A decorative graphic on the left side of the slide, composed of numerous colored dots in shades of green, blue, and purple, arranged in a roughly circular pattern.

EFFICIENT C/C++ TRACING WITH ECLIPSE

DOMINIQUE <DOT> TOUPIN <AT> ERICSSON <DOT> COM

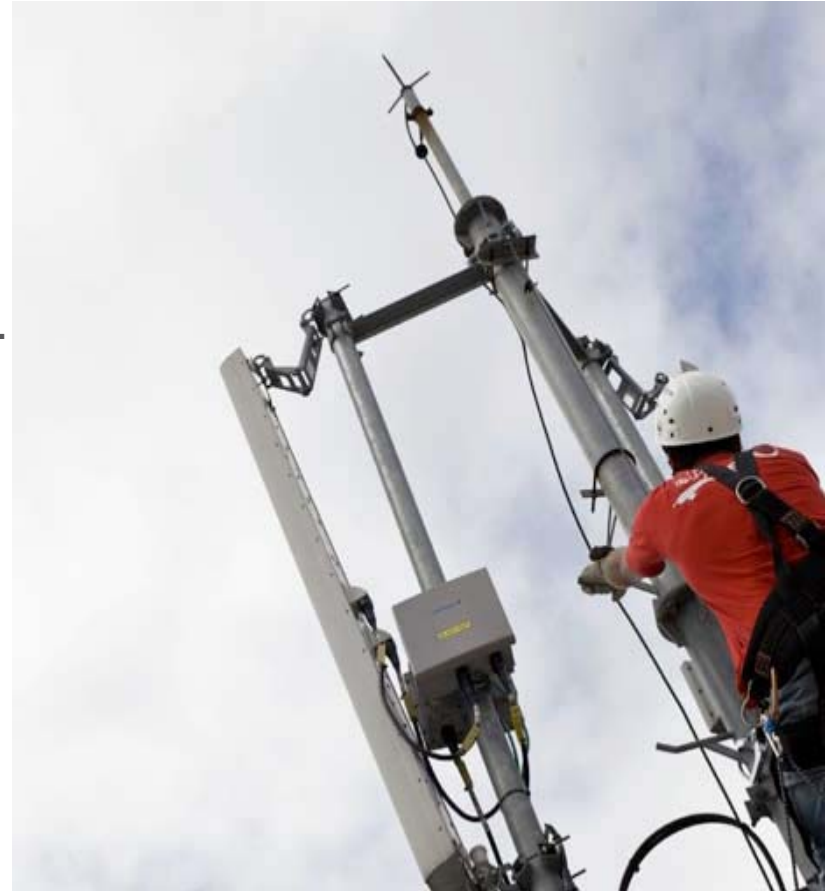
ABOUT ME

- › Developer Tool Manager at Ericsson, helping Ericsson sites to develop better software efficiently
- › Telecommunication systems
 - Open, standards-based common platform
 - High availability, 99.999 %
 - Broad range of support for both infrastructure and value-added applications
 - Multimedia, network and application processing capabilities



ABOUT ME

- › GDB improvements, non-stop, multi-process, global breakpoint, dynamic tracepoint, core awareness, OS awareness, PTC set, ...
- › LTTng improvements, user space, common trace format, multicore association, CEWG, ...
- › Eclipse GDB and LTTng integration, analysis
- › Linux tracing research project with Ecole Polytechnique
- › Linux Tracing Summit Organization
- › CDT Summit Organization



TRACING CONCEPTS

Static Tracepoints (e.g. linux kernel trace event, LTTng UST)

- Inserted at **compile** time, enable/disable at run-time
- **Low** overhead
- Represent the wisdom of developers who are most familiar with the code
- Rest of the world can use them to extract a great deal of useful information without having to know the code

Dynamic Tracepoint (e.g. Linux kernel kprobes, GDB tracepoint)

- Inserted at **run-time**, enable/disable at run-time
- **Big** overhead compared to static tracepoint
- Use it when a static tracepoint is missing
- If used a lot over a long period of time, think about converting them to static

TRACING USEFULNESS

- Very efficient compared to logging
- Performance tuning
- Diagnose multicore programming bug
- Monitoring a live system



LTTNG



- › Linux Tracing Toolkit next generation (LTTng)
- › Included in several [Linux Distros](#), more than 90 contributors from 20 different organizations
- › System-wide tracing across:
 - Kernel
 - Hypervisor
 - VM
 - Library
 - Application
- › **Precise** and **fast** clock sources with near cycle-level timestamps, i.e. 1-100 ns





LTTNG **DISTINCTIVE** FEATURES

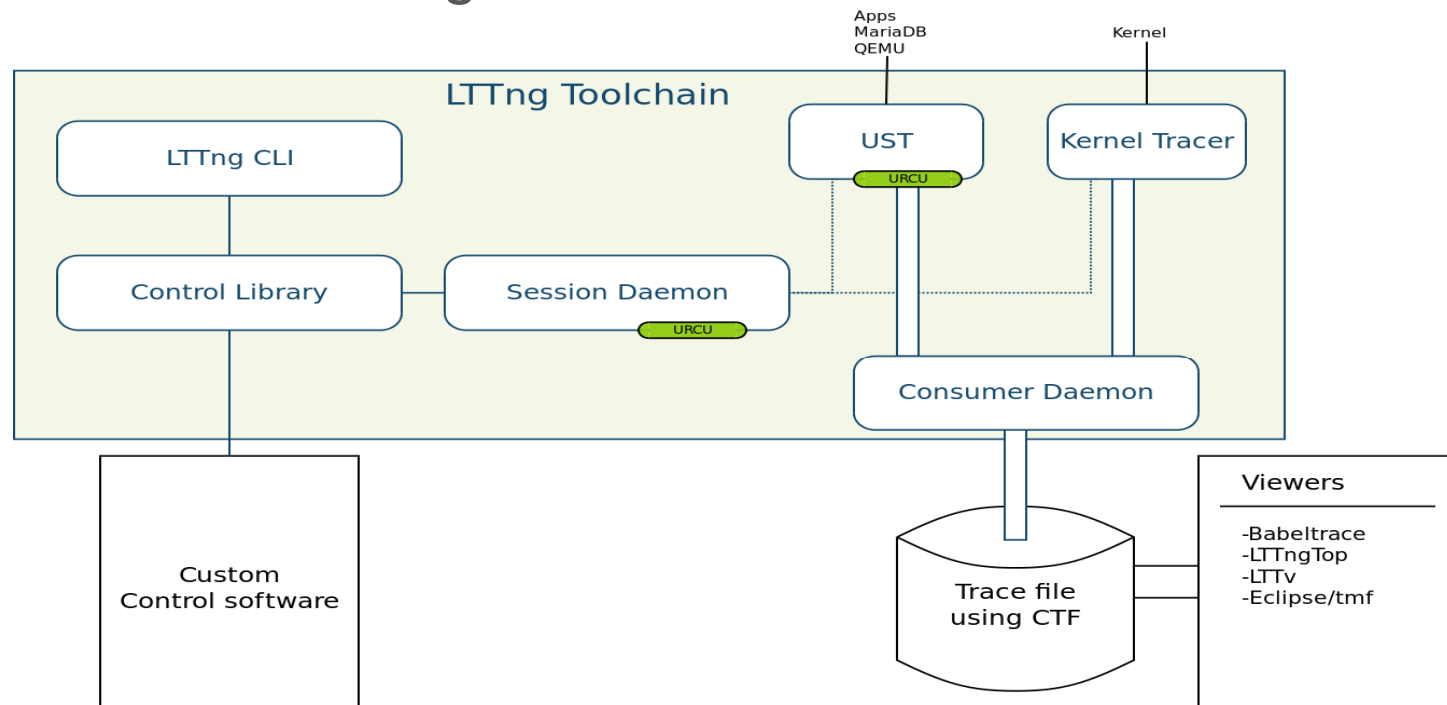
- › Efficient trace data transport
 - Compact Binary format with Common Trace Format
 - Per-core buffers ensures scalability
 - Fast-paths in caller context, amortized synchronization
 - Zero-copy using splice and mmap system calls, over disk, network or consumed in-place

