
Advanced Tracing Features using GDB and LTTng

GCC Summit 2010
2010-10-26

francois.chouinard@ericsson.com

marc.khouzam@ericsson.com

Summary

- › What is Tracing?
- › The Linux Tracing Toolkit next generation (LTTng)
- › User-space Tracer (UST)
- › Eclipse Tracing and Monitoring Framework
 - LTTng Eclipse Integration
 - Perspective and Views
 - Upcoming Features
- › Integration with GDB Tracepoints
- › Getting Eclipse to work for you in under a minute

What is Tracing?

- › Technique used to collect info to debug/monitor a system
- › Often used when debuggers are too intrusive
 - Real-time system
 - Live system
 - Race condition
- › Also used for performance tuning
- › Like logging but records events that:
 - happen much more frequently (gigabytes of data collected)
 - can be at a much lower level (system calls, scheduling, interrupts)
- › Requires efficient visualization tool to make sense of the vast amount of data collected

Interest in Tracing

- › The interest in tracing is growing tremendously.
- › Involvement from such companies :
 - Google
 - QNX
 - CodeSourcery
 - FreeScale
 - MentorGraphics
 - Texas Instrument
 - Red Hat
 - WindRiver
 - Ericsson
 - etc...

Linux Tracing Toolkit

- › Open-source project (<http://www.lttng.org>)
- › Aims at producing a highly efficient full system tracing solution
- › Composed of several components
 - Kernel tracing
 - User-space tracing
 - Trace viewer
 - Trace streaming
- › Partially already included in the Linux Kernel

Linux Tracing Toolkit

- › Highly optimized static tracepoints
- › Highly compact binary trace format
- › Efficient probes which do not use traps or system calls
- › Almost zero performance impact with instrumentation points disabled
- › Active instrumentation points have low performance impact
- › Zero copy from event generation to disk write
- › ...

User-Space Tracer (UST)

- › Static tracepoints for user-space
- › Port of the LTTng kernel tracer to user-space
- › Like LTTng performance is the main goal
 - Tracing does not require system calls or traps
- › Tracepoints may be added in any user-space code
 - Multi-threaded applications
 - signal handlers
 - libraries
- › Programs must be compiled with libust (-lust)

User-Space Tracer (UST)

