

Polytechnique – December 2017



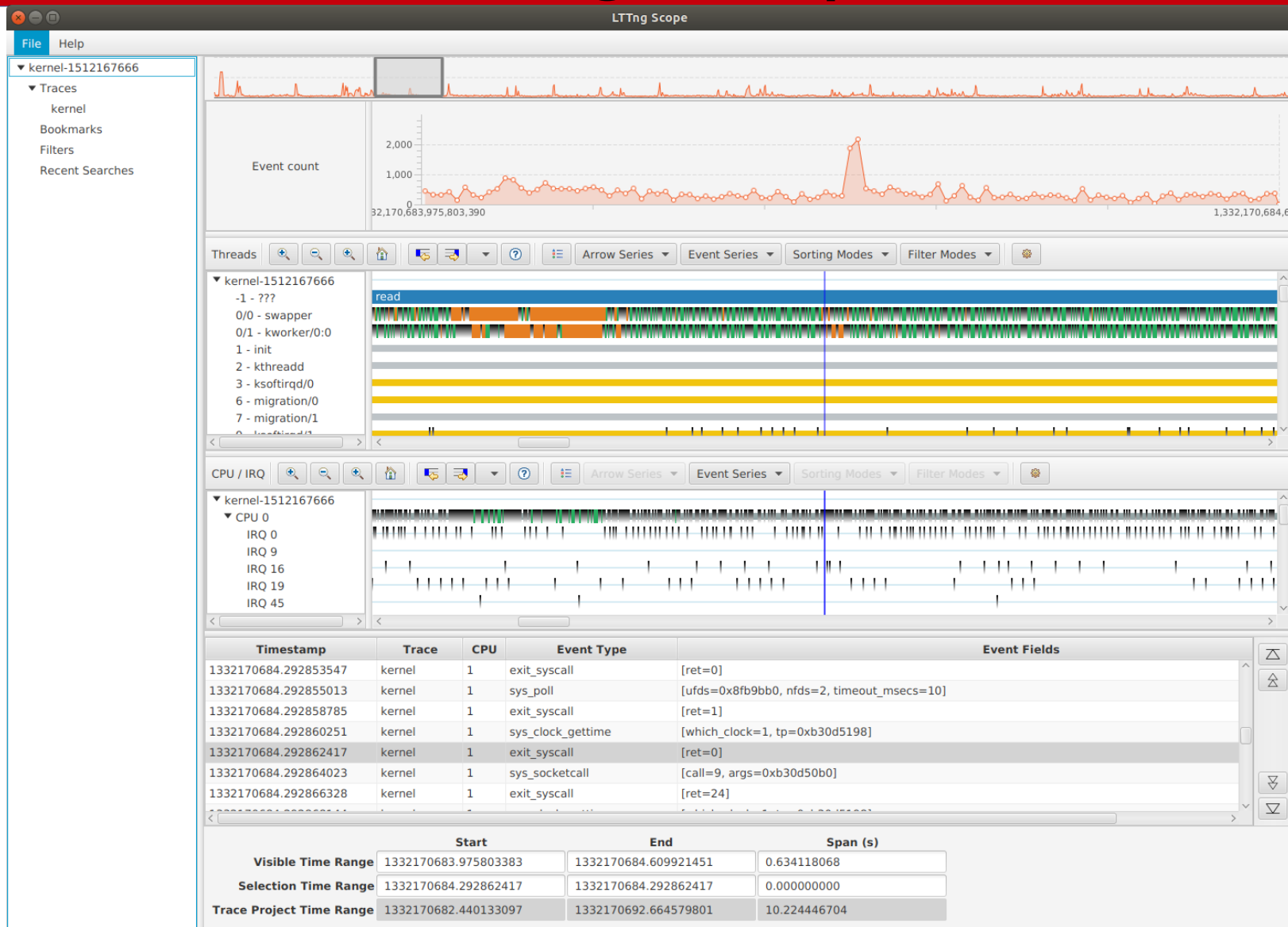
Introducing ***LTTng Scope***

*Effici***OS**

Content

- Project goals
- User experience design goals
- User interface design
- Architecture
- Current status
- Demo
- Roadmap

LTTng Scope



Project Goals

- Trace viewer for CTF / LTTng traces
- Streamline user experience
 - Easy to install
 - Discoverability of features
 - Scenario-driven user interface
- Layered architecture