- › Low-intrusiveness instrumentation and control
 - Per-process control thread with low footprint
 - Fast user-space process registration
 - Static Tracepoints enabled with static jump patching



LTTNG **DISTINCTIVE** FEATURES

- › Multi-session support with per host or per user daemon
- › Algorithms based on RCU verified by model checking
- › Designed to meet real-time constraints
- › Supports live streaming of the trace data



COMMON TRACE FORMAT

- [Ericsson](#) and [Linux Foundation CE Linux Workgroup](#)
- Reviewed by Linux kernel developers and SystemTAP communities
- In collaboration with [Multi-Core Association Tool Infrastructure Workgroup](#)



Freescale, Mentor Graphics, IBM, IMEC, National Instruments, Nokia Siemens Networks, Samsung, Texas Instruments, Tiler, Wind River, University of Houston, Polytechnique Montréal, University of Utah, ...



- Requirement, specification, reference implementation <http://www.efficios.com/ctf>



COMMON TRACE FORMAT

- Self describing
- Very compact binary trace format
- System-wide and multi-system trace representation in a common format, for integrated analysis:
 - Software traces
 - Across multiple CPUs
 - Across the software stack, e.g. hypervisor, kernel, library, applications
 - Hardware traces
 - DSPs, device-specific tracing components.
 - GPUs.

ECLIPSE TRACING MONITORING FRAMEWORK (TMF)



- › Eclipse Linux Tools Project
- › Framework to build trace visualization and analysis tool
- › Scalability allows to handle traces exceeding memory
- › Enable trace analysis from different sources
- › LTTng Eclipse integration is an implementation on top of TMF

The Linux Tools project aims to bring a **full-featured C and C++ IDE** to Linux developers. We build on the source editing and debugging features of the **CDT** and integrate popular native development tools such as the GNU Autotools, Valgrind, OProfile, RPM, SystemTap, GCov, GProf, LTTng, etc. Current projects include LTTng trace viewers and analyzers, an RPM .spec editor, Autotools build integration, a Valgrind heap usage analysis tool, and OProfile call profiling tools.



Project Expl

- MyCppProject
- MyGdbTraceProject
- MyJavaProject
- MyLTngProject
 - Experiments
 - MyFirstExp [2]
 - MySecondExp [3]
 - trace_116MB
 - trace_2GB
 - trace_4MB
 - Traces [5]
 - trace_116MB
 - trace_2GB
 - trace_4MB
 - Trace-2.5-15316
 - Trace-2.5-15471
- MyUstProject

Control Flow Resources Statistics

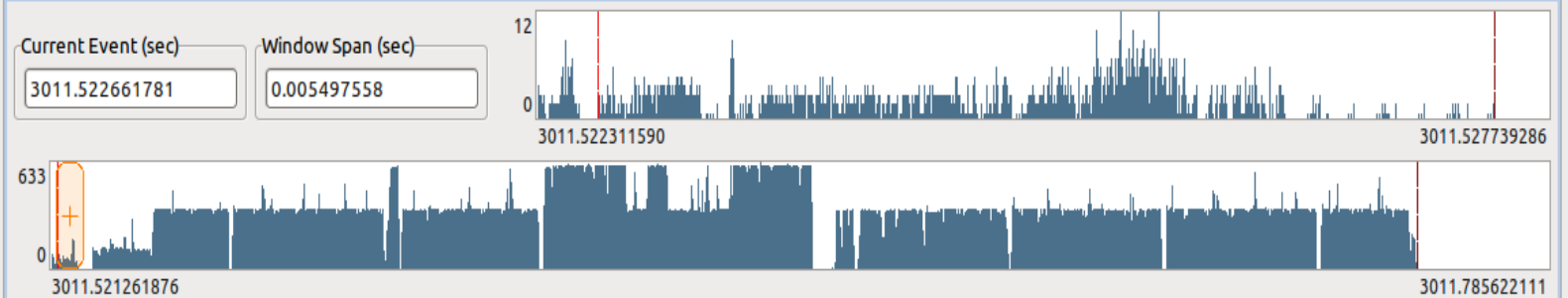
Process	Brand	PID	TGID	PPID	CPU	Birth sec	Birth nsec	TRACE	Legend			
UNNAMED		0	0	0	1	0	000000000	trace_4MB	3011:523	3011:524	3011:525	3011:526
UNNAMED		9	0	0	0	0	000000000	trace_4MB				
UNNAMED		2297	0	0	0	0	000000000	trace_4MB				
UNNAMED		2347	0	0	1	0	000000000	trace_4MB				
UNNAMED		12920	0	0	0	0	000000000	trace_4MB				
UNNAMED		12931	0	0	1	0	000000000	trace_4MB				
/bin/ping		12932	12932	12920	0	3011	522661781	trace_4MB				

Events - trace_4MB

Timestamp	Trace	Marker	Content
<srch>	<srch>	<srch>	<srch>
3011.522545898	trace_4MB	mm/1/page_free	pfn:9544,order:1
3011.522550522	trace_4MB	mm/1/add_to_page_cache	sdev:21,inode:2097156
3011.522552078	trace_4MB	kernel/0/sched_migrate_task	dest_cpu:0,state:256,pid:12920
3011.522661781	trace_4MB	kernel/0/process_fork	child_pid:12932,child_tgid:12932,parent_pid:12920
3011.522665277	trace_4MB	kernel/0/sched_migrate_task	dest_cpu:0,state:256,pid:12932
3011.522669142	trace_4MB	kernel/0/sched_wakeup_new_task	cpu_id:0,state:0,pid:12932
3011.522671202	trace_4MB	kernel/0/syscall_exit	sch:12932