- › Example of Marker in servers/slapd/search.c

```
int
do_search(
    Operation    *op,    /* info about the op to which we're responding */
    SlapReply    *rs /* all the response data we'll send */ )
{
    struct berval base = BER_BVNULL;
    ber_len_t    siz, off, i;

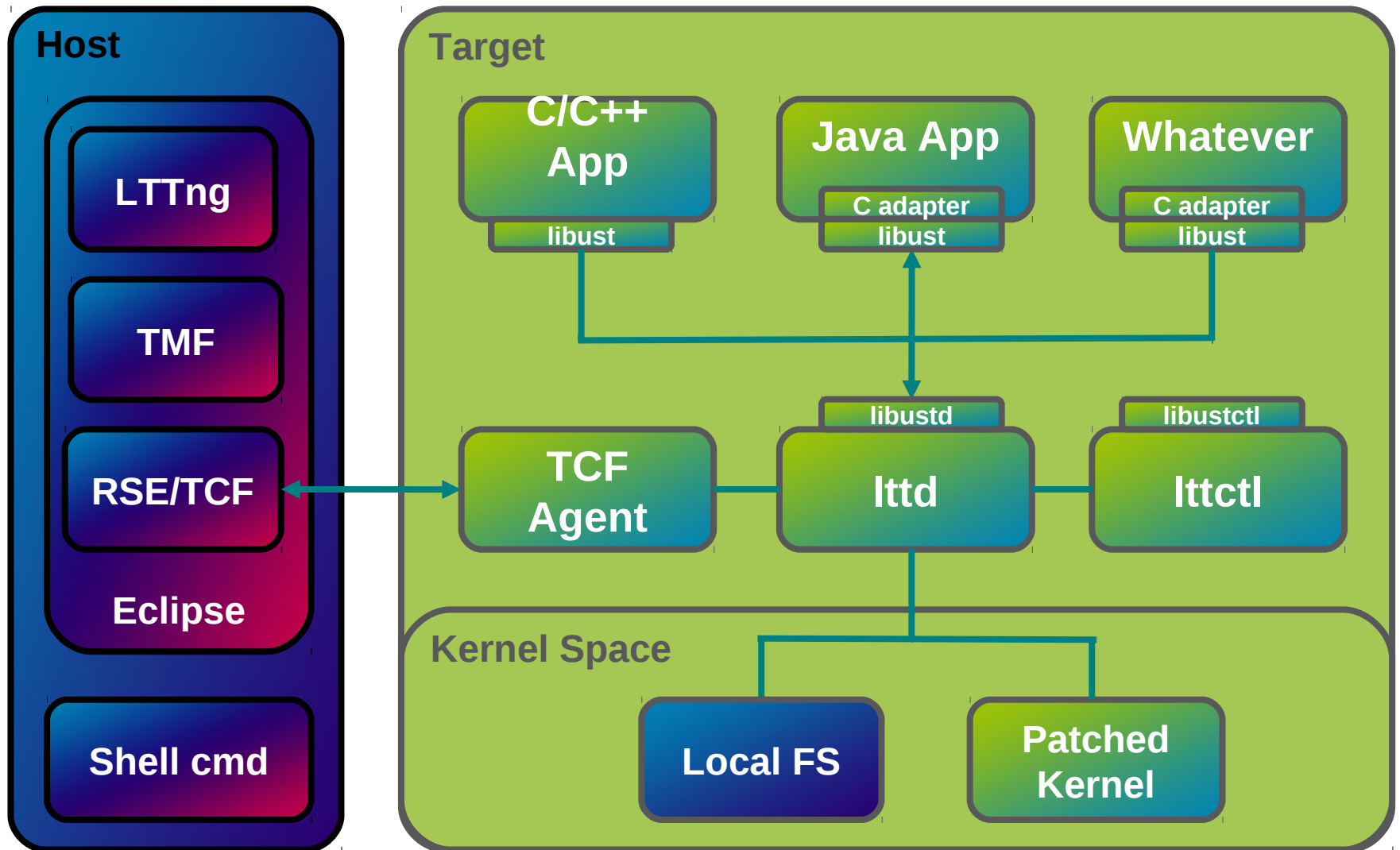
    trace_mark(ust, search_event, "DN %s", op->o_req_dn.bv_val);
}
```

- › Command: `ldapsearch -b "dc=rlnx,dc=com"`
- › Marker output: `{ "DN" = "dc=rlnx,dc=com" }`
- › Can be controlled and used by GDB for static tracing

