ERICSSON

REAL-TIME SYSTEM TROUBLESHOOTING WITH TRACE COMPASS

MARC-ANDRÉ LAPERLE, ERICSSON

PRESENTER

- Marc-André Laperle
  - Software Developer at Ericsson since 2013
  - Eclipse Committer for Trace Compass, CDT, Linux Tools and Orbit
  - Occasional contributor to other projects (Platform UI, SWT, EGit, Mylyn, PDE)

AGENDA

1. Introduction and Overview
2. Trouble-shooting Workflow
3. Timing Analysis (new!)
4. Examples & Demos
5. Status

ABOUT PROJECT

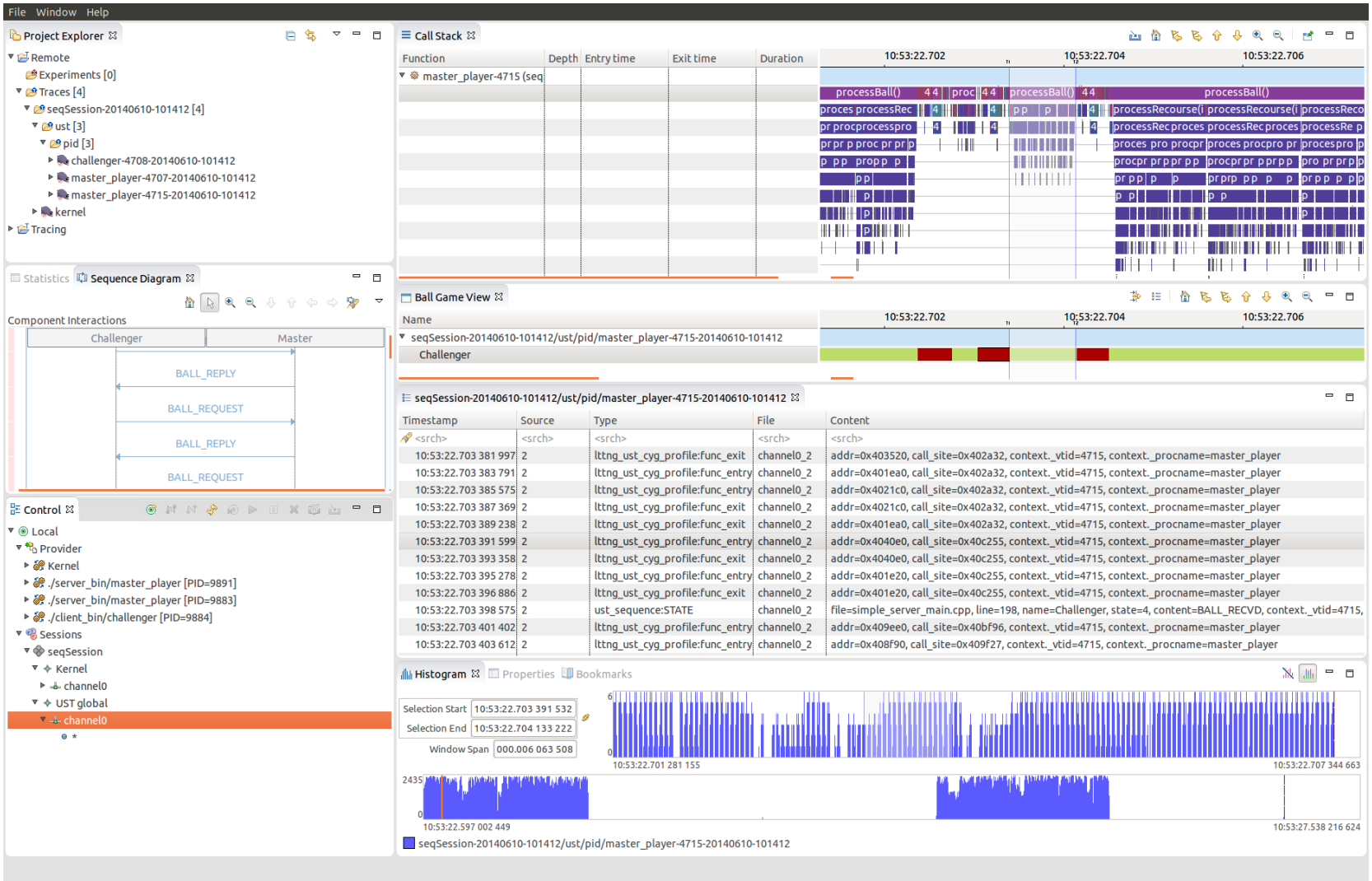
http://tracecompass.org

- Project under Eclipse Foundation
- Active committers and contributors from 3 organizations
  - Ericsson (5), EfficiOS (1), Polytechnique Montréal (2)
- Contributions and collaborations
  - Polytechnique, EfficiOS, Kalray, Windriver, CEA
- Trace Compass is part of PolarSys
- Strong relationship with the LTTng project (EfficiOS)
- Many interactions with the CDT Eclipse project
- Partnership with Trace Research Project at Polytechnique

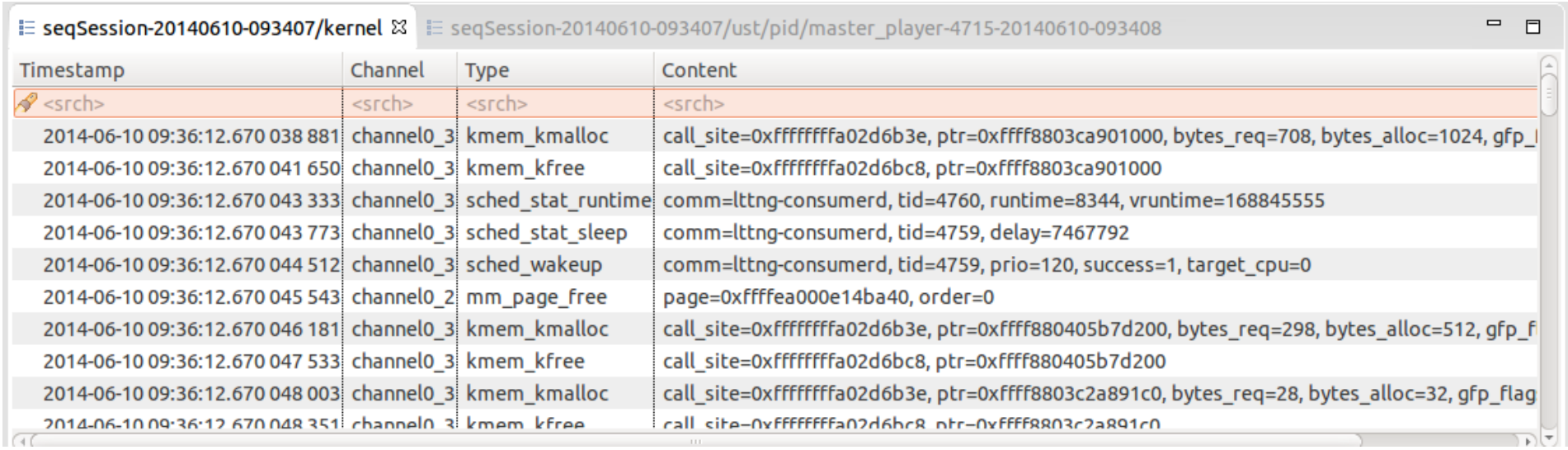
TRACE COMPASS

- Framework to build trace visualization and analysis tools
- Enables trace analysis/correlation from different sources
- Extensible for any trace/log format and analyses
- Reusable views and widgets
- Scalability allows to handle traces exceeding memory
- Support for standalone applications or set of plug-ins
- Support of various trace formats, e.g. CTF, PCAP, text…
- LTTng integration
- Trouble-shooting tool