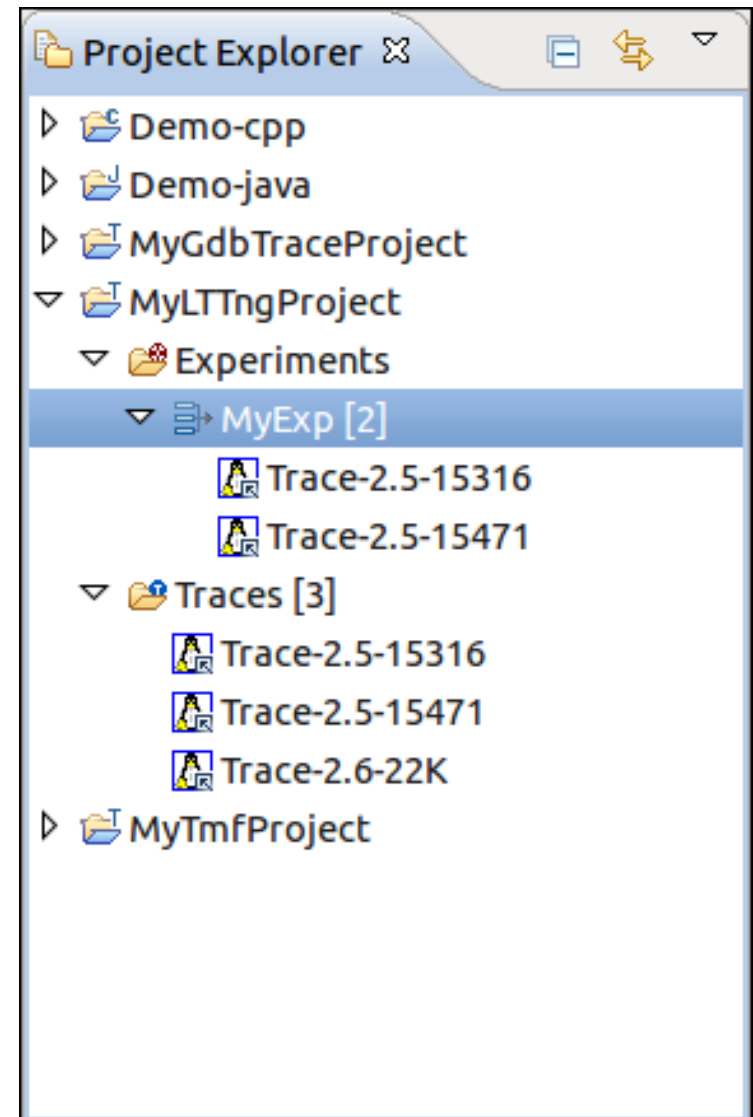
Histogram

Properties



PROJECT EXPLORER VIEW

- › **Tracing projects** integrate into Project Explorer view using Common Navigator Framework (CNF)
- › Tracing projects are used to group traces you are interested in
- › **Traces** folder contains the traces of interest
- › **Experiments** are the way to group and correlate traces

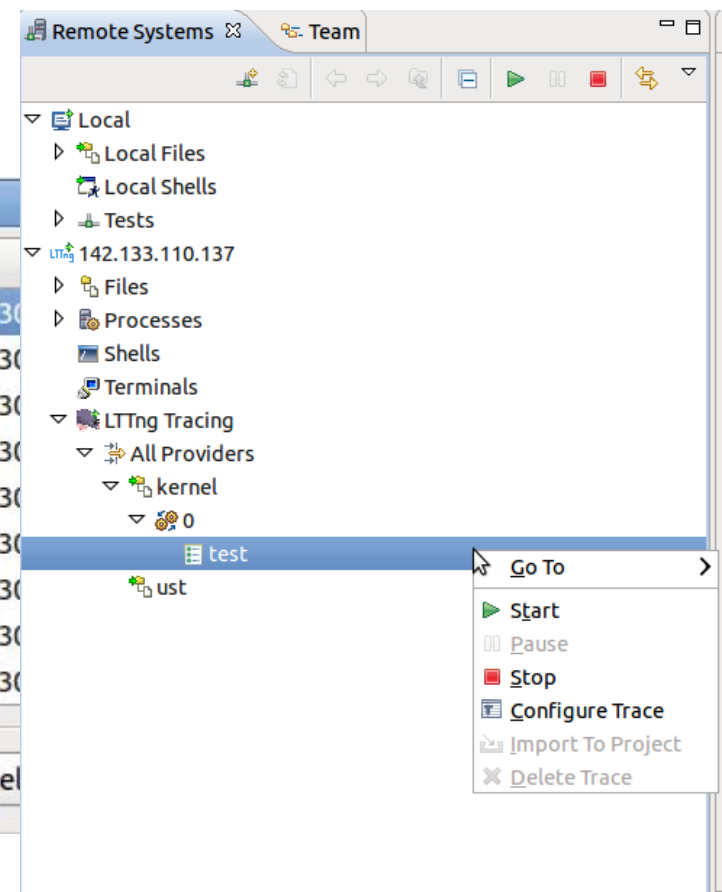


KERNEL TRACE CONTROL

- › Configuration, control and collection of traces for analysis
- › Streaming of traces

Configure markers					
	Name	Location	Format	Event_id	Call
<input checked="" type="checkbox"/>	rcu/tree_callb	rcu_trace	"func %p"	2	0xc01b1f30
<input checked="" type="checkbox"/>	rcu/tree_call_	rcu_trace	"func %p ip 0x%lX"	1	0xc01b1f30
<input checked="" type="checkbox"/>	rcu/tree_call_	rcu_trace	"func %p ip 0x%lX"	0	0xc01b1f30
<input checked="" type="checkbox"/>	userspace/ev	ltt_userspace_	"string %s"	0	0xc01b1f30
<input checked="" type="checkbox"/>	kprobe_state,	ltt_kprobes	"ip 0x%lX symbol %s"	0	0xc01b1f30
<input checked="" type="checkbox"/>	block/rq_abo	block_trace	"data_len %u rw %d erro 9	0xc01b1f30	
<input checked="" type="checkbox"/>	block/rq_abo	block_trace	"hard_sector %llu rw %d 18	0xc01b1f30	
<input checked="" type="checkbox"/>	block/rq_inse	block_trace	"data_len %u rw %d erro 23	0xc01b1f30	
<input checked="" type="checkbox"/>	block/rq_inse	block_trace	"hard_sector %llu rw %d 19	0xc01b1f30	