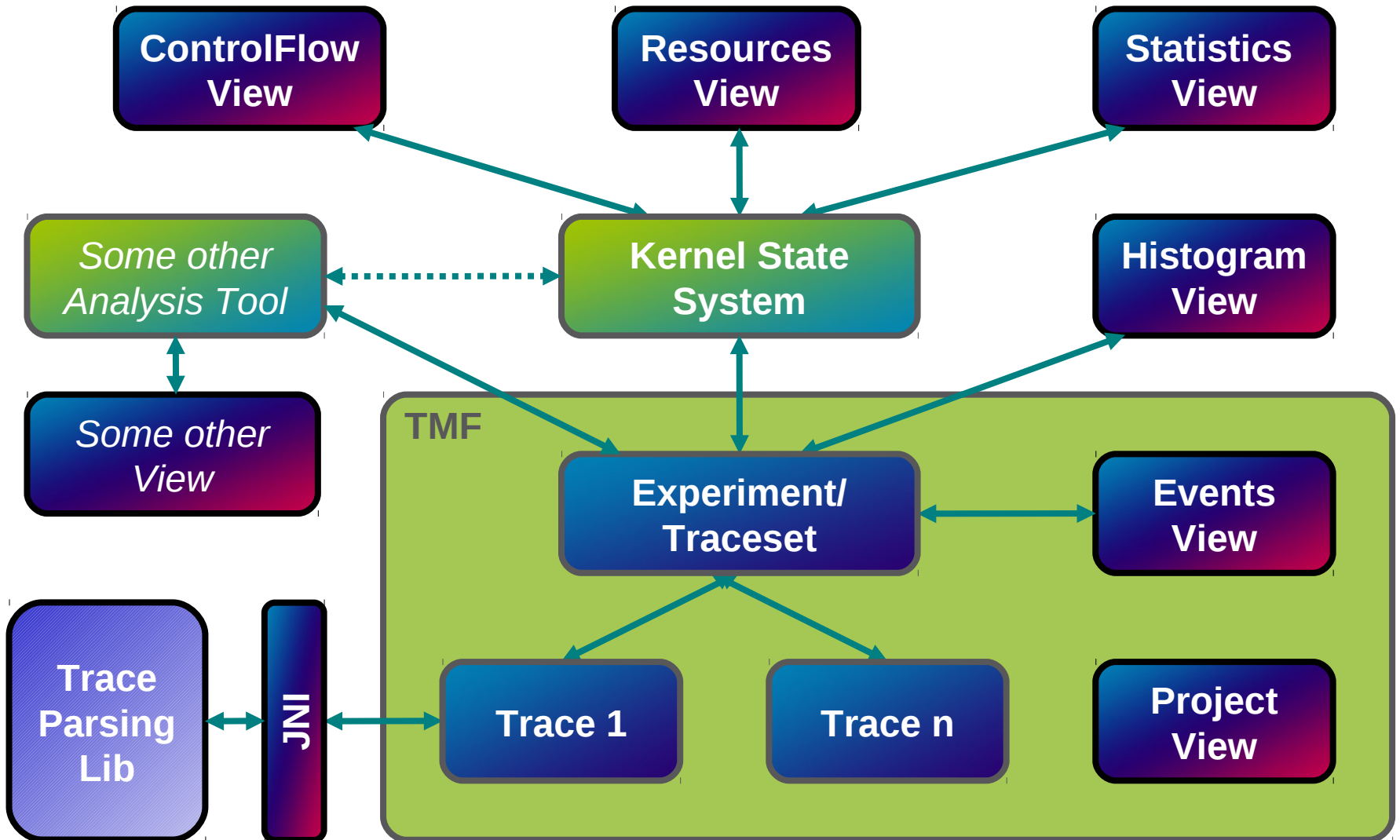
Trace visualization

- › Originally, making use of LTTng with LTTv
- › Integration of many different tools in Eclipse makes Eclipse a better fit
- › Focus is now on the new LTTng integration in Eclipse

LTTng Eclipse Integration



LTTng Eclipse Architecture



LTTng Perspective

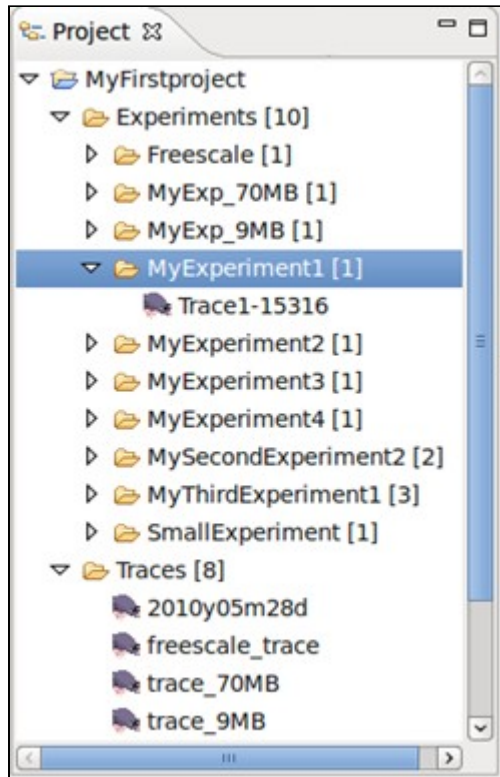
The screenshot displays the LTTng GUI interface, which is divided into several panes:

- Control Flow:** A table showing process execution details. The selected row is for the `ltd` process.
- Resources:** A timeline view for the selected process, showing activity on various system resources like CPU, IRQs, and traps.
- Events - MyExperiment1:** A table of recorded events with columns for timestamp, source, type, reference, and content.
- Statistics:** A tree view showing the number of events for different categories like CPUs and Event Types.
- Histogram:** A bar chart showing the distribution of event times over a specific window.