TRACE COMPASS



EVENTS TABLE

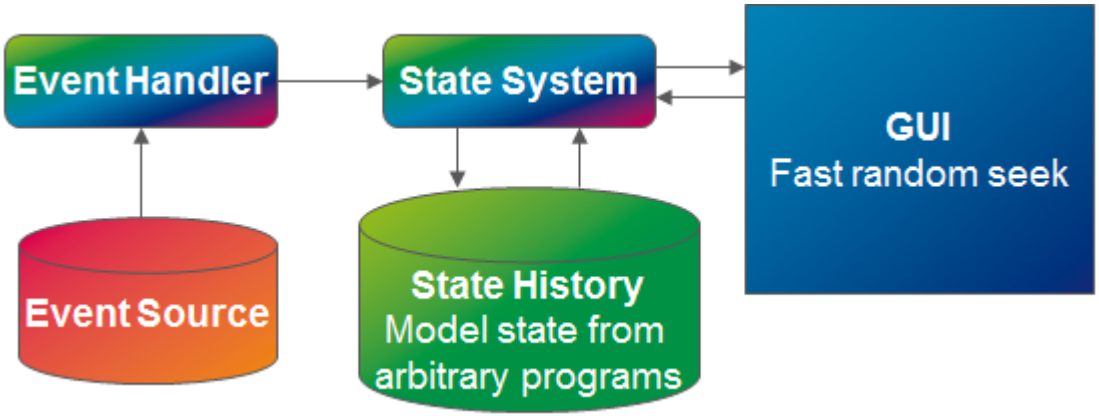


Timestamp	Channel	Type	Content
<srch>	<srch>	<srch>	<srch>
2014-06-10 09:36:12.670 038 881	channel0_3	kmem_kmalloc	call_site=0xffffffffa02d6b3e, ptr=0xffff8803ca901000, bytes_req=708, bytes_alloc=1024, gfp_f
2014-06-10 09:36:12.670 041 650	channel0_3	kmem_kfree	call_site=0xffffffffa02d6bc8, ptr=0xffff8803ca901000
2014-06-10 09:36:12.670 043 333	channel0_3	sched_stat_runtime	comm=lttng-consumerd, tid=4760, runtime=8344, vruntime=168845555
2014-06-10 09:36:12.670 043 773	channel0_3	sched_stat_sleep	comm=lttng-consumerd, tid=4759, delay=7467792
2014-06-10 09:36:12.670 044 512	channel0_3	sched_wakeup	comm=lttng-consumerd, tid=4759, prio=120, success=1, target_cpu=0
2014-06-10 09:36:12.670 045 543	channel0_2	mm_page_free	page=0xffffea000e14ba40, order=0
2014-06-10 09:36:12.670 046 181	channel0_3	kmem_kmalloc	call_site=0xffffffffa02d6b3e, ptr=0xffff880405b7d200, bytes_req=298, bytes_alloc=512, gfp_f
2014-06-10 09:36:12.670 047 533	channel0_3	kmem_kfree	call_site=0xffffffffa02d6bc8, ptr=0xffff880405b7d200
2014-06-10 09:36:12.670 048 003	channel0_3	kmem_kmalloc	call_site=0xffffffffa02d6b3e, ptr=0xffff8803c2a891c0, bytes_req=28, bytes_alloc=32, gfp_flag
2014-06-10 09:36:12.670 048 351	channel0_3	kmem_kfree	call_site=0xffffffffa02d6bc8, ptr=0xffff8803c2a891c0

Implemented as an Eclipse “editor”

7

STATE SYSTEM

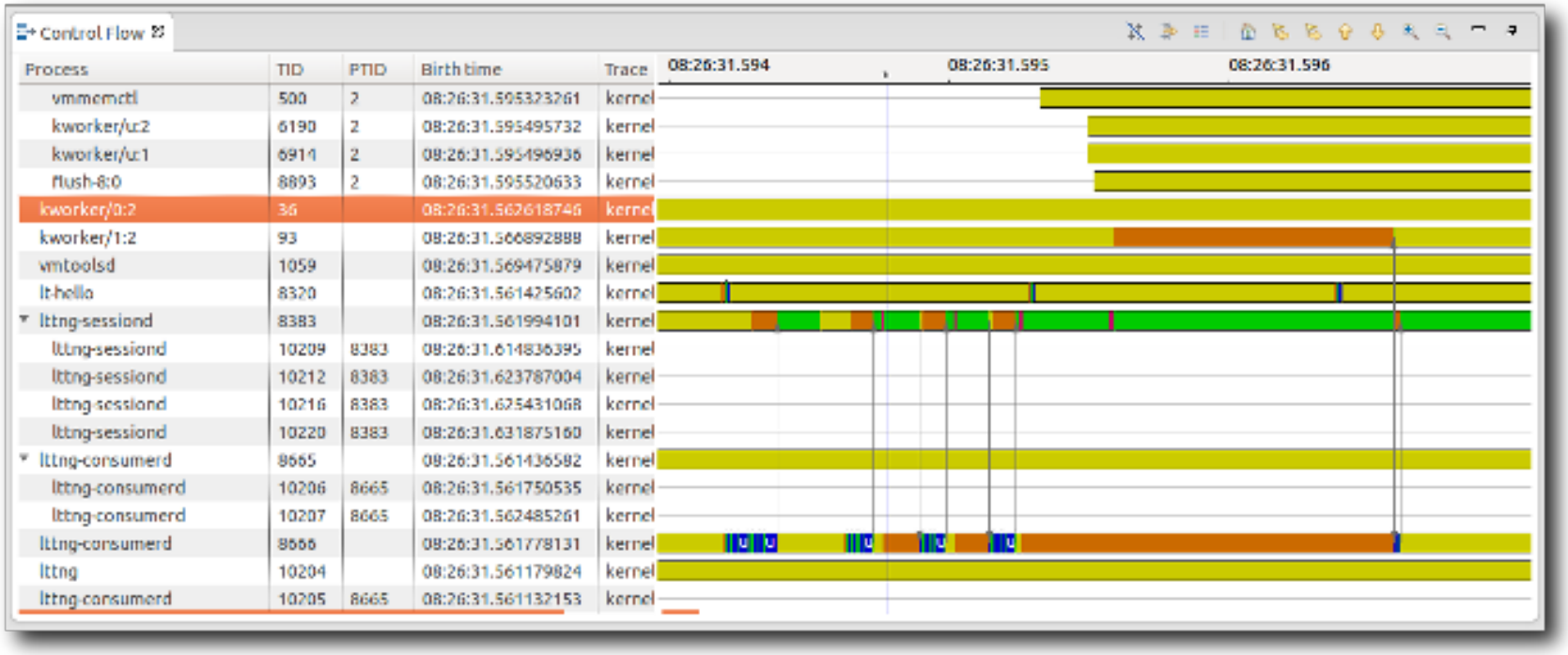


› State system abstracts events, analyses traces and creates models to be displayed