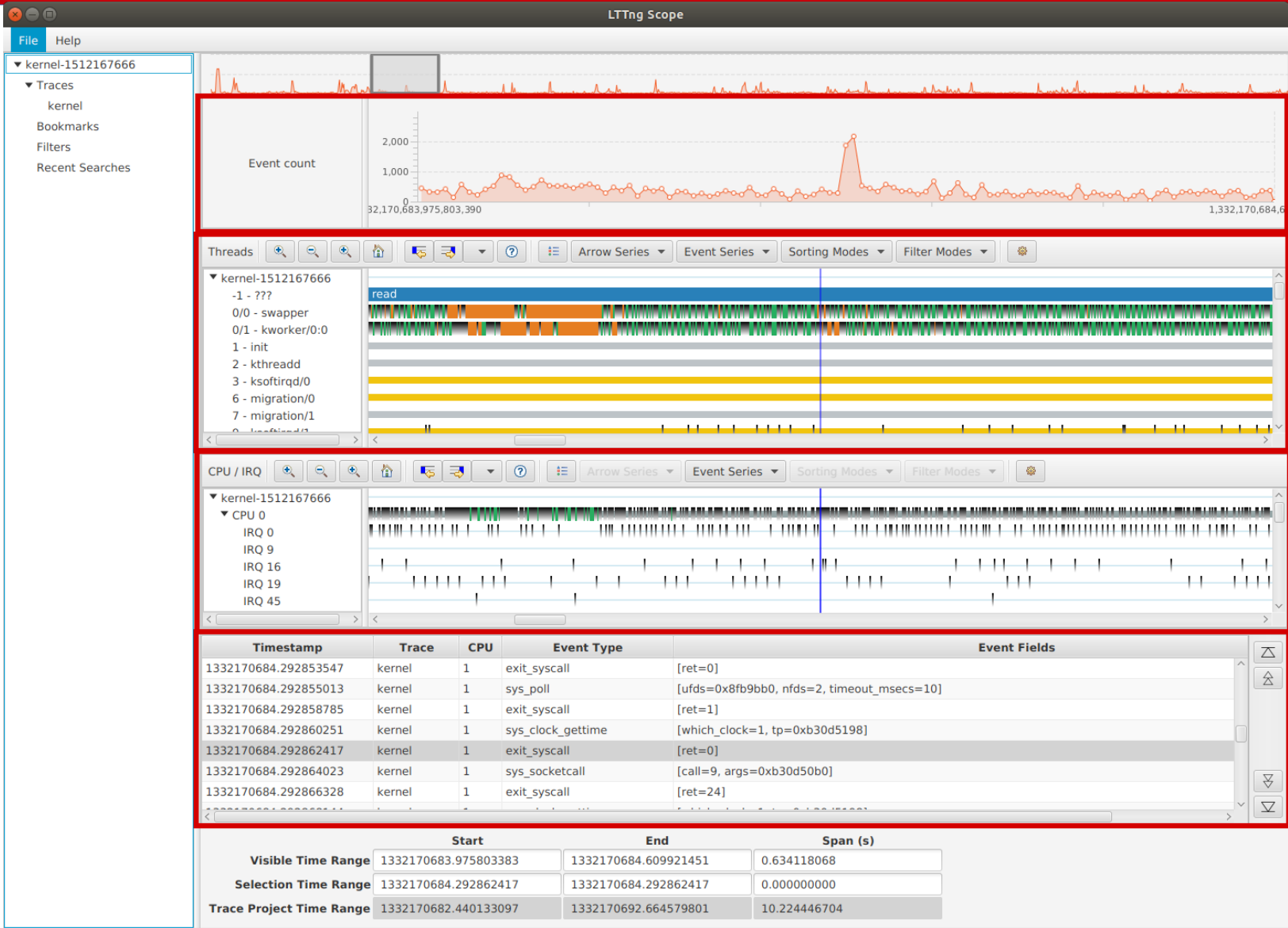
UX Design Goals

- Driven by investigation scenarios, rather than available data.
- Allow user to focus on the most important information at each step of the investigation.
- User should discover features intuitively without having to read documentation.

UI Design

- Views and analyses are exposed as available *widgets*.
- User can add/remove/reorder widgets in the view.
 - Can instantiate multiple widgets of the same type, then apply different settings.