Process	pid	ppid	tid	ppid	uid	gid	Trace
sendmail	16867	16867	16857	0	13589	763749454	Trace1-15316
ping	16885	16885	16865	0	13589	763752479	Trace1-15316
ltd	16887	16887	30068	0	13589	763755140	Trace1-15316
UNNAMED	16888	0	0	0	0	000000000	Trace1-15316
ltd	16889	16889	1	0	13589	763758054	Trace1-15316
ltd	16890	16889	16889	0	13589	781038220	Trace1-15316
/usr/local/bin/ltd	16891	16891	30068	0	13589	873239052	Trace1-15316
udev	18054	18054	1	0	13589	763696784	Trace1-15316

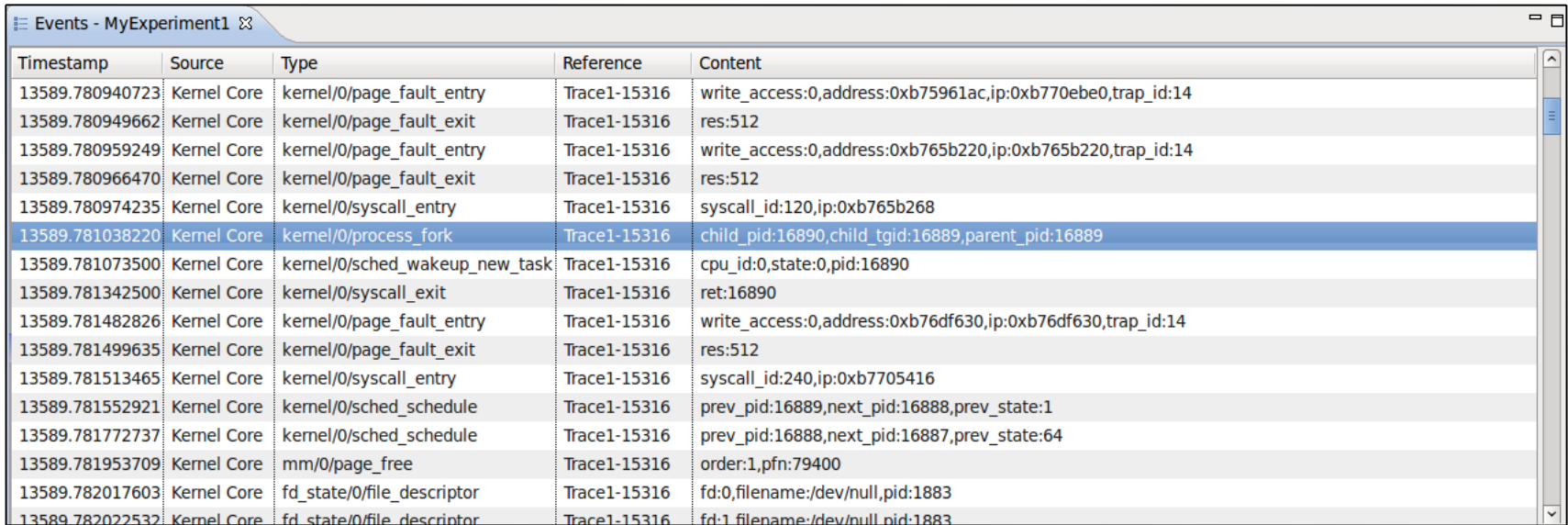
Timestamp	Source	Type	Reference	Content
13589.780940723	Kernel Core	kernel/0/page_fault_entry	Trace1-15316	write_access:0,address:0xb75961ac,ip:0xb770ebe0,trap_id:14
13589.780949662	Kernel Core	kernel/0/page_fault_exit	Trace1-15316	res:512
13589.780959249	Kernel Core	kernel/0/page_fault_entry	Trace1-15316	write_access:0,address:0xb765b220,ip:0xb765b220,trap_id:14
13589.780966470	Kernel Core	kernel/0/page_fault_exit	Trace1-15316	res:512
13589.780974235	Kernel Core	kernel/0/syscall_entry	Trace1-15316	syscall_id:120,ip:0xb765b268
13589.781038220	Kernel Core	kernel/0/process_fork	Trace1-15316	child_pid:16890,child_tgid:16889,parent_pid:16889
13589.781073500	Kernel Core	kernel/0/sched_wakeup_new_task	Trace1-15316	cpu_id:0,state:0,pid:16890
13589.781342500	Kernel Core	kernel/0/syscall_exit	Trace1-15316	ret:16890
13589.781482826	Kernel Core	kernel/0/page_fault_entry	Trace1-15316	write_access:0,address:0xb76df630,ip:0xb76df630,trap_id:14
13589.781499635	Kernel Core	kernel/0/page_fault_exit	Trace1-15316	res:512