CONTROL FLOW VIEW

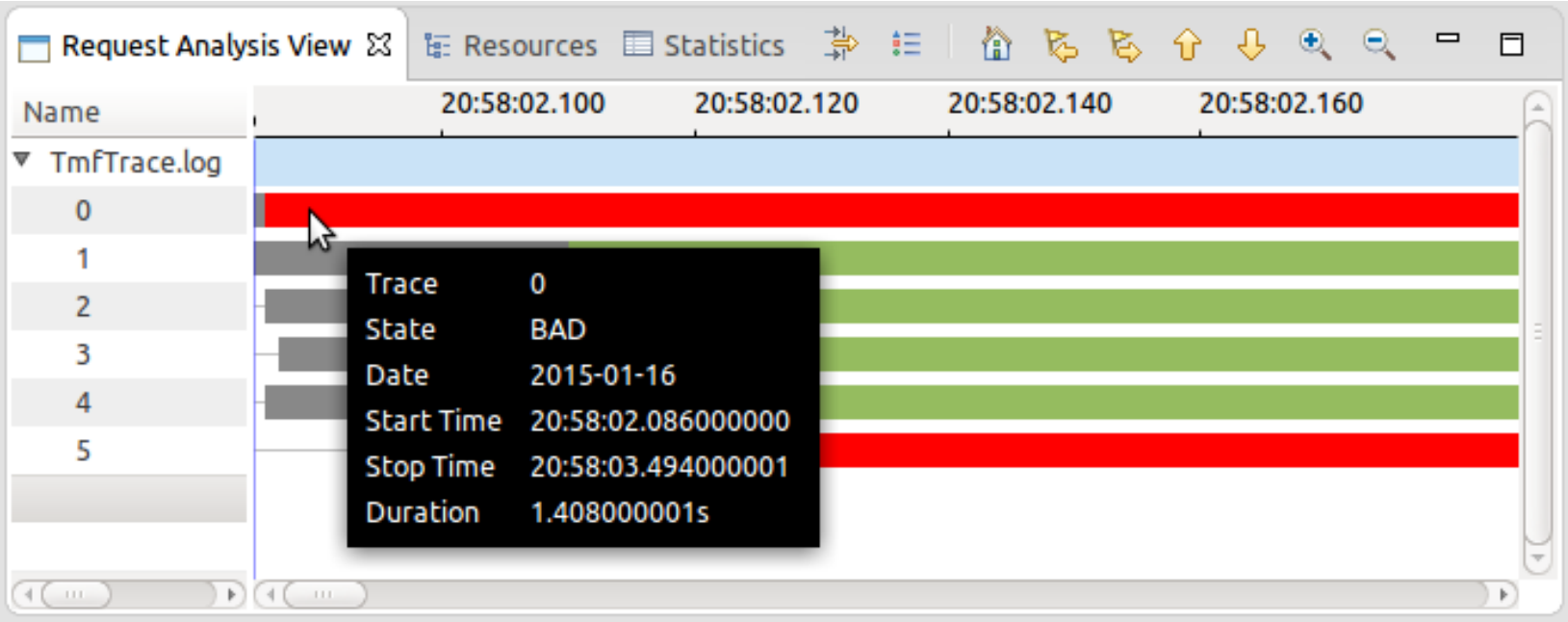
› Displays processes state changes (color-coded) over time

USERMODE, SYSCALL, INTERRUPED, WAIT_FOR_CPU, etc



XML VIEWS

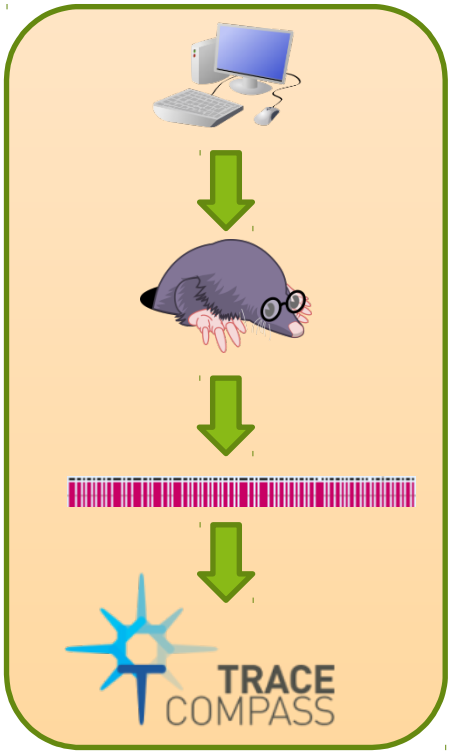
› EclipseCon NA 2015 example

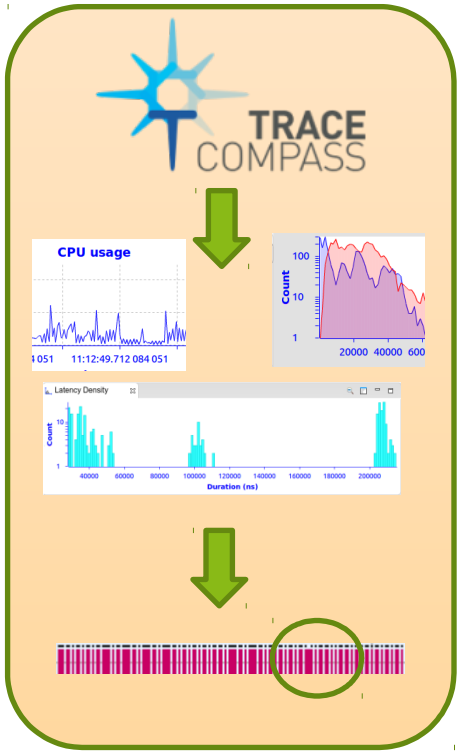


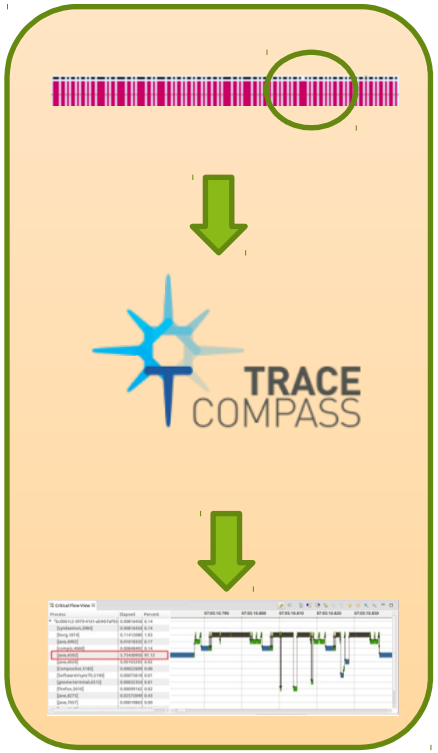

```
synchronized (TmfEventRequest.class) {
    fRequestId = fRequestNumber++;
}
```