Select All Deselect All Cancel



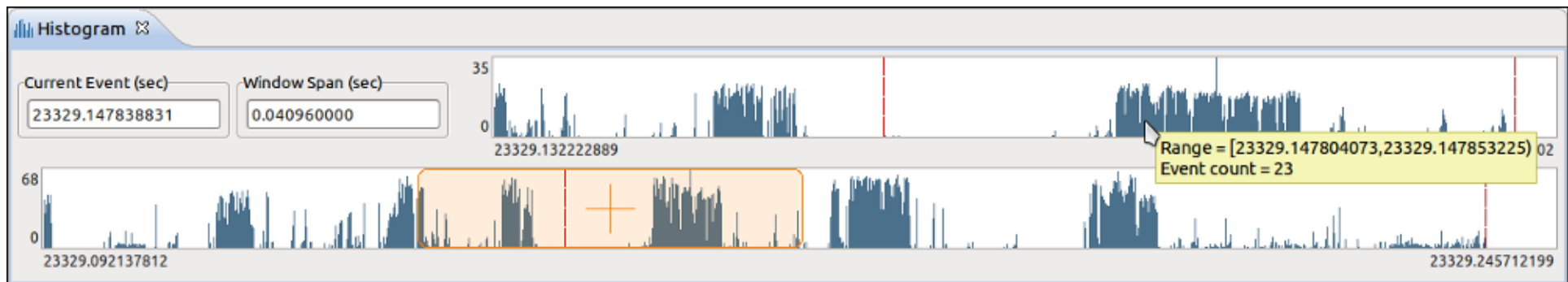
EVENTS VIEW

- › ‘Raw’ merged events in chronological order
- › Supports following functionalities
 - Searching
 - Filtering
 - Highlighting

Events - Trace1-15316				
Timestamp	Source	Type	Reference	Content
<srch>	<srch>	<srch>	<srch>	. *pid:244.*
13589.786284802	Kernel Core	fd_state/0/file_descriptor	Trace1-15316	filename:/dev/null,pid:24464,fd:0
13589.786286742	Kernel Core	fd_state/0/file_descriptor	Trace1-15316	filename:/dev/null,pid:24464,fd:1
13589.786288632	Kernel Core	fd_state/0/file_descriptor	Trace1-15316	filename:/dev/null,pid:24464,fd:2
13589.786294704	Kernel Core	fd_state/0/file_descriptor	Trace1-15316	filename:/proc/acpi/event,pid:24464,fd:3
13589.786297300	Kernel Core	fd_state/0/file_descriptor	Trace1-15316	filename:socket:[120515],pid:24464,fd:4
13589.786300104	Kernel Core	fd_state/0/file_descriptor	Trace1-15316	filename:socket:[120543],pid:24464,fd:5
13589.786302451	Kernel Core	fd_state/0/file_descriptor	Trace1-15316	filename:socket:[120517],pid:24464,fd:6
13589.786305224	Kernel Core	fd_state/0/file_descriptor	Trace1-15316	filename:socket:[121534],pid:24464,fd:7
13589.786311513	Kernel Core	fd_state/0/file_descriptor	Trace1-15316	filename:/dev/null,pid:24616,fd:0

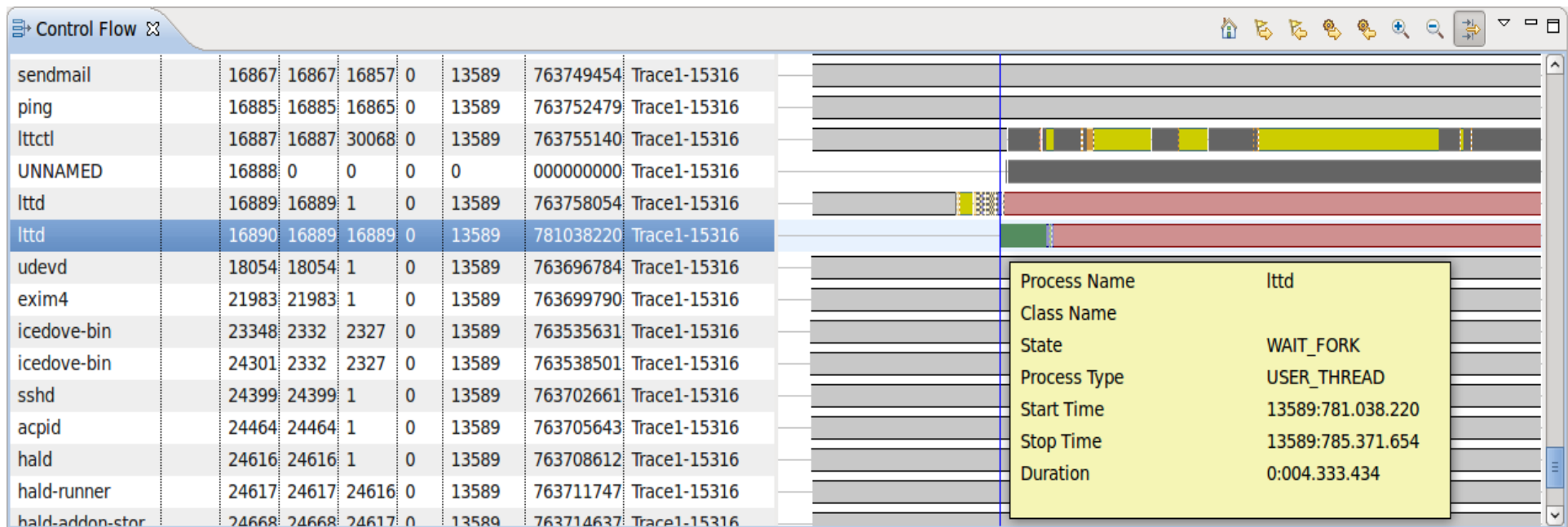
HISTOGRAM VIEW

- › Event distribution over full traceset and selected window
- › Controls to modify current event and event window
- › Zoom in/out using mouse scroll