UI Design



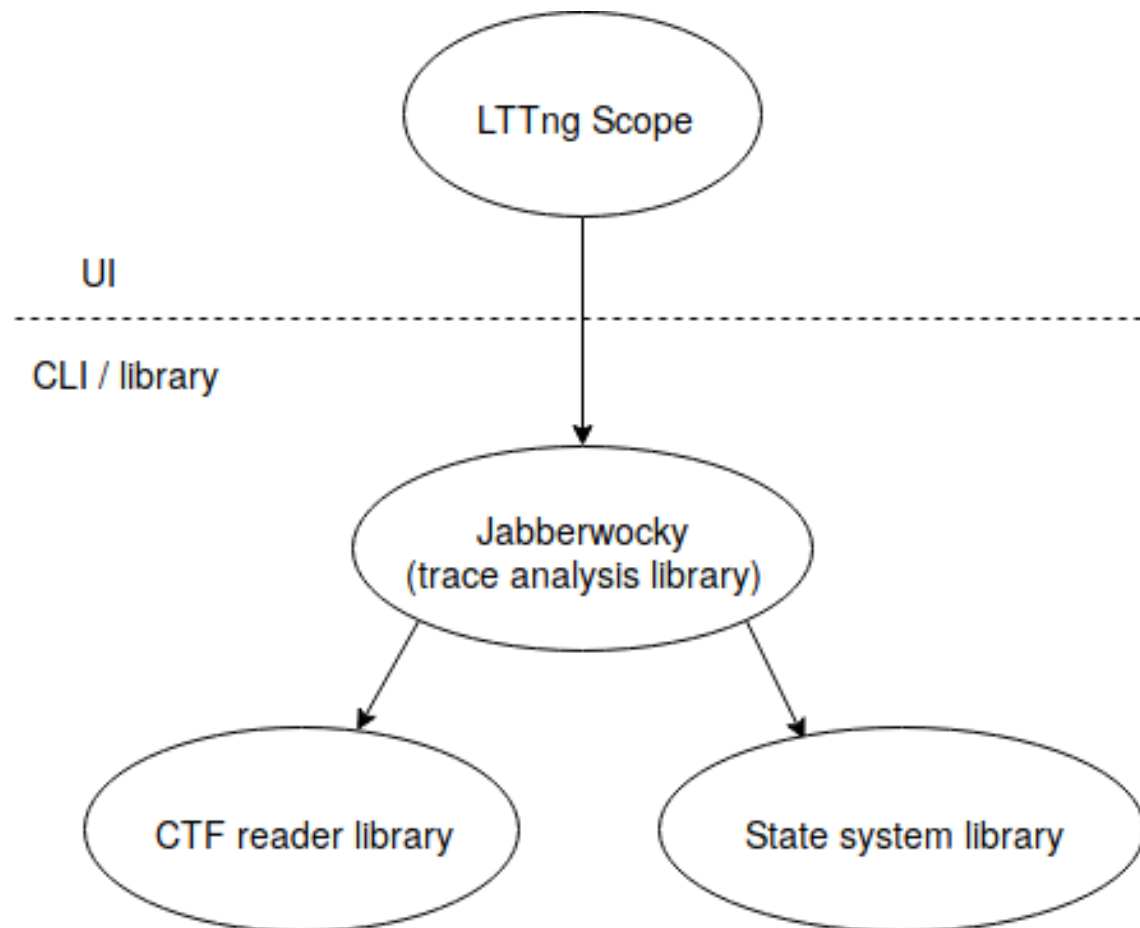
UI Design

- Widgets should work *together*, not as separate silos.
- Project-wide settings like filters, bookmarks should apply to all widgets similarly.

UI Design

- Filtering is key:
 - Tracing generates a lot of data,
 - The tool should reduce noise,
 - Allow the user to focus on interesting data.