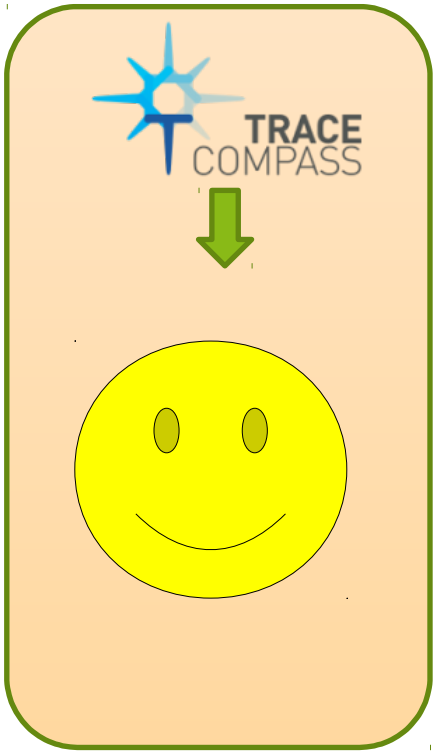

10

TROUBLE-SHOOTING WORKFLOW





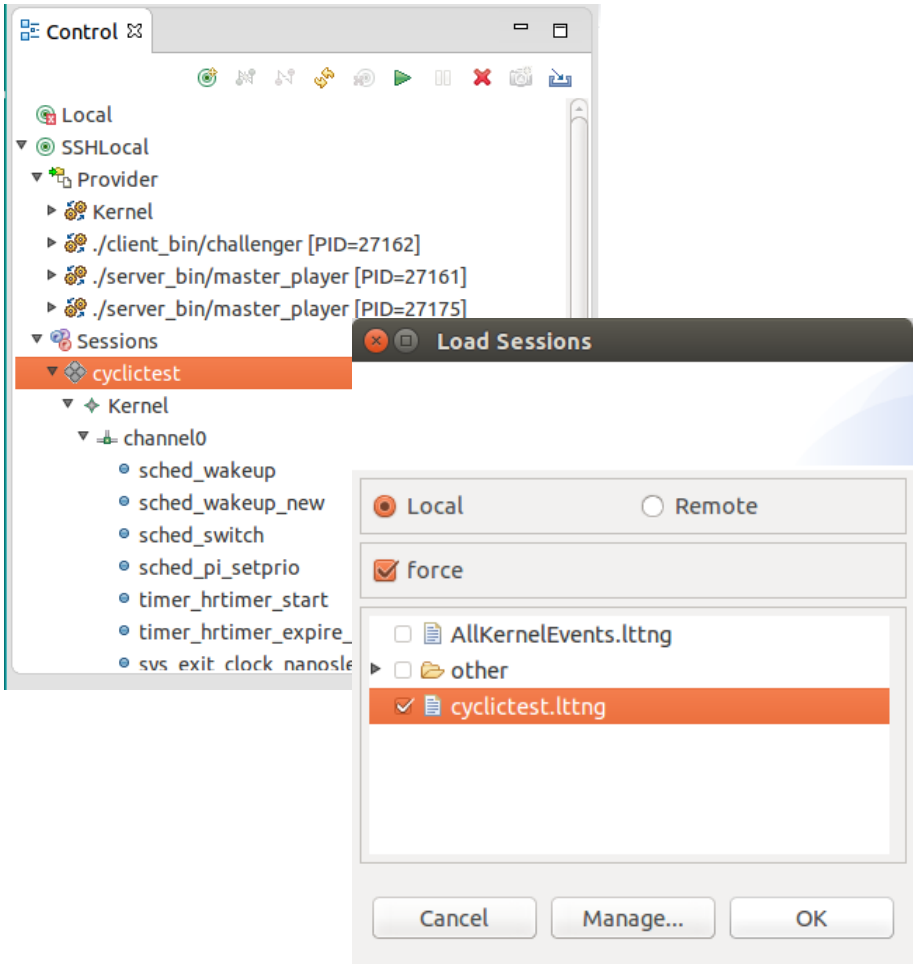




Collect → Locate → Analyze → Conclude

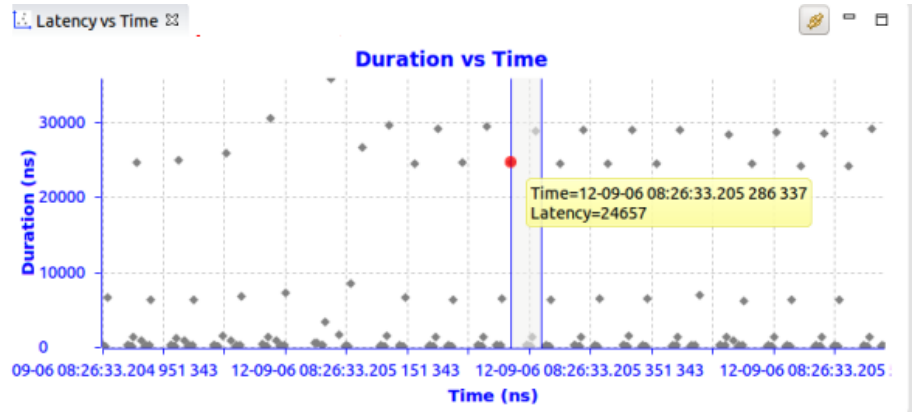
COLLECT DATA

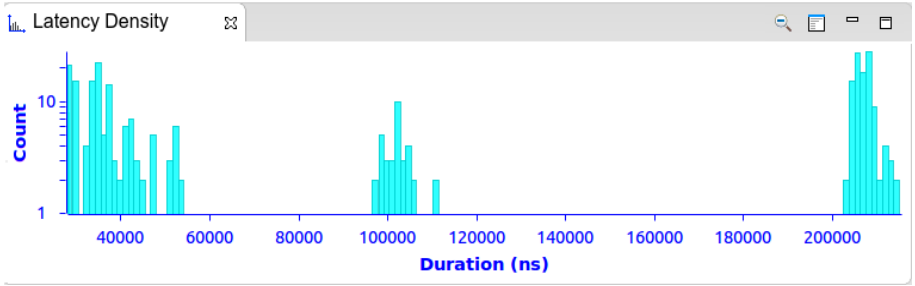
- Trace session configuration
  - LTTng control
  - Trace profiles support
- Streaming
- Importing traces or archives
  - Local
  - Remote
- Support multiple formats

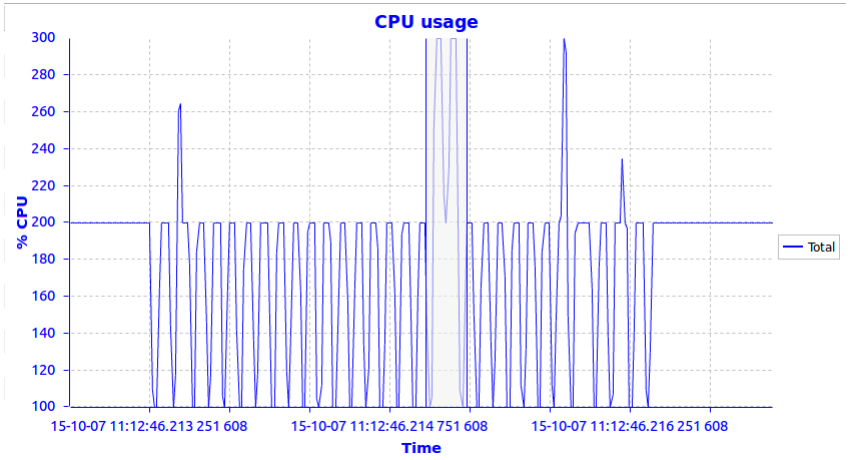


LOCATE PROBLEM

- Statistics
- Scatter Charts
- Histograms
- CPU Usage
- Memory Usage
- Pattern Matching & Filtering
- Searching
- Highlighting (markers)
- User-defined views (XML)