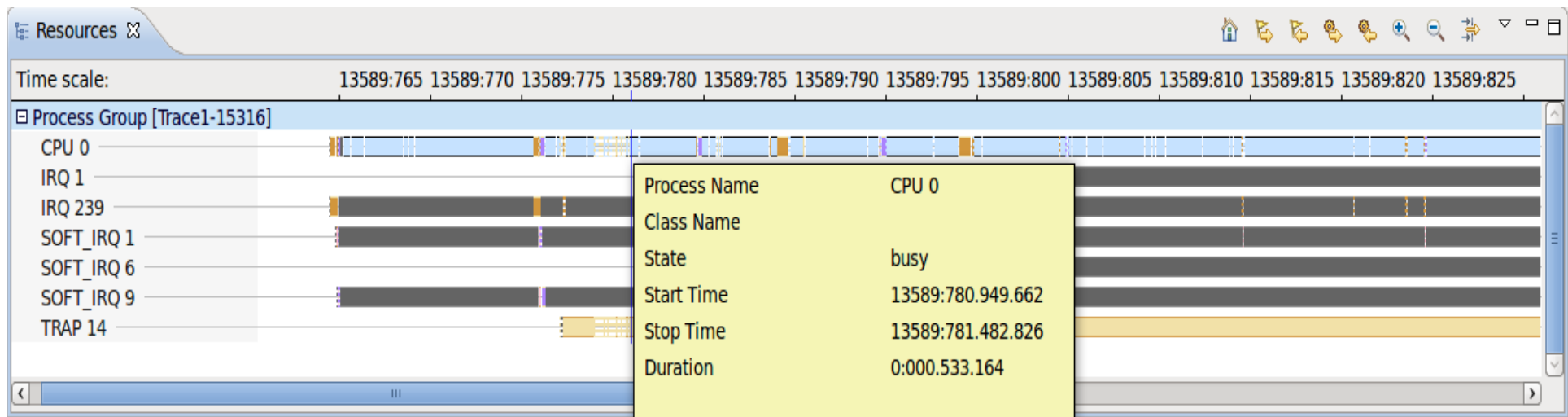
CONTROL FLOW VIEW

- › Displays **processes** **state changes** (color-coded) over time
- › State 'tooltips'
- › Zooming and filtering
- › Quick navigation between processes, states



RESOURCE VIEW

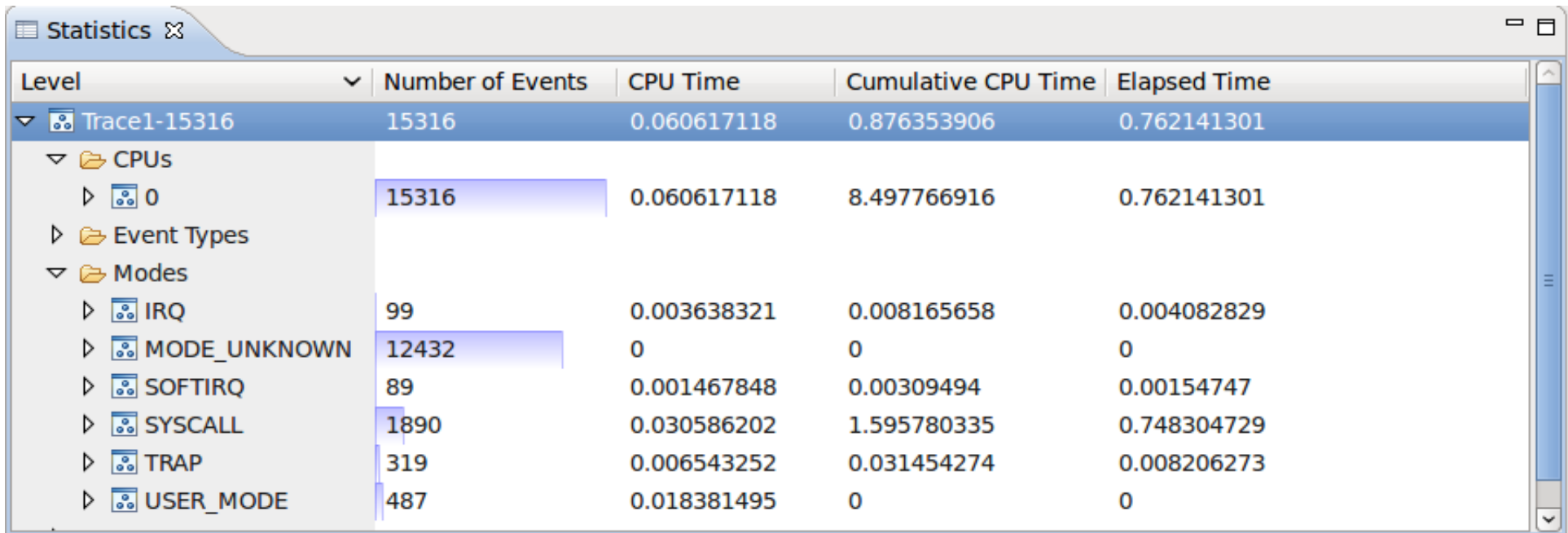
- › Displays **system resource states** (color-coded) over time
- › State 'tooltips'
- › Zooming and filtering
- › Quick navigation between resources, states



