ADVANCED TROUBLE-SHOOTING OF REAL-TIME SYSTEMS

BERND HUFMANN, ERICSSON

AGENDA

1 Introduction

2 Trace Compass Overview

3 Timing Analysis

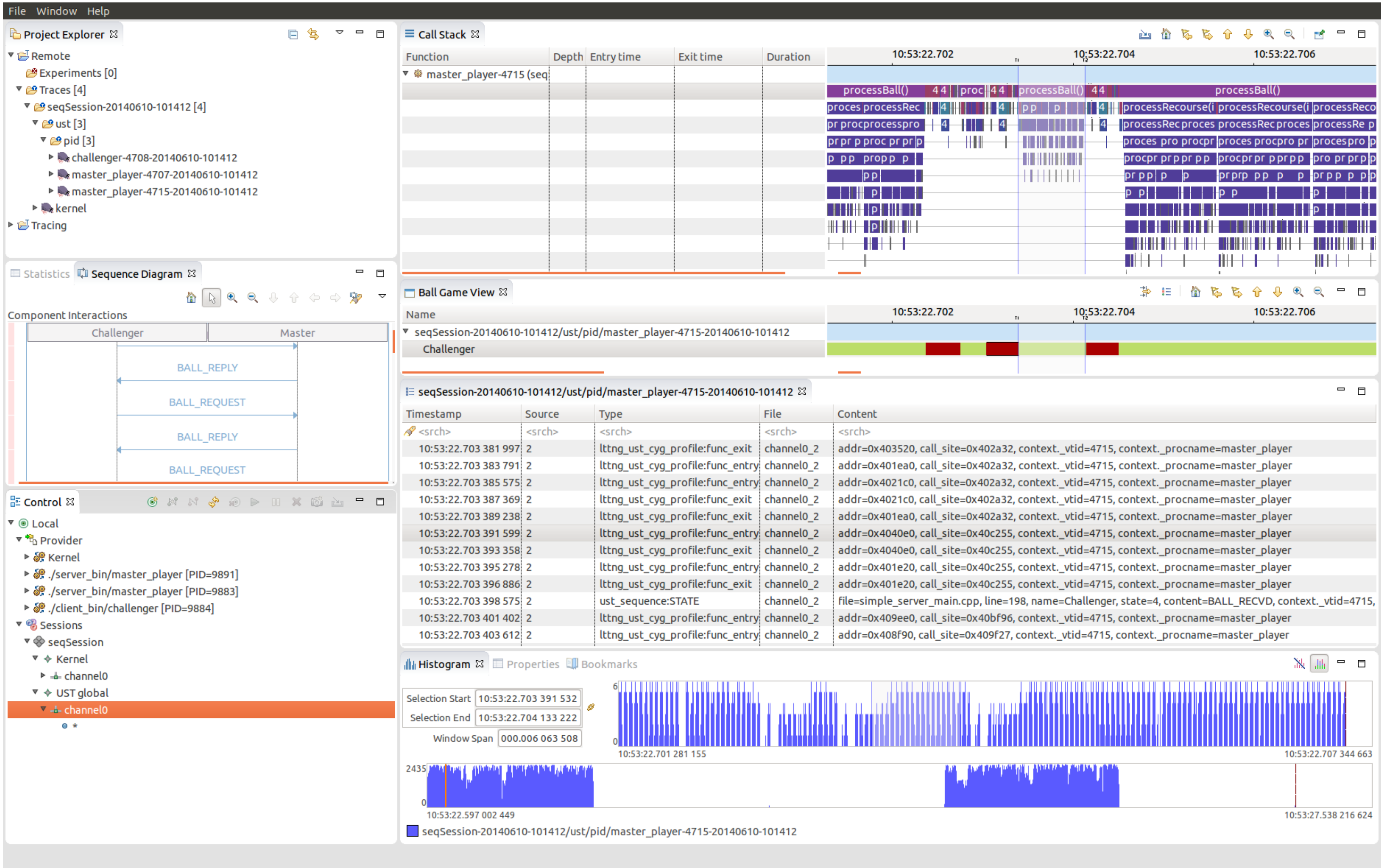
4 References

5 Q&A

TRACE COMPASS OVERVIEW

- Troubleshooting tool
- Framework to build trace visualization and analysis tools
- Scalable: handle traces exceeding memory
- Extensible for any trace or log format: Binary, text, XML etc.
- Reusable views and widgets
- Available as standalone product or set of plug-ins

TRACE COMPASS OVERVIEW



Project Explorer

- Remote
  - Experiments [0]
  - Traces [4]
    - seqSession-20140610-101412 [4]
      - ust [3]
        - pid [3]
          - challenger-4708-20140610-101412
          - master_player-4707-20140610-101412
          - master_player-4715-20140610-101412
      - kernel
- Tracing