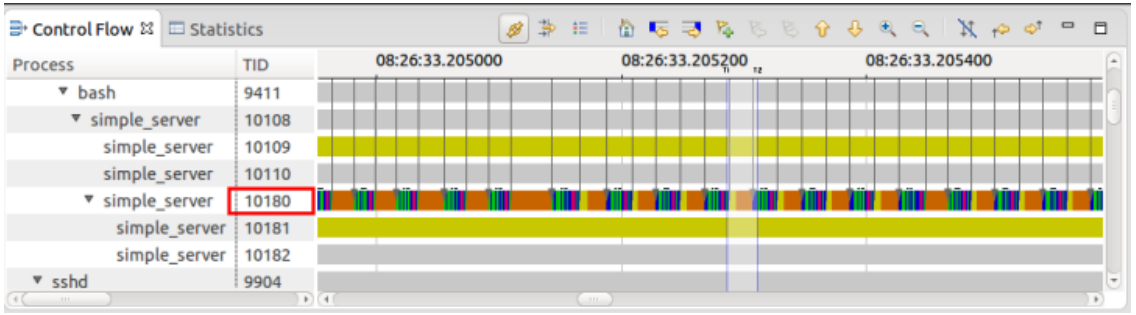


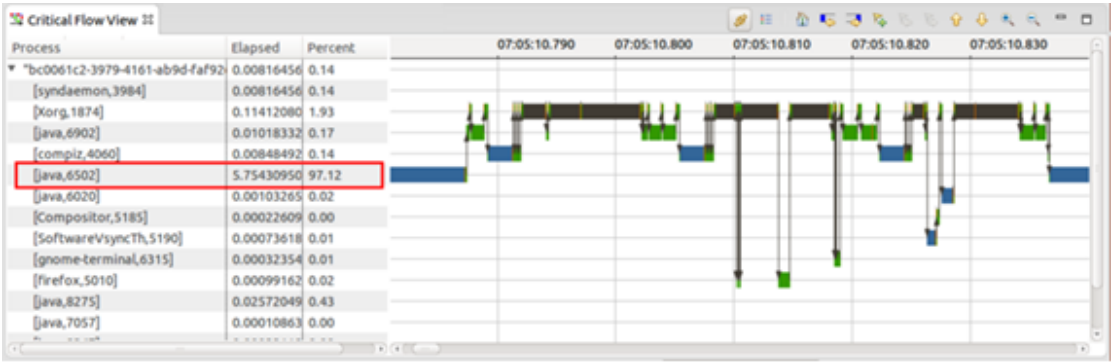




ANALYZE DATA

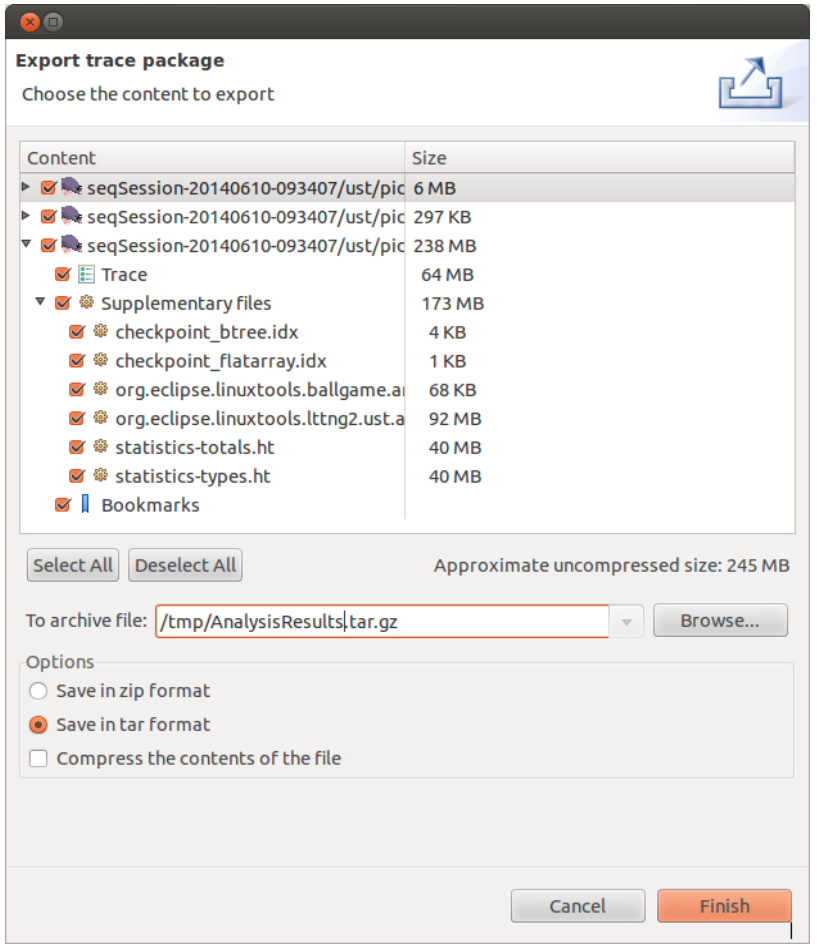
- Control Flow
- Resources Stats
- Critical Path
- Call Stacks
- Trace correlation
- Time-alignment
- Time synchronization





CONCLUDE

- Find Root Cause
  - Domain experts
- Report and share
  - Intermediate data
  - Settings, filters
  - Custom analysis (XML)
  - User bookmarks
- Solution



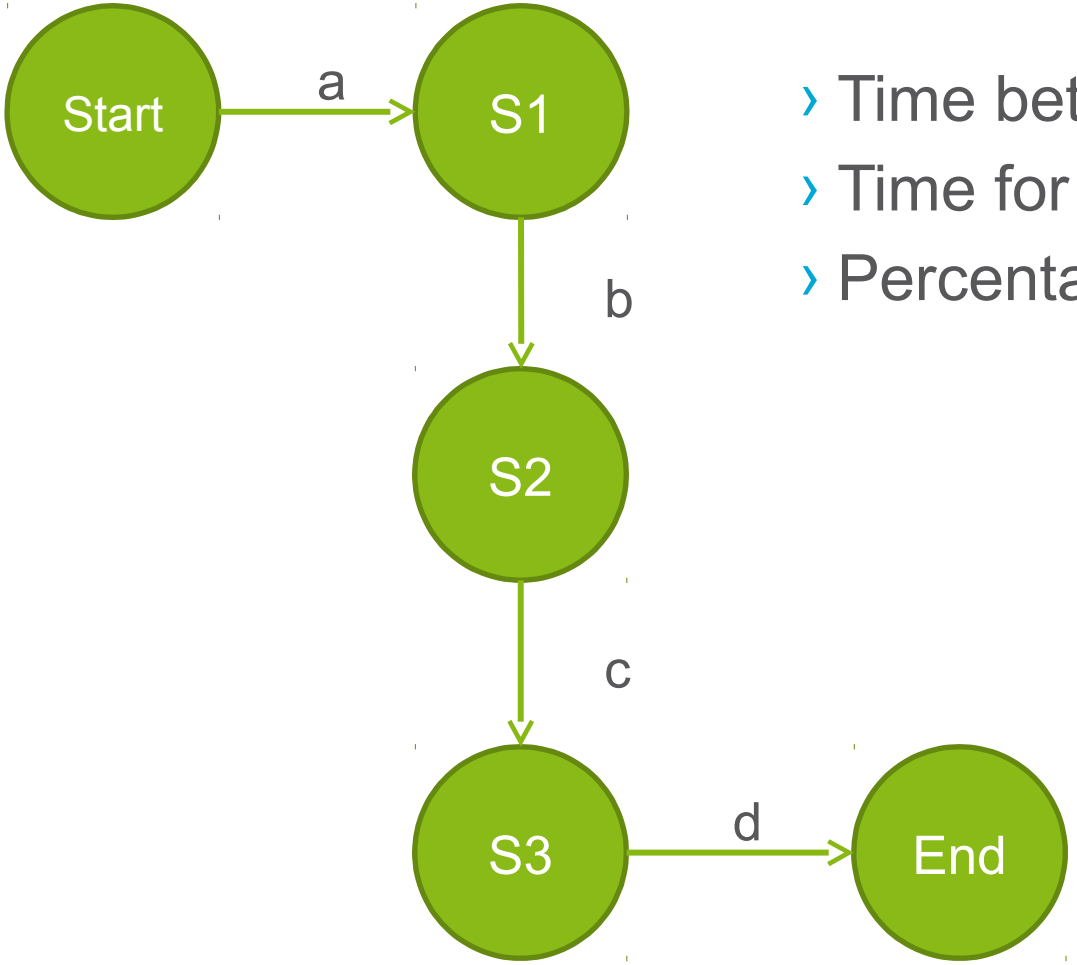
TIMING ANALYSIS

- We have two metrics to analyse, what is the data and when did it come.
- Every event has time info, let's take advantage of that.
- Potential delays (find problem before it occurs)
- Execution times, latencies, latency chains etc.
- Difficult to debug if it happens sporadically