STATISTICS VIEW

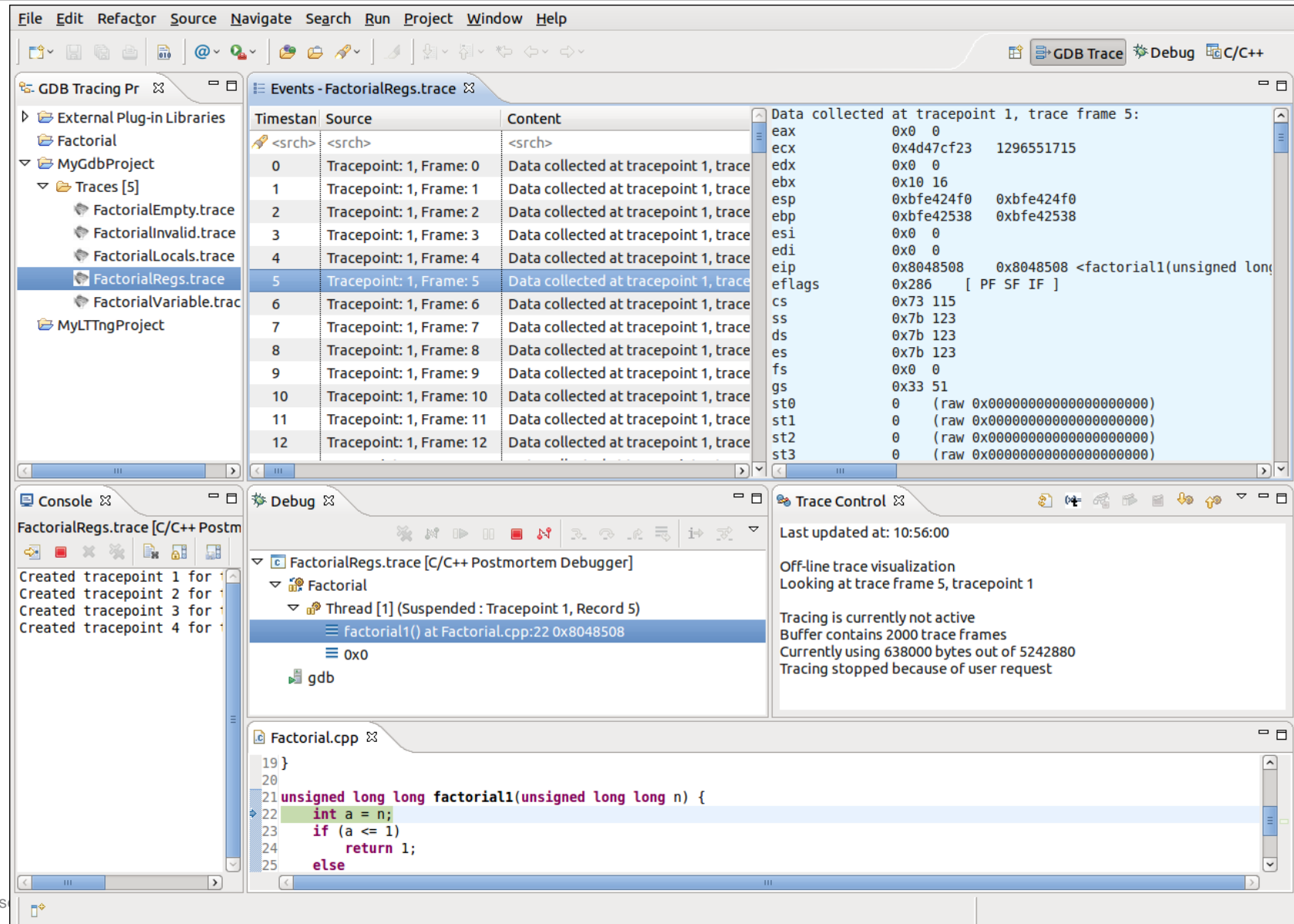
› Displays **basic trace statistics**

- Per process, event type
- Per CPU, mode



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

GDB TRACEPOINT INTEGRATION



The screenshot displays the Eclipse IDE interface with the following components:

- Menu Bar:** File, Edit, Refactor, Source, Navigate, Search, Run, Project, Window, Help.
- Toolbar:** Includes icons for file operations, search, and debugging.
- GDB Tracing Pr Panel:**
 - External Plug-in Libraries: Factorial
 - MyGdbProject:
 - Traces [5]:
 - FactorialEmpty.trace
 - FactorialInvalid.trace
 - FactorialLocals.trace
 - FactorialRegs.trace** (selected)
 - FactorialVariable.trace
 - MyLTngProject
- Events - FactorialRegs.trace Panel:**

Timestamp	Source	Content
<srch>	<srch>	<srch>
0	Tracepoint: 1, Frame: 0	Data collected at tracepoint 1, trace
1	Tracepoint: 1, Frame: 1	Data collected at tracepoint 1, trace
2	Tracepoint: 1, Frame: 2	Data collected at tracepoint 1, trace
3	Tracepoint: 1, Frame: 3	Data collected at tracepoint 1, trace
4	Tracepoint: 1, Frame: 4	Data collected at tracepoint 1, trace
5	Tracepoint: 1, Frame: 5	Data collected at tracepoint 1, trace
6	Tracepoint: 1, Frame: 6	Data collected at tracepoint 1, trace
7	Tracepoint: 1, Frame: 7	Data collected at tracepoint 1, trace
8	Tracepoint: 1, Frame: 8	Data collected at tracepoint 1, trace
9	Tracepoint: 1, Frame: 9	Data collected at tracepoint 1, trace
10	Tracepoint: 1, Frame: 10	Data collected at tracepoint 1, trace
11	Tracepoint: 1, Frame: 11	Data collected at tracepoint 1, trace
12	Tracepoint: 1, Frame: 12	Data collected at tracepoint 1, trace
- Console Panel:**

```

Created tracepoint 1 for
Created tracepoint 2 for
Created tracepoint 3 for
Created tracepoint 4 for

```
- Debug Panel:**
 - FactorialRegs.trace [C/C++ Postmortem Debugger]
 - Factorial
 - Thread [1] (Suspended : Tracepoint 1, Record 5)
 - factorial1() at Factorial.cpp:22 0x8048508
 - 0x0
 - gdb
- Trace Control Panel:**
 - Last updated at: 10:56:00
 - Off-line trace visualization
 - Looking at trace frame 5, tracepoint 1
 - Tracing is currently not active
 - Buffer contains 2000 trace frames
 - Currently using 638000 bytes out of 5242880
 - Tracing stopped because of user request
- Factorial.cpp Panel:**

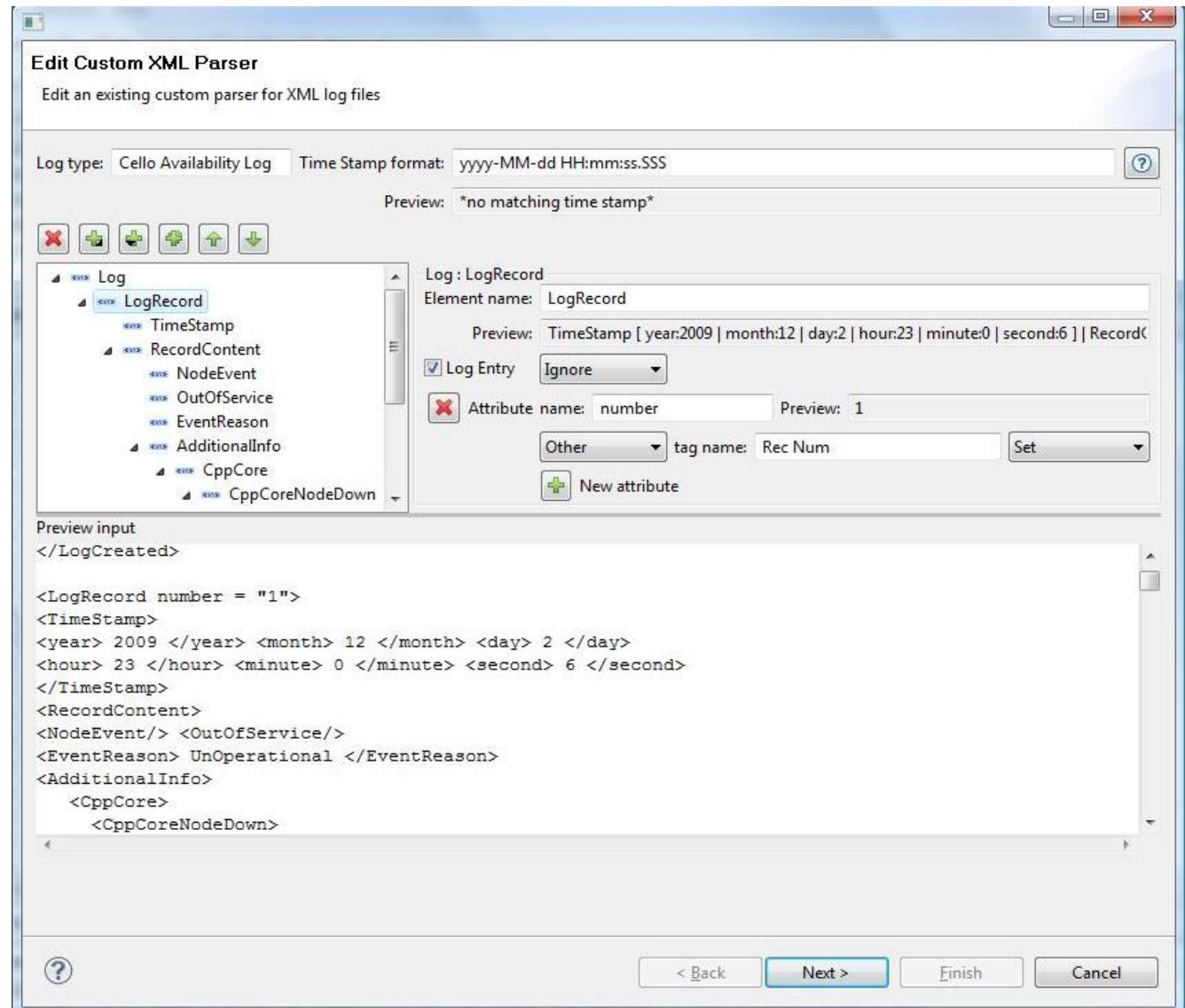
```

19 }
20
21 unsigned long long factorial1(unsigned long long n) {
22     int a = n;
23     if (a <= 1)
24         return 1;
25     else