Call Stack

- master_player-4715 (seq

Ball Game View

- seqSession-20140610-101412/ust/pid/master_player-4715-20140610-101412
  - Challenger

Control

- Local
  - Provider
    - Kernel
    - ./server_bin/master_player [PID=9891]
    - ./server_bin/master_player [PID=9883]
    - ./client_bin/challenger [PID=9884]
  - Sessions
    - seqSession
      - Kernel
        - channel0
      - UST global
        - channel0
          - *

seqSession-20140610-101412/ust/pid/master_player-4715-20140610-101412

Timestamp	Source	Type	File	Content
<srch>	<srch>	<srch>	<srch>	<srch>
10:53:22.703 381 997	2	lttng_ust_cyg_profile:func_exit	channel0_2	addr=0x403520, call_site=0x402a32, context._vtid=4715, context._procname=master_player
10:53:22.703 383 791	2	lttng_ust_cyg_profile:func_entry	channel0_2	addr=0x401ea0, call_site=0x402a32, context._vtid=4715, context._procname=master_player
10:53:22.703 385 575	2	lttng_ust_cyg_profile:func_entry	channel0_2	addr=0x4021c0, call_site=0x402a32, context._vtid=4715, context._procname=master_player
10:53:22.703 387 369	2	lttng_ust_cyg_profile:func_exit	channel0_2	addr=0x4021c0, call_site=0x402a32, context._vtid=4715, context._procname=master_player
10:53:22.703 389 238	2	lttng_ust_cyg_profile:func_exit	channel0_2	addr=0x401ea0, call_site=0x402a32, context._vtid=4715, context._procname=master_player
10:53:22.703 391 599	2	lttng_ust_cyg_profile:func_entry	channel0_2	addr=0x4040e0, call_site=0x40c255, context._vtid=4715, context._procname=master_player
10:53:22.703 393 358	2	lttng_ust_cyg_profile:func_exit	channel0_2	addr=0x4040e0, call_site=0x40c255, context._vtid=4715, context._procname=master_player
10:53:22.703 395 278	2	lttng_ust_cyg_profile:func_entry	channel0_2	addr=0x401e20, call_site=0x40c255, context._vtid=4715, context._procname=master_player
10:53:22.703 396 886	2	lttng_ust_cyg_profile:func_exit	channel0_2	addr=0x401e20, call_site=0x40c255, context._vtid=4715, context._procname=master_player
10:53:22.703 398 575	2	ust_sequence:STATE	channel0_2	file=simple_server_main.cpp, line=198, name=Challenger, state=4, content=BALL_RECVD, context._vtid=4715,
10:53:22.703 401 402	2	lttng_ust_cyg_profile:func_entry	channel0_2	addr=0x409ee0, call_site=0x40bf96, context._vtid=4715, context._procname=master_player
10:53:22.703 403 612	2	lttng_ust_cyg_profile:func_entry	channel0_2	addr=0x408f90, call_site=0x409f27, context._vtid=4715, context._procname=master_player