TIMING ANALYSIS

- Measure time between a start and end state
  - Simple: Start and end event
  - Often: State Machine to determine start and end
- Locate timing problems
- Analyse timing problems
- Find root cause and solution

TIMING ANALYSIS



- Time between start and end
- Time for each transition
- Percentage sub-duration vs total

TIMING ANALYSIS

- State machine for timing analysis
  - Implementation in Java as Trace Compass extension
  - Data-driven pattern matching (in XML)
    - Defining timing analyses on-the-fly

VISUALIZATION

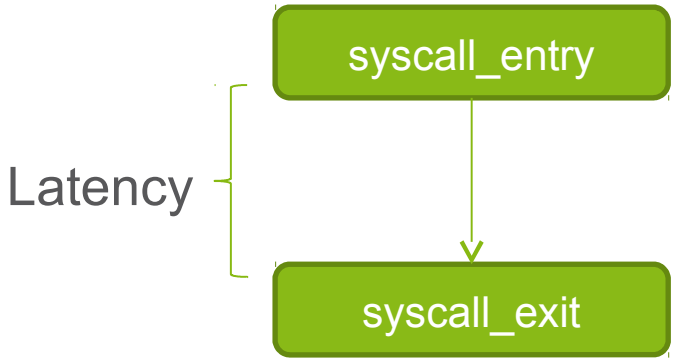
- Table
  - Get raw data
  - Sorting, highlighting, filtering
- Scatter Chart
  - Have a big picture of the current range
- Distribution Chart
  - Find outliers and modes easily

VISUALIZATION

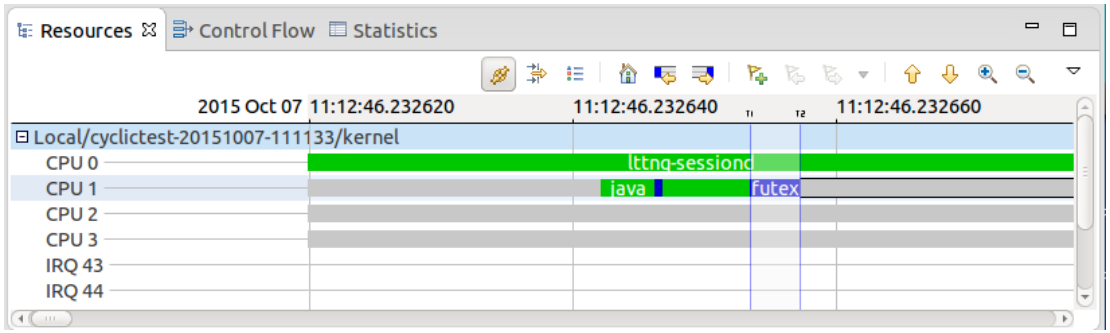
- Markers
  - Per trace, view and user defined
- Statistics
  - Min, max, average, standard deviation
  - Find worst offenders
  - Find worst possible offender combinations