```

TEXT PARSER WIZARD

- › Supports **text** and **XML** traces
- › Customizable by the user via a wizard
- › **Regular expressions** used for parsing text traces
- › Parsed traces integrate with all TMF views



Edit Custom XML Parser
Edit an existing custom parser for XML log files

Log type: Cello Availability Log Time Stamp format: yyyy-MM-dd HH:mm:ss.SSS
Preview: *no matching time stamp*

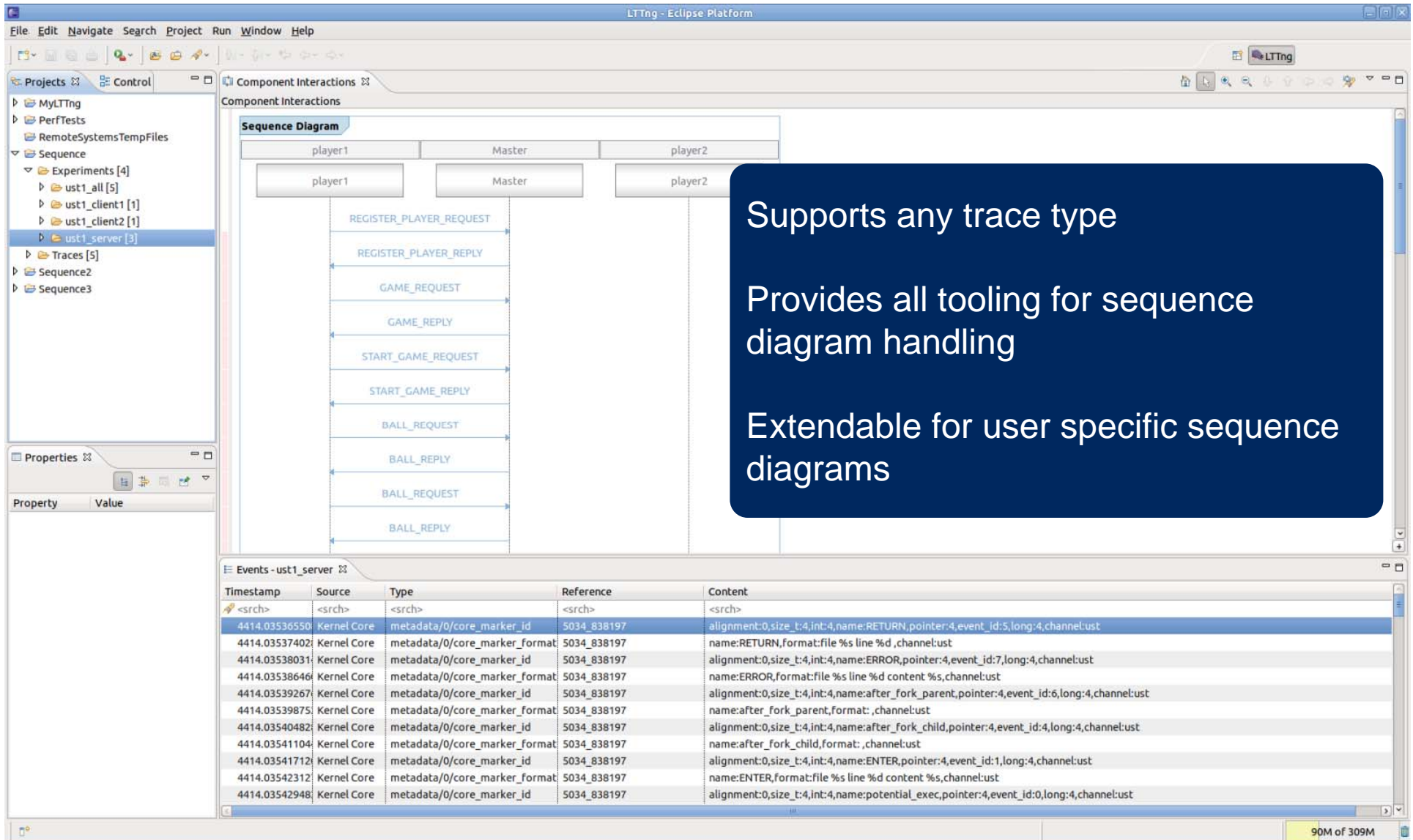
Log : LogRecord
Element name: LogRecord
Preview: TimeStamp [year:2009 | month:12 | day:2 | hour:23 | minute:0 | second:6] | Record<
☒ Log Entry Ignore
☒ Attribute name: number Preview: 1
Other tag name: Rec Num Set
New attribute

Preview input
</LogCreated>

<LogRecord number = "1">
<TimeStamp>
<year> 2009 </year> <month> 12 </month> <day> 2 </day>
<hour> 23 </hour> <minute> 0 </minute> <second> 6 </second>
</TimeStamp>
<RecordContent>
<NodeEvent/> <OutOfService/>
<EventReason> UnOperational </EventReason>
<AdditionalInfo>
<CppCore>
<CppCoreNodeDown>

< Back Next > Finish Cancel

SEQUENCE DIAGRAM FRAMEWORK



The screenshot displays the LTTng Eclipse Platform interface. The main window shows a sequence diagram with three participants: player1, Master, and player2. The diagram includes messages such as REGISTER_PLAYER_REQUEST, REGISTER_PLAYER_REPLY, GAME_REQUEST, GAME_REPLY, START_GAME_REQUEST, START_GAME_REPLY, BALL_REQUEST, BALL_REPLY, and another BALL_REQUEST. A blue callout box on the right lists the following features:

- Supports any trace type
- Provides all tooling for sequence diagram handling
- Extendable for user specific sequence diagrams