Histogram

Selection Start 10:53:22.703 391 532

Selection End 10:53:22.704 133 222

Window Span 000.006 063 508

10:53:22.701 281 155

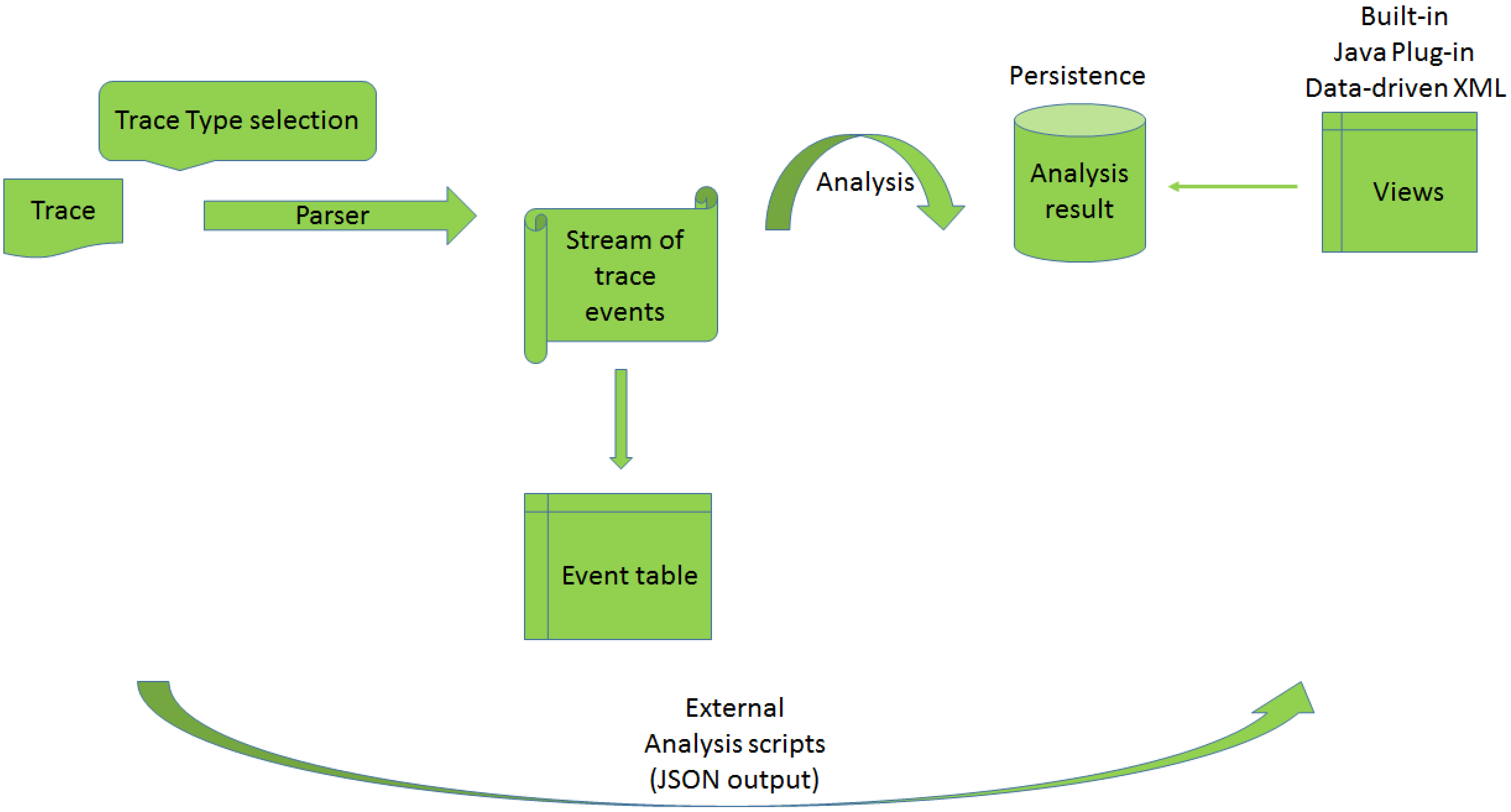
10:53:22.707 344 663

10:53:22.597 002 449

10:53:27.538 216 624

seqSession-20140610-101412/ust/pid/master_player-4715-20140610-101412

DATA FLOW



Built-in
Java Plug-in
Data-driven XML

Persistence

Trace Type selection

Trace

Parser

Stream of trace events

Analysis