Layered Architecture



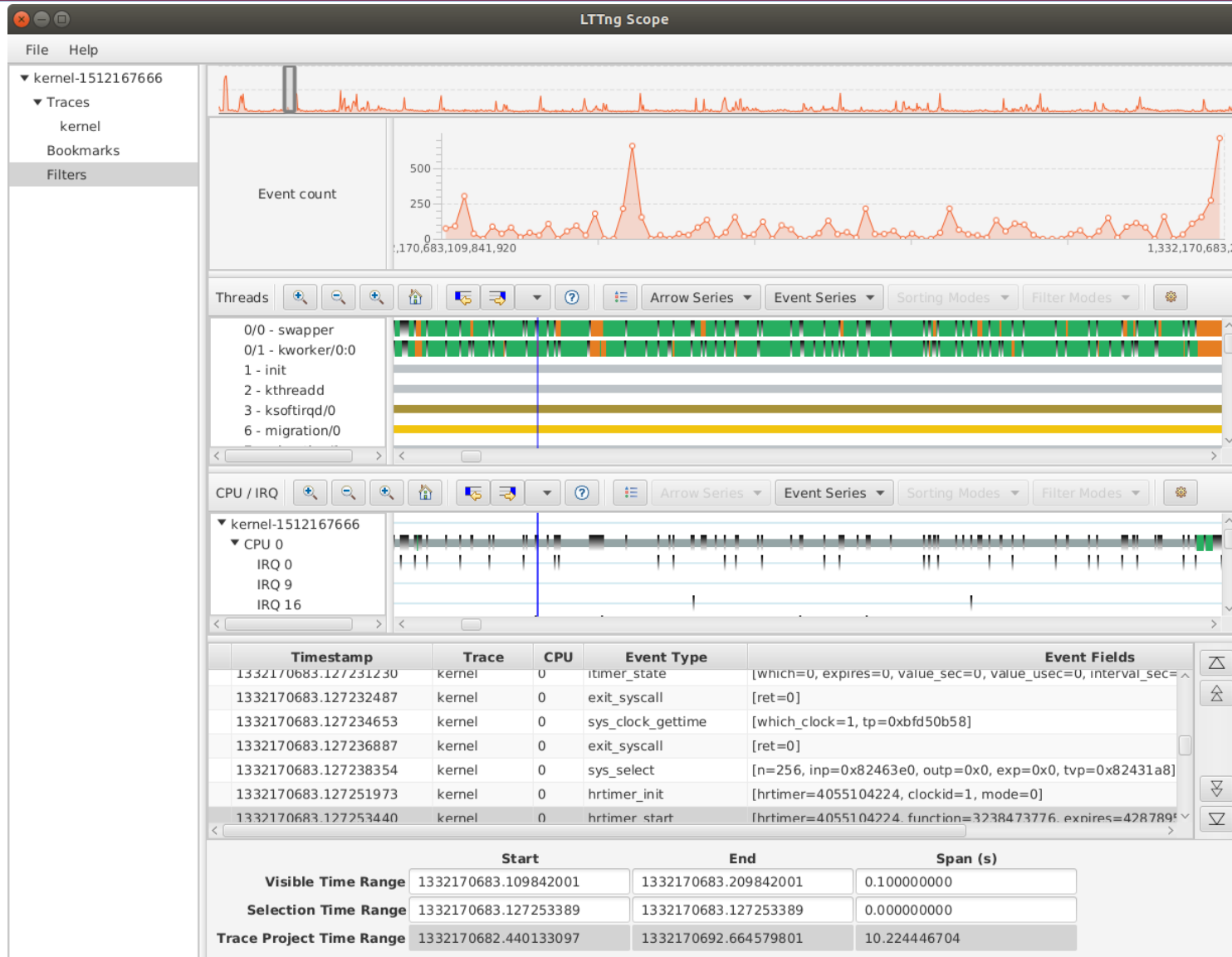
Current Status

- Standalone application based on JavaFX.
- ~6 MB package!
 - Including the client and library
- Implemented features
 - Event table (multiple instances)
 - Event count histograms (XY-charts)
 - Time graph views for kernel traces
 - Threads, CPUs, IRQs
 - Filters based on event name

Demo

Filter results in different views

Demo



Demo

The screenshot displays the LTTng Scope application interface. The left sidebar shows a tree view with 'kernel-1512167666' expanded to 'Traces', where 'Filters' and 'Create Filter...' are circled in red. The main area features three panels: 'Event count' (top), 'Threads' (middle), and 'CPU / IRQ' (bottom). The 'Event count' panel shows a line graph of event counts over time. The 'Threads' panel shows a Gantt chart for threads: 0/0 - swapper, 0/1 - kworker/0:0, 1 - init, 2 - kthreadd, 3 - ksoftirqd/0, and 6 - migration/0. The 'CPU / IRQ' panel shows activity for CPU 0 and IRQs 0, 9, and 16. A table at the bottom lists event details:

Timestamp	Trace	CPU	Event Type	Event Fields
1332170683.127231230	kernel	0	itimer_state	[which=0, expires=0, value_sec=0, value_usec=0, interval_sec=...
1332170683.127232487	kernel	0	exit_syscall	[ret=0]
1332170683.127234653	kernel	0	sys_clock_gettime	[which_clock=1, tp=0xbfd50b58]
1332170683.127236887	kernel	0	exit_syscall	[ret=0]
1332170683.127238354	kernel	0	sys_select	[n=256, inp=0x82463e0, outp=0x0, exp=0x0, tvp=0x82431a8]
1332170683.127251973	kernel	0	hrtimer_init	[hrtimer=4055104224, clockid=1, mode=0]
1332170683.127253440	kernel	0	hrtimer_start	[hrtimer=4055104224, function=3238473776, expires=428789...

Below the table, a summary section shows time ranges:

	Start	End	Span (s)
Visible Time Range	1332170683.109842001	1332170683.209842001	0.100000000
Selection Time Range	1332170683.127253389	1332170683.127253389	0.000000000
Trace Project Time Range	1332170682.440133097	1332170692.664579801	10.224446704

Demo

The screenshot shows the LTTng Scope interface. A 'Create New Filter' dialog box is open in the center, with the following fields:

- Filter Definition**
- Event Name:
- Symbol**
- Color:
- Shape:
- Buttons: Cancel, OK

The background interface includes:

- Event count:** A line graph showing event frequency over time.
- Threads:** A list of threads including 0/0 - swapper, 0/1 - kworker/0:0, 1 - init, 2 - kthreadd, 3 - ksoftirqd/0, and 6 - migration/0.
- CPU / IRQ:** A timeline view showing CPU activity and interrupt (IRQ) events.
- Event List Table:**

Timestamp	Trace	CPU	Event Type	Event Fields
1332170683.127231230	kernel	0	itimer_state	[which=0, expires=0, value_sec=0, value_usec=0, interval_sec=...
1332170683.127232487	kernel	0	exit_syscall	[ret=0]
1332170683.127234653	kernel	0	sys_clock_gettime	[which_clock=1, tp=0xbfd50b58]
1332170683.127236887	kernel	0	exit_syscall	[ret=0]
1332170683.127238354	kernel	0	sys_select	[n=256, inp=0x82463e0, outp=0x0, exp=0x0, tvp=0x82431a8]
1332170683.127251973	kernel	0	hrtimer_init	[hrtimer=4055104224, clockid=1, mode=0]
1332170683.127253440	kernel	0	hrtimer_start	[hrtimer=4055104224, function=3738473776, expires=428789...

At the bottom, there is a summary table:

	Start	End	Span (s)
Visible Time Range	1332170683.109842001	1332170683.209842001	0.100000000
Selection Time Range	1332170683.127253389	1332170683.127253389	0.000000000
Trace Project Time Range	1332170682.440133097	1332170692.664579801	10.224446704

