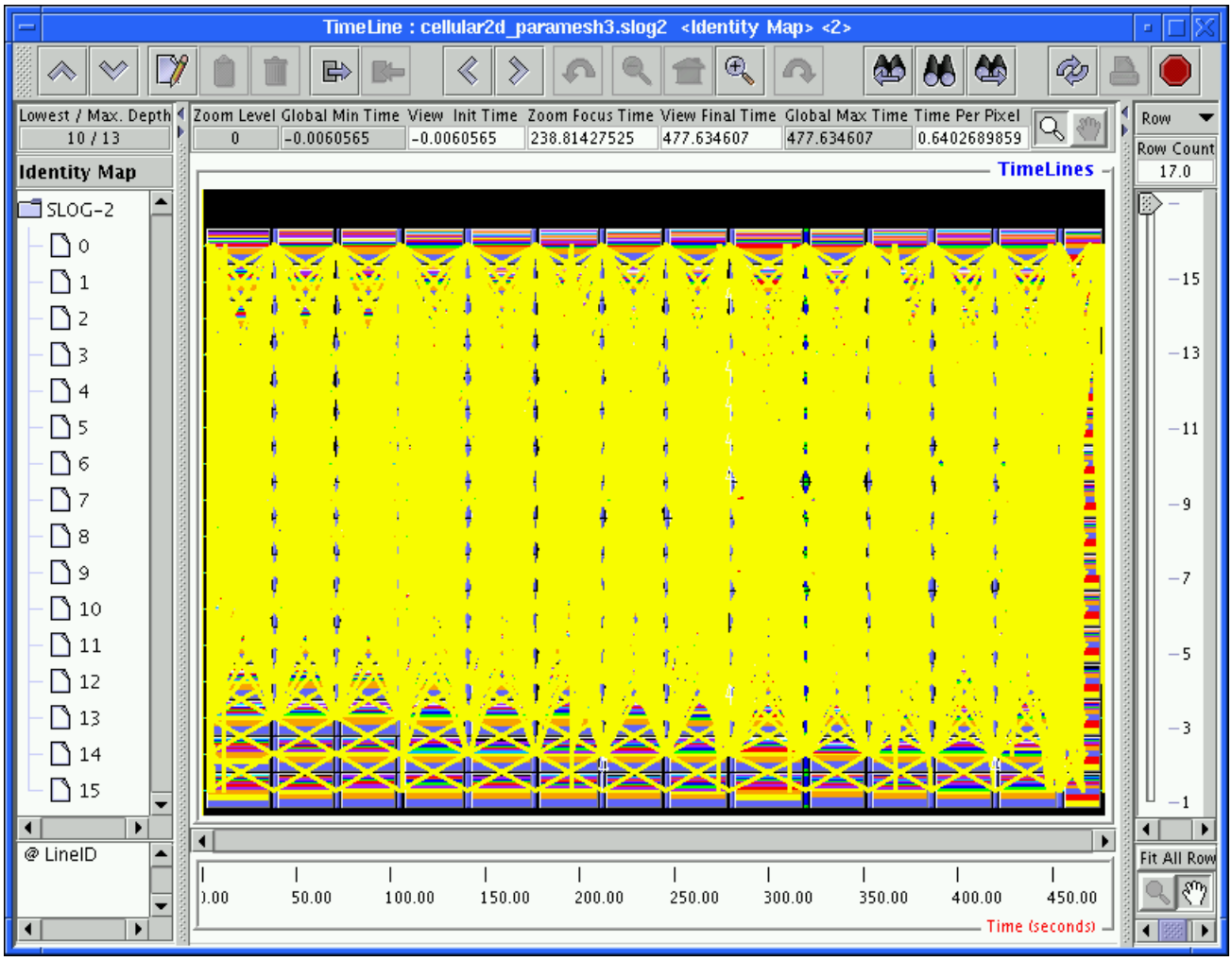


Figure 4.3: The initial Timeline window with finer preview resolution.

One of the major improvements in the new Jumpshot and SLOG-2 is the scalability in terms of visualization performance. As shown in Figure 3.6, there are 7 bands of preview states covering the whole canvas in the initial Timeline window. By incrementing the parameter, INIT_SLOG2_LEVEL_READ, by 1 in Preference window as in Table 3.10, the initial Timeline window can be redisplayed and is shown in Figure 4.3. The new Timeline window has 14 bands of preview states instead of 7. The increase preview resolution does offer a more detailed description of logfile but at the expense of graphical performance. Because every increment of INIT_SLOG2_LEVEL_READ will roughly double the number of drawables to be iterated during every zooming or scrolling[2]. The biggest demand of graphical performance occurs when the zooming to the level of only pure real drawables from the lower zoom level. The value of INIT_SLOG2_LEVEL_READ should be chosen so that Jumpshot does not appear to be too slow during the zooming to the pure real drawable level. For a fast graphics system, INIT_SLOG2_LEVEL_READ should be set higher than the default value 3, like 4 or 5, so that more information is present during each view.

[2]it is true for a binary tree

Another parameter also signifcantly affects the graphical performance is ARROW_ANTIALIASING in Preference window. Setting the paramters to ON will force Jumpshot to draw all arrows including preview arrows with anti-aliasing lines. This proves to be an expensive graphical operation[3]. Except when high quality picture is needed like during screen capture for picture or when antialiasing lines are drawn with graphics hardware support, it is not recommended to turn on ARROW_ANTIALIASING.

[3]A typical timeline canvas with arrows will draw roughly a factor 3 slower with anti-aliasing on.