LTTng – Project View



- › Projects are used to group traces that you wish to correlate
- › Experiments are specific correlations between selected trace files
- › Traces are all trace files currently included in the project

LTTng – Events View

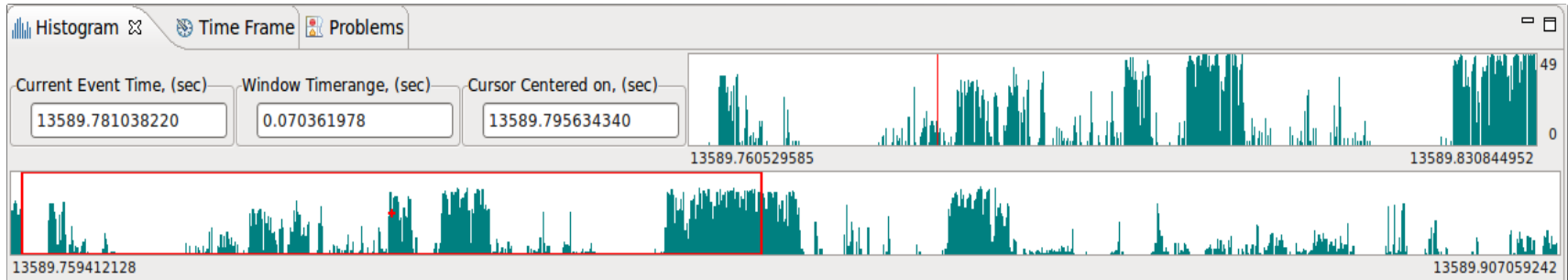


The screenshot shows the 'Events - MyExperiment1' window with a table of kernel events. The table has five columns: Timestamp, Source, Type, Reference, and Content. The events are sorted chronologically by timestamp. The event at timestamp 13589.781038220 is highlighted in blue.

Timestamp	Source	Type	Reference	Content
13589.780940723	Kernel Core	kernel/0/page_fault_entry	Trace1-15316	write_access:0,address:0xb75961ac,ip:0xb770ebe0,trap_id:14
13589.780949662	Kernel Core	kernel/0/page_fault_exit	Trace1-15316	res:512
13589.780959249	Kernel Core	kernel/0/page_fault_entry	Trace1-15316	write_access:0,address:0xb765b220,ip:0xb765b220,trap_id:14
13589.780966470	Kernel Core	kernel/0/page_fault_exit	Trace1-15316	res:512
13589.780974235	Kernel Core	kernel/0/syscall_entry	Trace1-15316	syscall_id:120,ip:0xb765b268
13589.781038220	Kernel Core	kernel/0/process_fork	Trace1-15316	child_pid:16890,child_tgid:16889,parent_pid:16889
13589.781073500	Kernel Core	kernel/0/sched_wakeup_new_task	Trace1-15316	cpu_id:0,state:0,pid:16890
13589.781342500	Kernel Core	kernel/0/syscall_exit	Trace1-15316	ret:16890
13589.781482826	Kernel Core	kernel/0/page_fault_entry	Trace1-15316	write_access:0,address:0xb76df630,ip:0xb76df630,trap_id:14
13589.781499635	Kernel Core	kernel/0/page_fault_exit	Trace1-15316	res:512
13589.781513465	Kernel Core	kernel/0/syscall_entry	Trace1-15316	syscall_id:240,ip:0xb7705416
13589.781552921	Kernel Core	kernel/0/sched_schedule	Trace1-15316	prev_pid:16889,next_pid:16888,prev_state:1
13589.781772737	Kernel Core	kernel/0/sched_schedule	Trace1-15316	prev_pid:16888,next_pid:16887,prev_state:64
13589.781953709	Kernel Core	mm/0/page_free	Trace1-15316	order:1,pfn:79400
13589.782017603	Kernel Core	fd_state/0/file_descriptor	Trace1-15316	fd:0,filename:/dev/null,pid:1883
13589.782022532	Kernel Core	fd_state/0/file_descriptor	Trace1-15316	fd:1,filename:/dev/null,pid:1883