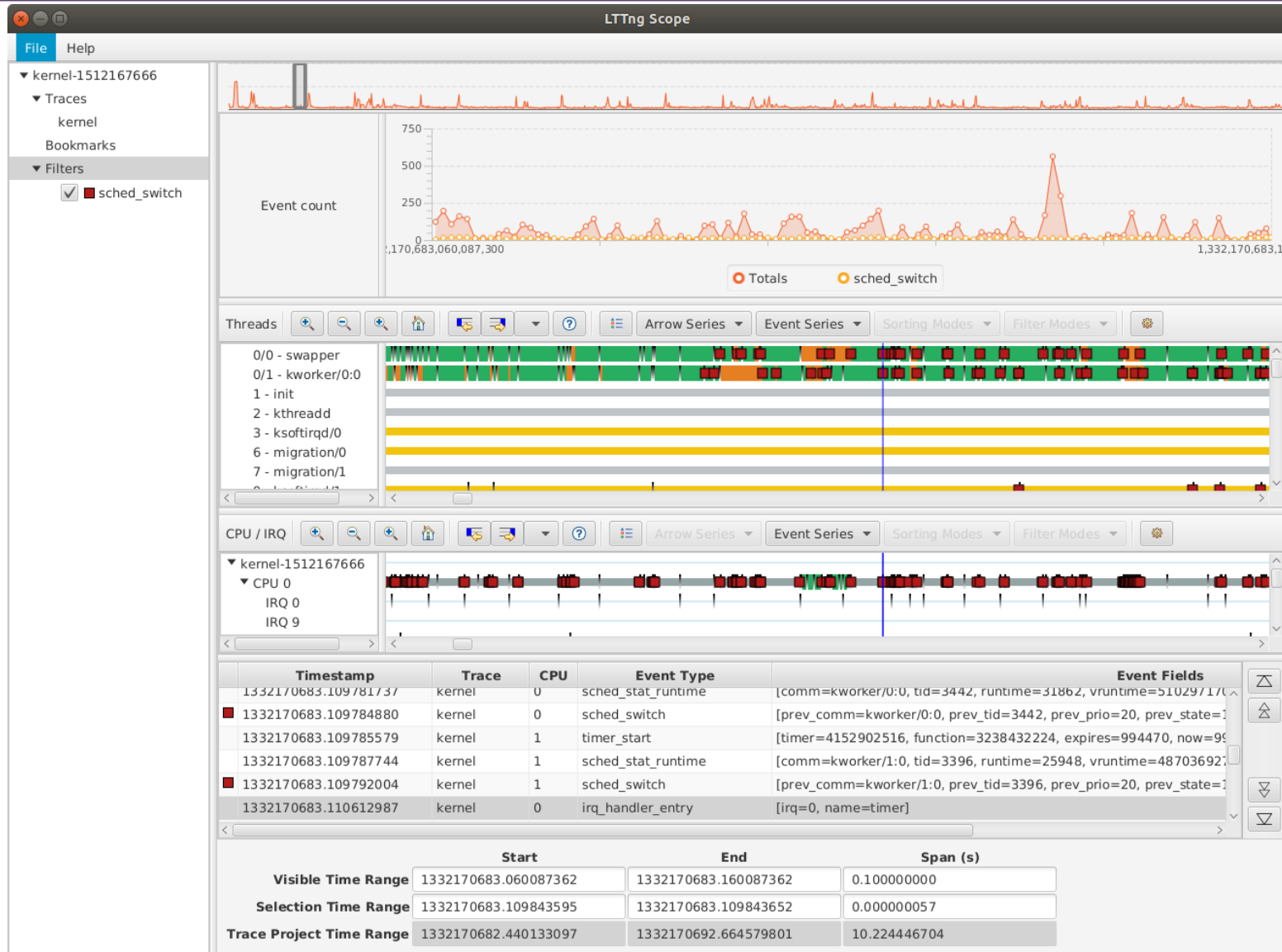
Demo

The screenshot shows the LTTng Scope interface. A 'Create New Filter' dialog box is open, allowing the user to define a filter for the event 'sched_switch'. The dialog includes fields for 'Event Name' (set to 'sched_switch'), 'Symbol', 'Color' (set to '#b31a1a'), and 'Shape' (set to a square). The 'OK' button is highlighted with a red circle. The background interface displays an 'Event count' graph, a 'Threads' list, a 'CPU / IRQ' view, and a table of event data.

Timestamp	Trace	CPU	Event Type	Event Fields
1332170683.127231230	kernel	0	itimer_state	[which=0, expires=0, value_sec=0, value_usec=0, interval_sec=
1332170683.127232487	kernel	0	exit_syscall	[ret=0]
1332170683.127234653	kernel	0	sys_clock_gettime	[which_clock=1, tp=0xbfd50b58]
1332170683.127236887	kernel	0	exit_syscall	[ret=0]
1332170683.127238354	kernel	0	sys_select	[n=256, inp=0x82463e0, outp=0x0, exp=0x0, tvp=0x82431a8]
1332170683.127251973	kernel	0	hrtimer_init	[hrtimer=4055104224, clockid=1, mode=0]
1332170683.127253440	kernel	0	hrtimer_start	[hrtimer=4055104224, function=3238473776, expires=428789

	Start	End	Span (s)
Visible Time Range	1332170683.109842001	1332170683.209842001	0.100000000
Selection Time Range	1332170683.127253389	1332170683.127253389	0.000000000
Trace Project Time Range	1332170682.440133097	1332170692.664579801	10.224446704

Demo



Demo

The screenshot shows the LTTng Scope interface with the following components and annotations:

- Filters:** A red circle highlights the 'sched_switch' filter in the 'Filters' section. A red note states: "Filter is now defined at the project level."
- Event Count:** A line graph showing event counts over time. A red circle highlights the 'sched_switch' series. A red note states: "XY charts add series corresponding to the matching events (Series style to be based on symbol color?)".
- Threads:** A timeline view showing threads like '0/0 - swapper', '0/1 - kworker/0:0', etc. A red note states: "Time graphs indicate matching events on the timeline."
- CPU / IRQ:** A timeline view showing CPU and IRQ activity.
- Event Table:** A table with columns: Timestamp, Trace, CPU, Event Type, and Event Fields. Two rows are highlighted with red circles. A red note states: "Event table tags matching events with the symbol."
- Summary:** A table at the bottom showing time ranges for Visible, Selection, and Trace Project.

Timestamp	Trace	CPU	Event Type	Event Fields
1332170683.109781737	kernel	0	sched_stat_runtime	[comm=kworker/0:0, tid=3442, runtime=31862, vruntime=510297170]
1332170683.109784880	kernel	0	sched_switch	[prev_comm=kworker/0:0, prev_tid=3442, prev_prio=20, prev_state=1]
1332170683.109785579	kernel	1	timer_start	[timer=4152902516, function=3238432224, expires=994470, now=994470]
1332170683.109787744	kernel	1	sched_stat_runtime	[comm=kworker/1:0, tid=3396, runtime=25948, vruntime=48703692]
1332170683.109792004	kernel	1	sched_switch	[prev_comm=kworker/1:0, prev_tid=3396, prev_prio=20, prev_state=1]
1332170683.110612987	kernel	0	irq_handler_entry	[irq=0, name=timer]

	Start	End	Span (s)
Visible Time Range	1332170683.060087362	1332170683.160087362	0.100000000
Selection Time Range	1332170683.109843595	1332170683.109843652	0.000000057
Trace Project Time Range	1332170682.440133097	1332170692.664579801	10.224446704

Demo

The screenshot shows the LTTng Scope interface. On the left, the 'Filters' panel has a red circle around the 'sched_switch' filter, which is currently disabled (checkbox is unchecked). A red text annotation below it reads: "Filter can be disabled and restored with one click." The main window displays an 'Event count' graph at the top, showing a significant spike in events. Below the graph are three views: 'Threads', 'CPU / IRQ', and a detailed event list. The 'Threads' view shows various kernel threads like 'swapper', 'kworker/0:0', and 'init'. The 'CPU / IRQ' view shows activity on CPU 0 and IRQ 0 and 9. The event list at the bottom shows several 'sched_switch' events with their respective timestamps, trace IDs, CPU numbers, and event fields.

Timestamp	Trace	CPU	Event Type	Event Fields
1332170683.109781737	kernel	0	sched_stat_runtime	[comm=kworker/0:0, tid=3442, runtime=3186z, vruntime=510297170]
1332170683.109784880	kernel	0	sched_switch	[prev_comm=kworker/0:0, prev_tid=3442, prev_prio=20, prev_state=1]
1332170683.109785579	kernel	1	timer_start	[timer=4152902516, function=3238432224, expires=994470, now=99]
1332170683.109787744	kernel	1	sched_stat_runtime	[comm=kworker/1:0, tid=3396, runtime=25948, vruntime=48703692]
1332170683.109792004	kernel	1	sched_switch	[prev_comm=kworker/1:0, prev_tid=3396, prev_prio=20, prev_state=1]
1332170683.110612987	kernel	0	irq_handler_entry	[irq=0, name=timer]

	Start	End	Span (s)
Visible Time Range	1332170683.060087362	1332170683.160087362	0.100000000
Selection Time Range	1332170683.109843595	1332170683.109843652	0.000000057
Trace Project Time Range	1332170682.440133097	1332170692.664579801	10.224446704