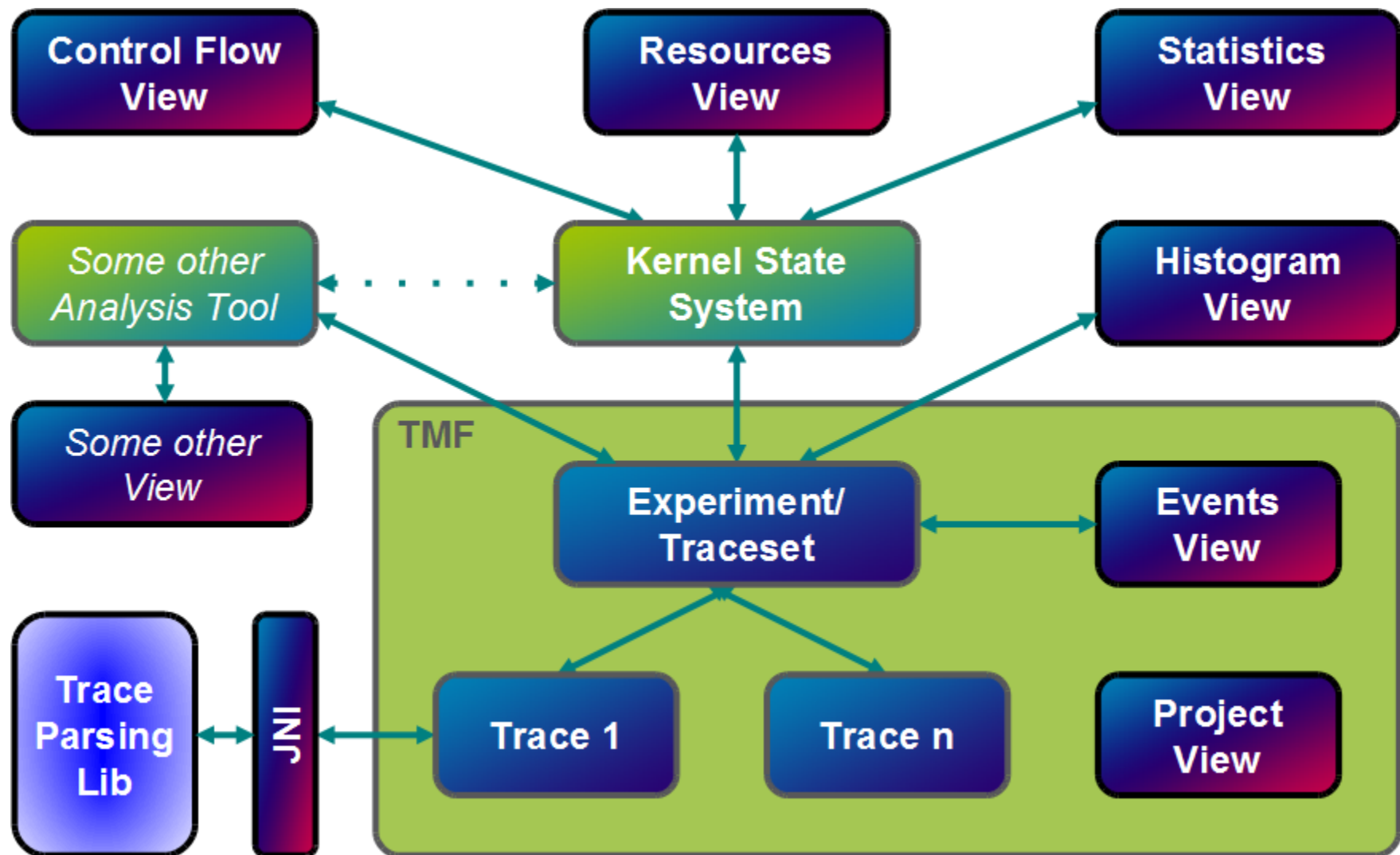
At the bottom, the 'Events - ust1_server' panel shows a table of events:

Timestamp	Source	Type	Reference	Content
<srch>	<srch>	<srch>	<srch>	<srch>
4414.03536550	Kernel Core	metadata/0/core_marker_id	5034_838197	alignment:0,size_t:4,int:4,name:RETURN,pointer:4,event_id:5,long:4,channel:ust
4414.03537402	Kernel Core	metadata/0/core_marker_format	5034_838197	name:RETURN,format:file %s line %d,channel:ust
4414.03538031	Kernel Core	metadata/0/core_marker_id	5034_838197	alignment:0,size_t:4,int:4,name:ERROR,pointer:4,event_id:7,long:4,channel:ust
4414.03538646	Kernel Core	metadata/0/core_marker_format	5034_838197	name:ERROR,format:file %s line %d content %s,channel:ust
4414.03539267	Kernel Core	metadata/0/core_marker_id	5034_838197	alignment:0,size_t:4,int:4,name:after_fork_parent,pointer:4,event_id:6,long:4,channel:ust
4414.03539875	Kernel Core	metadata/0/core_marker_format	5034_838197	name:after_fork_parent,format:,channel:ust
4414.03540482	Kernel Core	metadata/0/core_marker_id	5034_838197	alignment:0,size_t:4,int:4,name:after_fork_child,pointer:4,event_id:4,long:4,channel:ust
4414.03541104	Kernel Core	metadata/0/core_marker_format	5034_838197	name:after_fork_child,format:,channel:ust
4414.03541712	Kernel Core	metadata/0/core_marker_id	5034_838197	alignment:0,size_t:4,int:4,name:ENTER,pointer:4,event_id:1,long:4,channel:ust
4414.03542312	Kernel Core	metadata/0/core_marker_format	5034_838197	name:ENTER,format:file %s line %d content %s,channel:ust
4414.03542948	Kernel Core	metadata/0/core_marker_id	5034_838197	alignment:0,size_t:4,int:4,name:potential_exec,pointer:4,event_id:0,long:4,channel:ust

CURRENT DEVELOPMENT

- › Support for LTTng 2.0
 - CTF-based Kernel and UST traces
 - Trace control
 - Session management
 - Support for multiple trace sessions
 - Streaming
- › Juno
 - Project graduation work (Linux Tools 1.0)
 - Uplift to Eclipse 4.X
- › More analysis tools
 - Latency analysis, clock adjustment, trace comparison, etc.

ECLIPSE TRACING MONITORING FRAMEWORK ARCHITECTURE



SOME REFERENCES

› Linux Tools

- Linux Tools: <http://www.eclipse.org/linuxtools/index.php>
- Update Site: <http://download.eclipse.org/technology/linuxtools/update>

LTTng (Eclipse)

- LTTng Eclipse Project: <http://www.eclipse.org/linuxtools/projectPages/lttnng>
- LTTng Eclipse Wiki: http://wiki.eclipse.org/Linux_Tools_Project/LTTng
- LTTng User Guide: http://wiki.eclipse.org/Linux_Tools_Project/LTTng/User_Guide
- TMF User Guide: http://wiki.eclipse.org/Linux_Tools_Project/TMF/User_Guide

› LTTng project: <http://lttnng.org>

› For more info, questions, discussions:

- linuxtools-dev@eclipse.org

Q&A