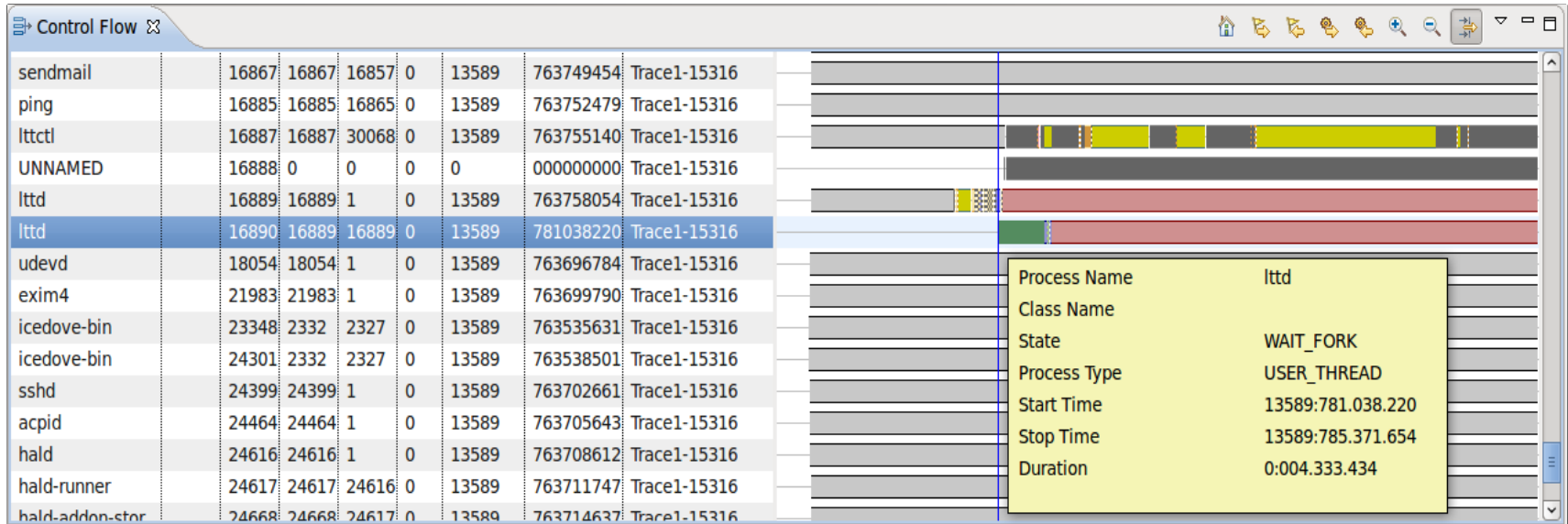
- › 'Raw' merged events in chronological order
- › Synchronized on timestamp with other views
- › Upcoming feature:
 - Event filtering on time range, event type, field value (e.g. pid), ...
 - Individual trace tabs