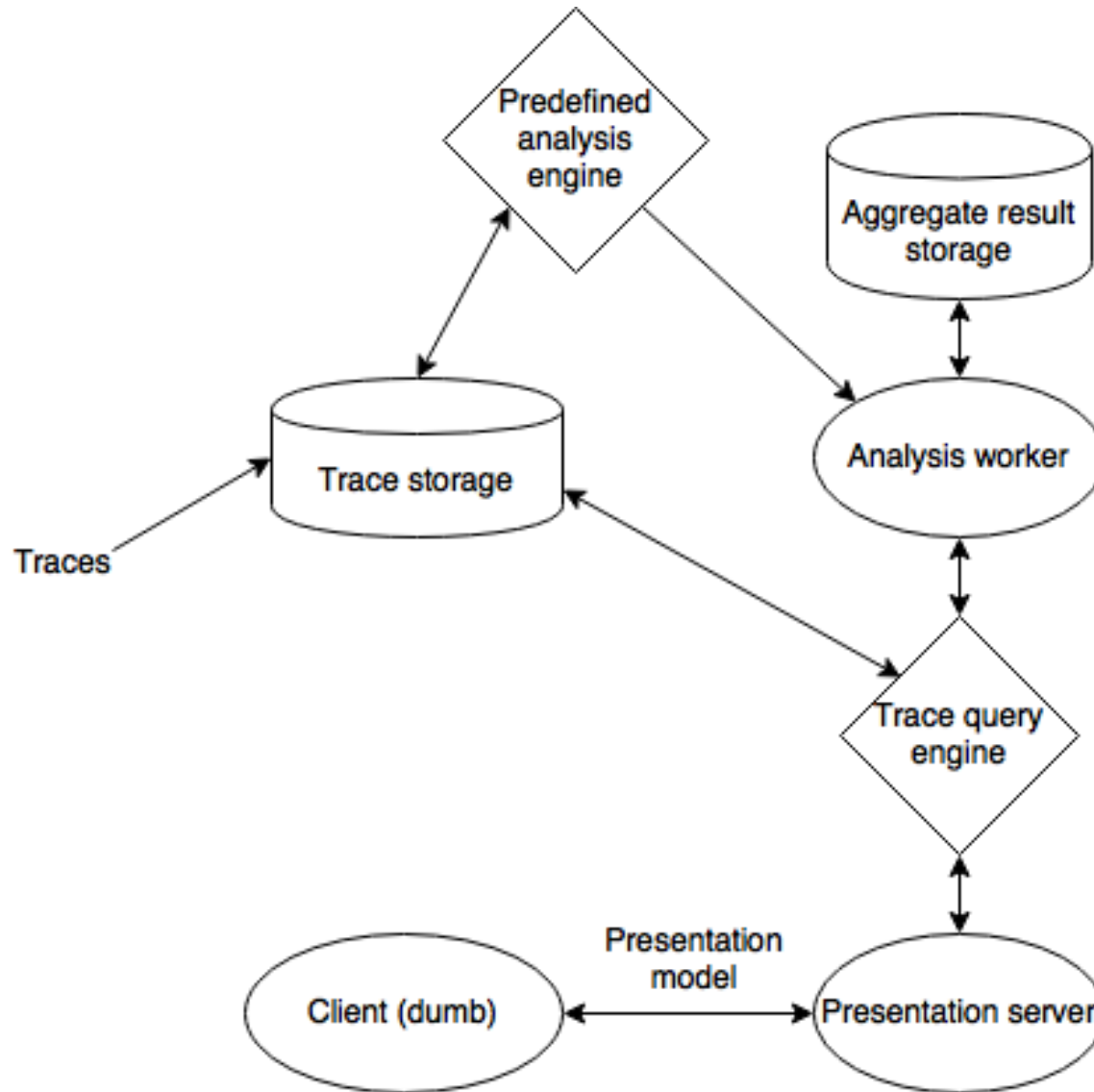
UI Design Roadmap

- *New project* dialog starts with high-level concepts:
 - CPU analysis
 - I/O analysis
 - Network analysis
 - ...
- This sets up a pre-defined set of widgets with pre-defined settings.
- User can then customize, add/remove, etc.

Architecture Roadmap

- Split client/server architecture
 - Choose the best frontend for the job:
 - CLI, Desktop, Web
 - Front-end should be as simple as possible.
 - Opens the door to completely new use cases:
 - Continuous Integration plugin embedding analysis results.
 - Distribute analyses over a cluster/cloud.

Architecture Roadmap



Architecture Roadmap

- Collaboration with Polytechnique students and Ericsson Trace Compass developers on defining a presentation protocol.

Links

- <https://github.com/lttng/lttng-scope>
 - Installation instructions
 - Report issues