EXAMPLE

› System Call Latency, e.g. futex

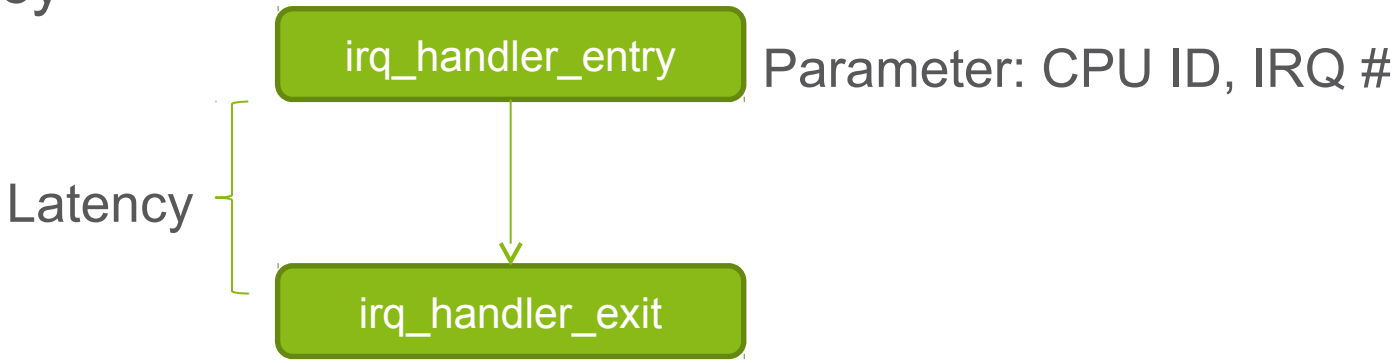


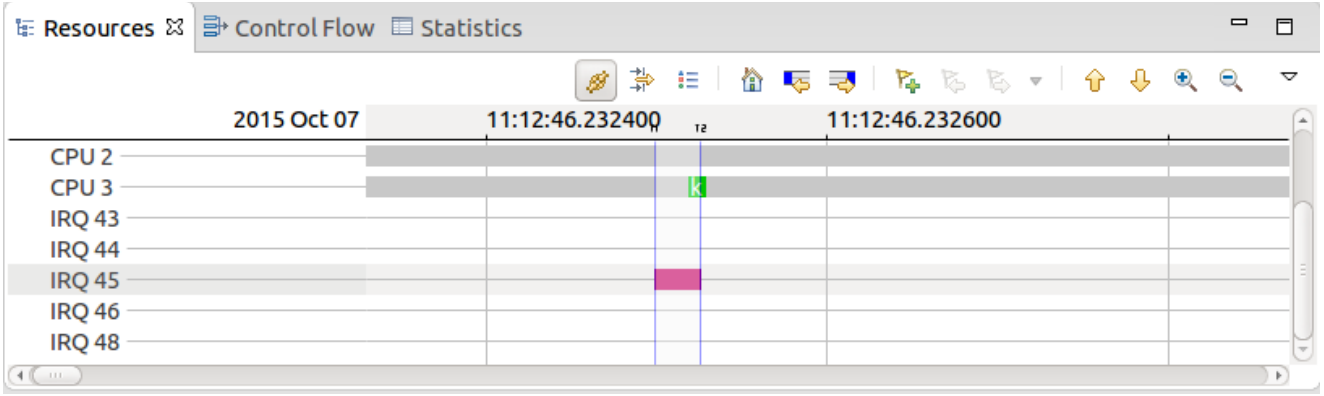
Parameter: name, CPU ID, TID



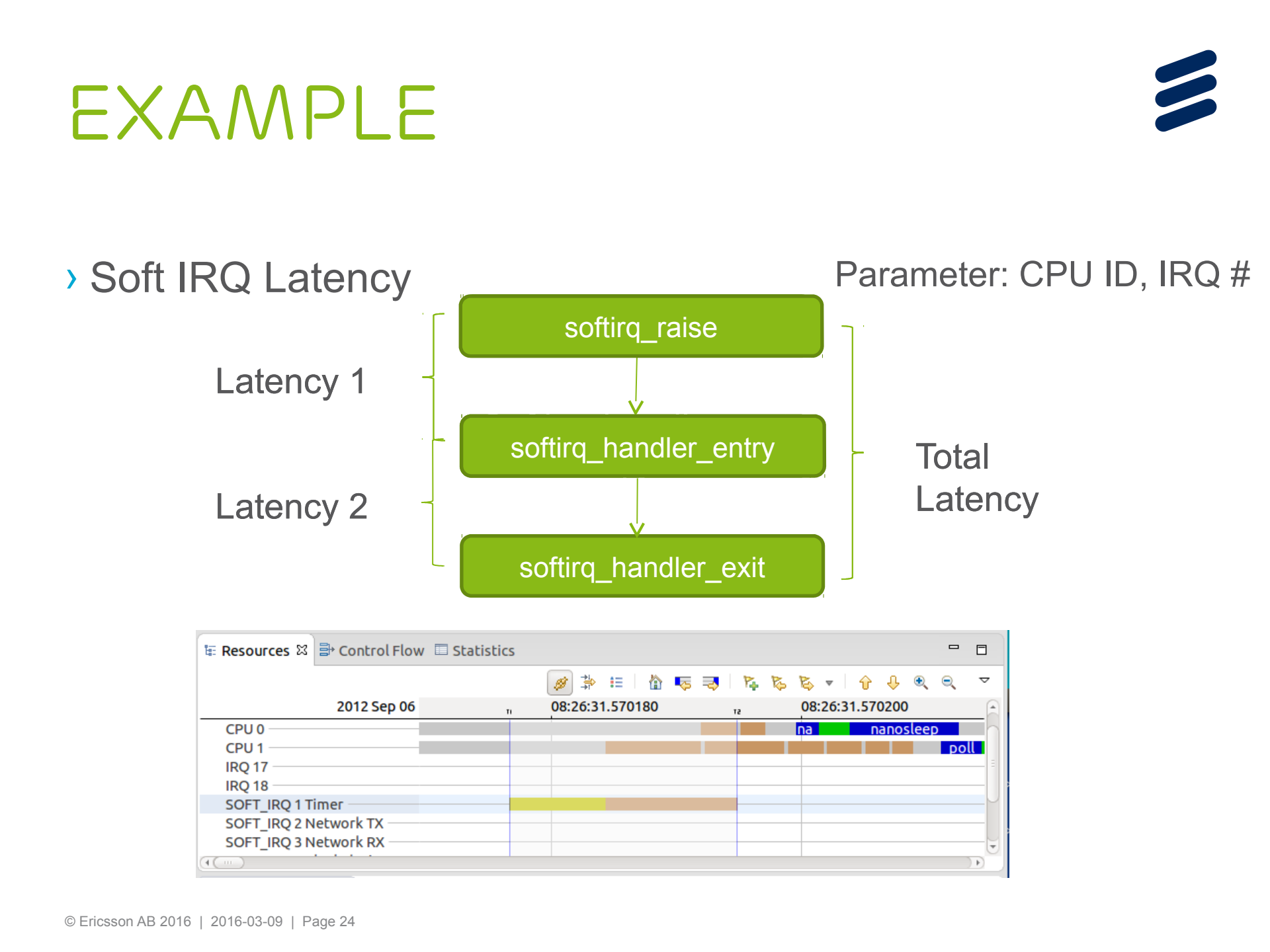
EXAMPLE

- IRQ Latency





EXAMPLE

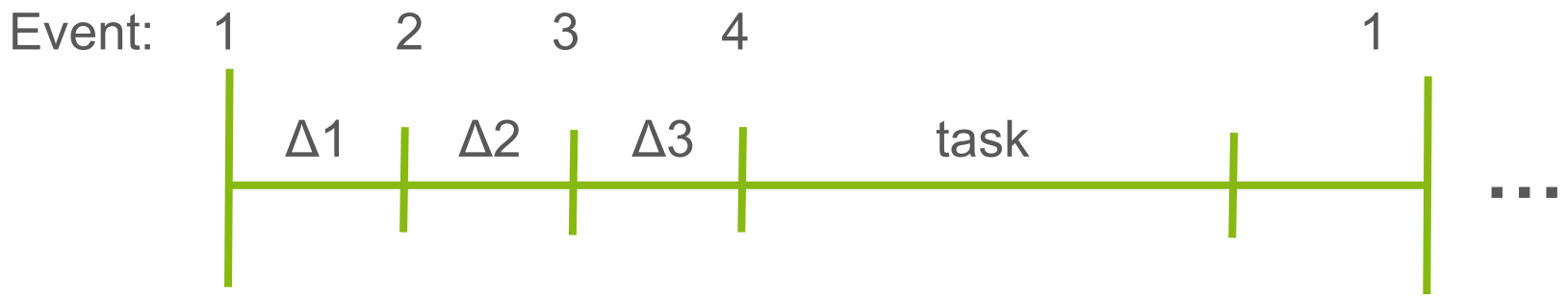


› Soft IRQ Latency

Parameter: CPU ID, IRQ #

EXAMPLE APPLICATION

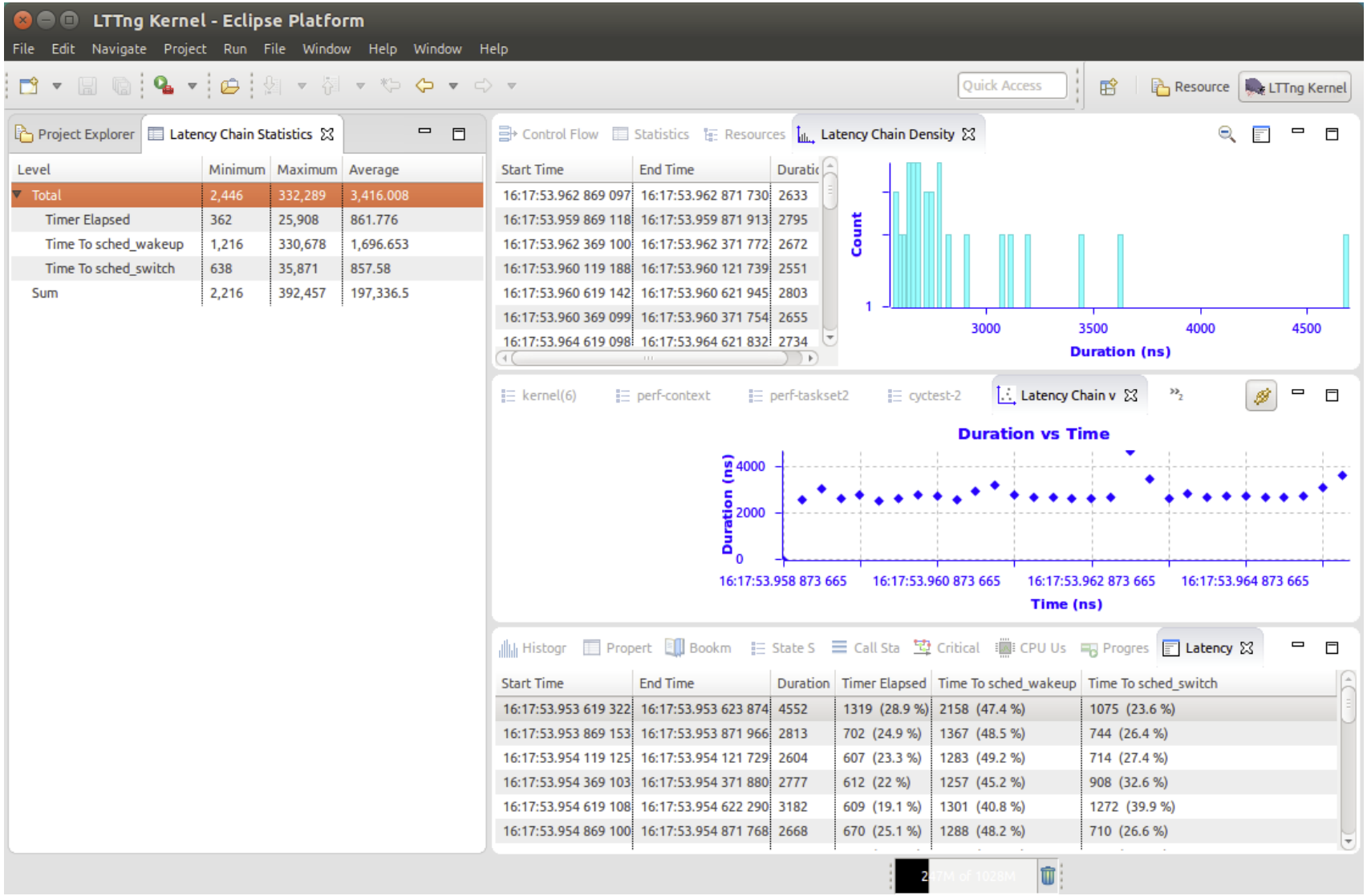
- High Resolution Timer – cyclictest application of rt-tests
- Latency between timer expiry till task starts