LTTng – Histogram View



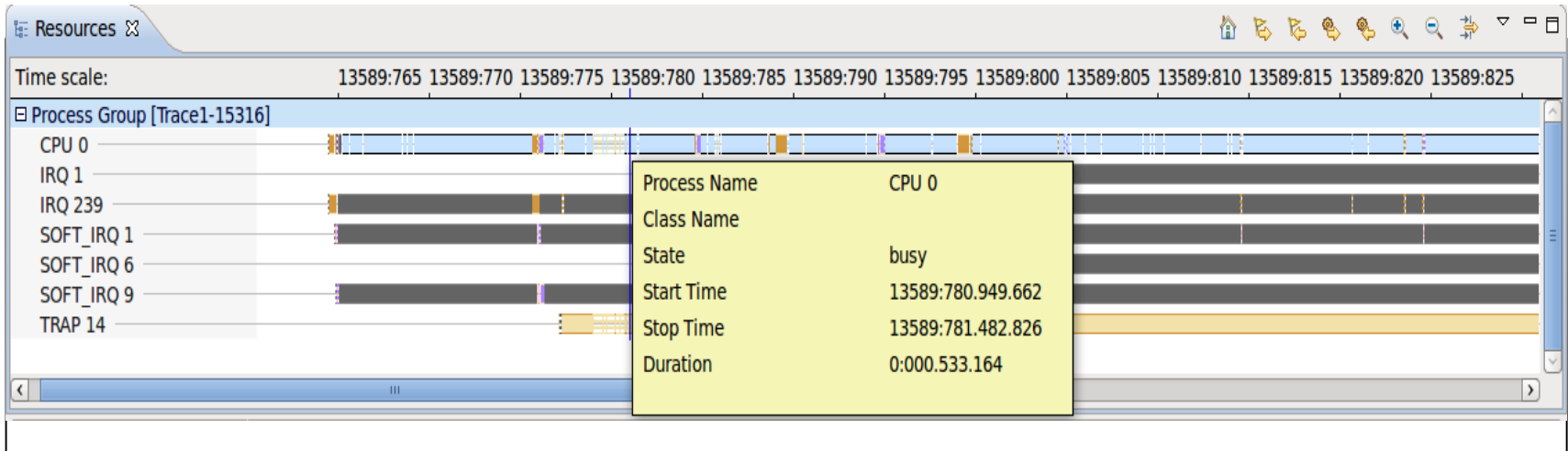
- › Event distribution over full traceset and selected window
- › Controls to modify current event and event window
- › Synchronized on current window and current event
- › Upcoming feature:
 - Zooming the selected window using the mouse

LTTng – Control Flow View



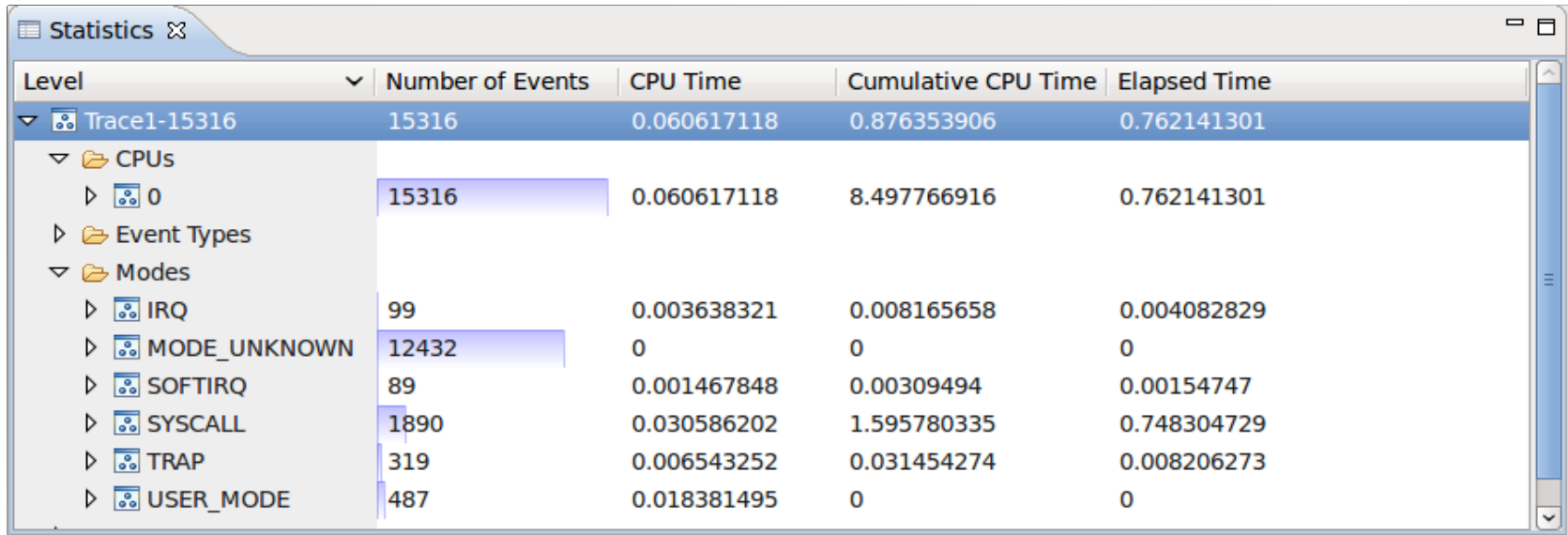
- › Displays processes states (color-coded) over time
- › State 'tooltips' through hovering
- › Zooming and filtering
- › Quick navigation between processes, states
- › Upcoming features :
 - Color legend
 - Configurable color scheme

LTTng – Resources View



- › Displays system resource states (color-coded) over time
- › State 'tooltips'
- › Zooming and filtering
- › Quick navigation between resources, states
- › Upcoming features :
 - Color legend
 - Configurable color scheme

LTTng – Statistics View



The screenshot shows a window titled "Statistics" with a tree view on the left and a table of statistics on the right. The table has five columns: Level, Number of Events, CPU Time, Cumulative CPU Time, and Elapsed Time. The tree view shows a hierarchy starting with "Trace1-15316", which is expanded to show "CPUs" (0), "Event Types", and "Modes" (IRQ, MODE_UNKNOWN, SOFTIRQ, SYSCALL, TRAP, USER_MODE). The table data is as follows:

Level	Number of Events	CPU Time	Cumulative CPU Time	Elapsed Time
Trace1-15316	15316	0.060617118	0.876353906	0.762141301
CPUs				
0	15316	0.060617118	8.497766916	0.762141301
Event Types				
Modes				
IRQ	99	0.003638321	0.008165658	0.004082829
MODE_UNKNOWN	12432	0	0	0
SOFTIRQ	89	0.001467848	0.00309494	0.00154747
SYSCALL	1890	0.030586202	1.595780335	0.748304729
TRAP	319	0.006543252	0.031454274	0.008206273
USER_MODE	487	0.018381495	0	0

- › Displays basic CPU usage statistics
- › Upcoming feature:
 - Make the view generic (decoupled from the kernel events structure)

LTTng – Upcoming Features

> General

- Tracing tool control
- Trace streaming
- Correlation of heterogeneous traces
- User Space Tracing
- Source lookup
- Performance tuning

> Analyses

- Time correction (traces synchronization)
 - > Multi-core, multi-level, multi-node
- Timing dependencies (processes interactions e.g. startup time)
- Pattern matching (security e.g. intrusion detection)

Integration with GDB Tracepoints

- › GDB Tracepoints are currently visualized through the debugger views
- › Current work to use TMF/LTTng views with GDB Tracepoints
 - Histogram view
 - Events view
- › Other discussed visualizations such as :
 - Variable variation over time

LTTng – Pointers

LTTng Eclipse Project (<http://www.eclipse.org/linuxtools/projectPages/lttng>)

LTTng Eclipse Wiki (http://wiki.eclipse.org/Linux_Tools_Project/LTTng)

Linux Tools (<http://www.eclipse.org/linuxtools/index.php>)

Update Site (<http://download.eclipse.org/technology/linuxtools/update>)

LTTng Project (<http://lttng.org>)

Tracing Wiki (<http://lttng.org/tracingwiki/index.php/TracingBook>)

Getting Eclipse to work for you in under a minute

1. Downloading Eclipse Linux Package:

- <http://eclipse.org/downloads>
- Choose: “Eclipse IDE for C/C++ Linux Developers”

2. Extract it: `tar xf <packageFile>`

3. Run it: `cd <packageDir> ; ./eclipse`

4. Create a (dummy) C/C++ project: “Hello World” is fine

5. Start debugging... GDB... GCC... etc...

Questions?



ERICSSON