- Latency = Δ1+ Δ2 + Δ3

- Event 1: Timer expires
- Event 2: Interrupt handler marks the task to react
- Event 3: Linux scheduler switches to the task
- Event 4: Application task begins executing

EXAMPLE ANALYSIS



DEMO

Noticing some driver issue

STATUS

- Framework support for timing analyses
  - Extensible timing analysis module
  - Extensible views (table, scatter chart, density, stats)
  - Data storage (in memory)
- System Call Analysis
  - Written in Java
  - Table, scatter chart, statistics, density
- Critical Path Analysis and view

STATUS

- User defined markers in Time Graph views
  - For annotating trace (bookmarks)
  - Single time or time ranges
- Trace Markers
  - Extensible in Java
  - Lost events markers in Time Graph views
- Possible to navigate between markers
- Possible to enable/disable display of marker categories

ONGOING

- Integration of Pattern Matching
  - Data-driven approach
  - Enables enhanced filtering
  - Enables defining on-the-fly latency analyses
- Improve IRQ/Soft IRQ latency analyses and views
- User-defined periodic markers (XML)
- Integration of Python analysis, reports etc. (EfficiOS)

NEW IN NEON

1 Pie charts in Statistics view

2 Time graph view improvements
- 2.1 Gridlines in time graph views
- 2.2 Persist settings for open traces
- 2.3 Bookmarks and custom markers in time graph views

3 System call latency analysis
- 3.1 System call latency table
- 3.2 System call latency scatter graph
- 3.3 System call latency statistics
- 3.4 System call latency density

4 Critical flow view

5 Virtual CPU view

6 Bookmarks and custom markers
- 6.1 Support for user bookmarks in time graph views
- 6.2 Lost event markers in time graph views
- 6.3 Navigation for trace markers in time graph Views
- 6.4 API for trace specific markers

7 Display of soft IRQ names in the Resources view

8 Support for sorting in Control Flow view

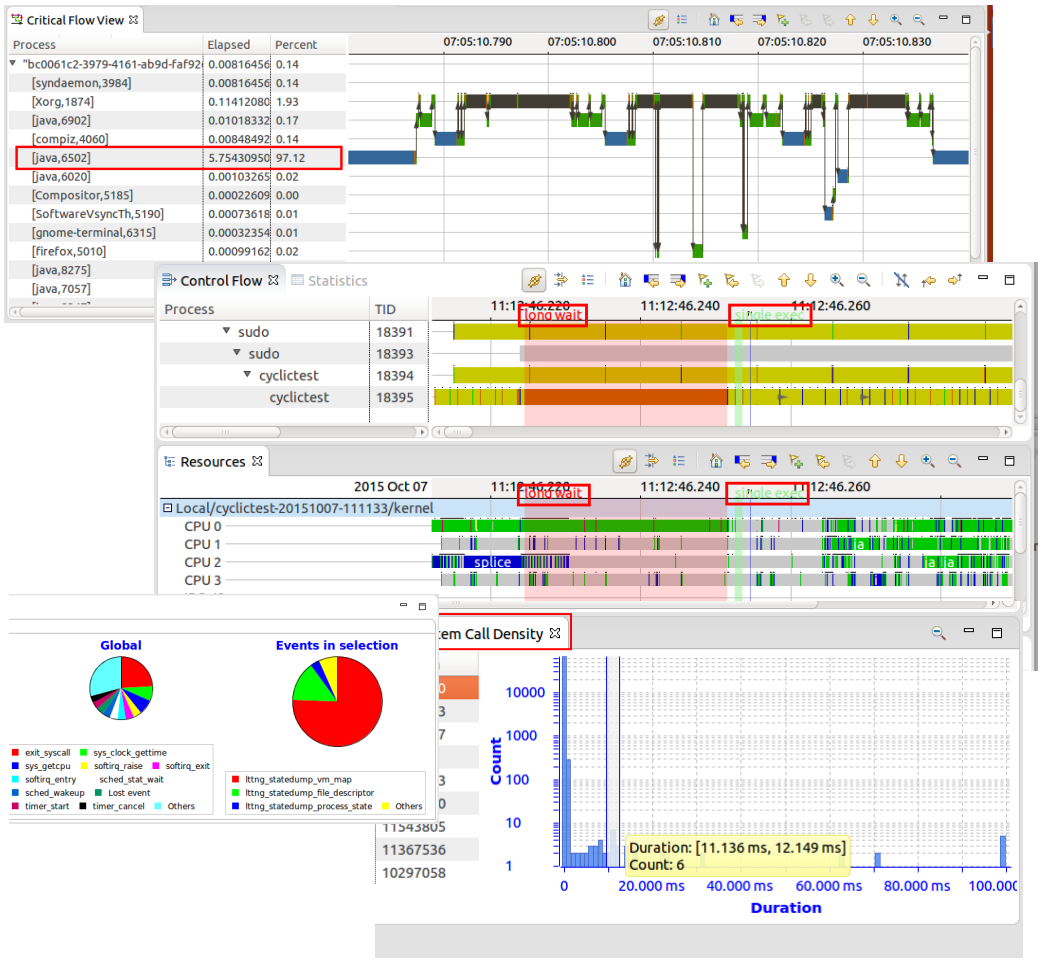
9 Support of vertical zooming in time graph views

10 Manage XML analysis files

11 Display of analysis properties

12 Visualization of debug info for source code lookup coming with LTTng 2.8 traces

13 Bugs fixed in this release



› https://wiki.eclipse.org/Trace_Compass/News/NewIn20

PLANNED

- Data-driven definition of latency views
- Filtering within latency views (e.g. per syscall type)
- Performance optimisations for latency analysis (memory!)
- Search and filtering improvements in Events table

REFERENCES

- Project pages
  - http://tracecompass.org
    https://dev.eclipse.org/mailman/listinfo/tracecompass-dev
  - Is Trace Compass Fast Yet? http://istmffastyet.dorsal.polymtl.ca/
  - http://lttng.org/
  - https://www.polarsys.org/
  - http://www.diamon.org/
  - http://tracingsummit.org/

- Documentations
  - Trace Compass User Guide
  - Trace Compass Developer Guide

CONTACT

- Mailing list: tracecompass-dev@eclipse.org
- IRC: oftc.net #tracecompass
- Mattermost:

  https://mattermost-test.eclipse.org/eclipse/channels/trace-compass

Q&A

eclipsecon
RESTON VIRGINIA 2016

Evaluate the Sessions

Sign in and vote at **eclipsecon.org**

- 1 0 + 1

ERICSSON