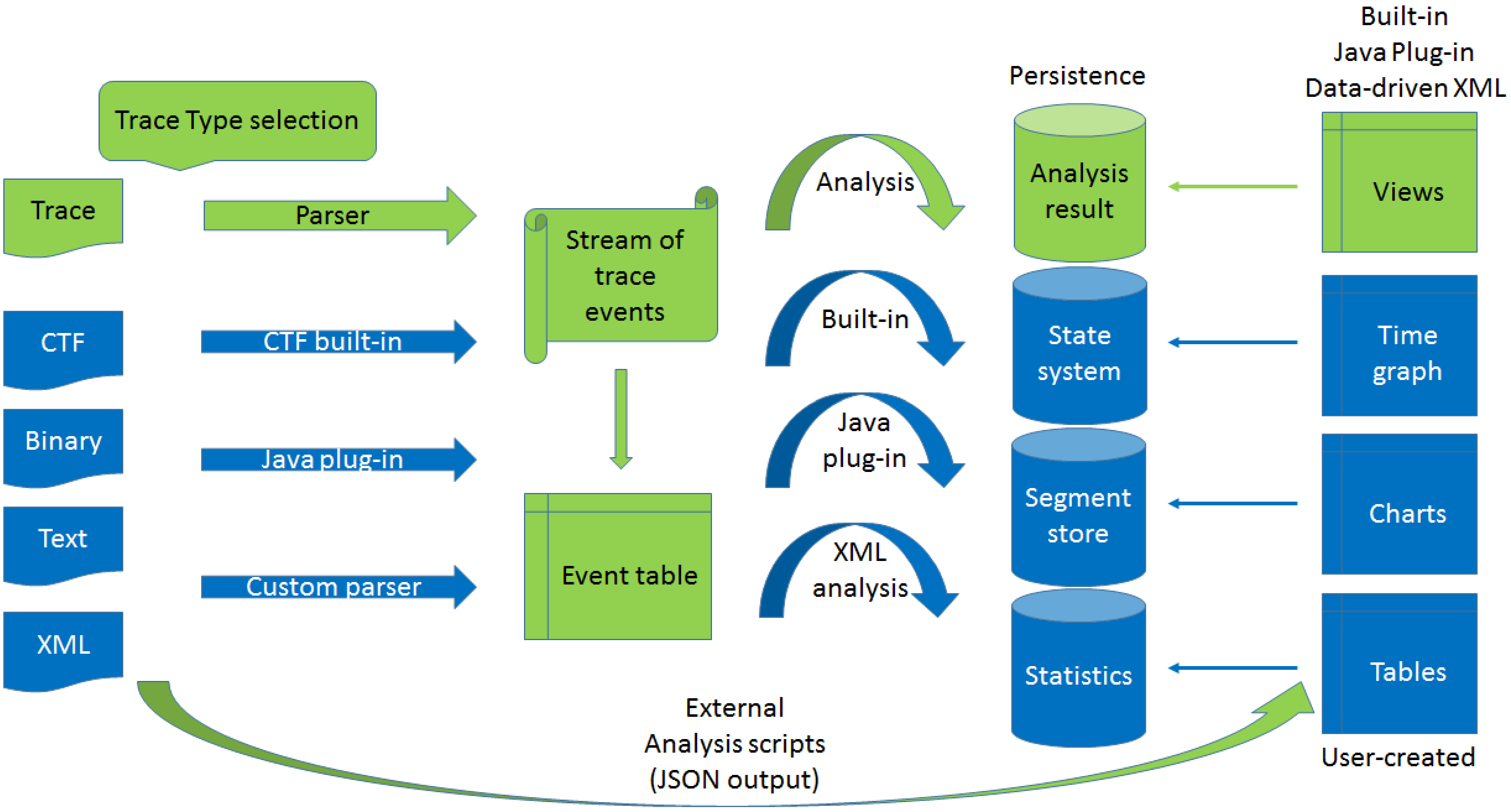
Analysis result

Views

Event table

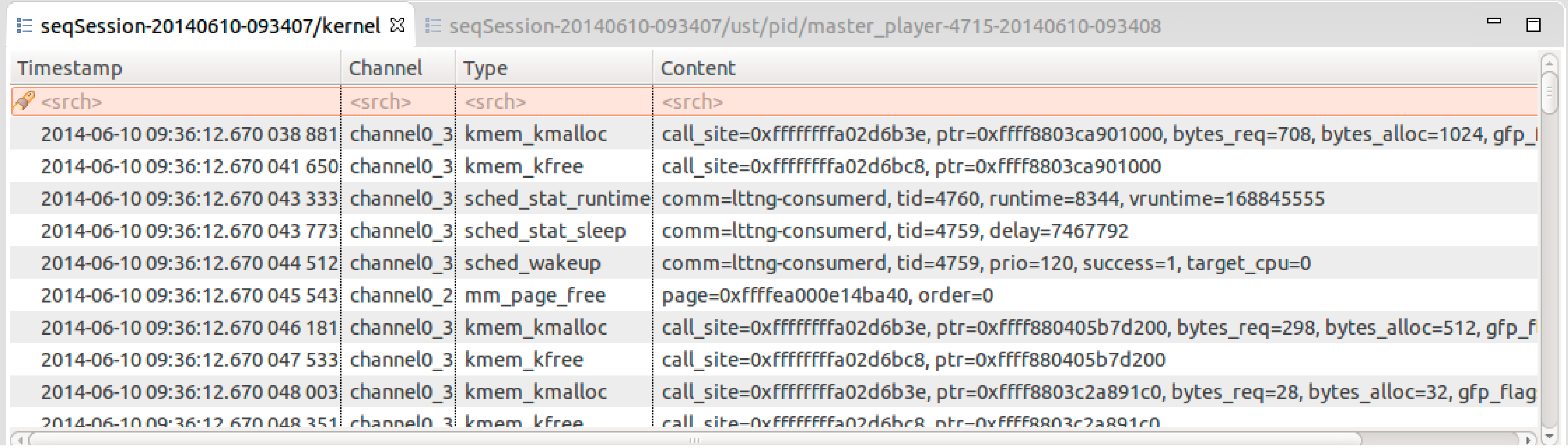
External
Analysis scripts
(JSON output)

DATA FLOW


Trace Type selection
Trace
CTF
Binary
Text
XML
Parser
CTF built-in
Java plug-in
Custom parser
Stream of trace events
Event table
Analysis
Built-in
Java plug-in
XML analysis
Persistence
Analysis result
State system
Segment store
Statistics
Built-in
Java Plug-in
Data-driven XML
Views
Time graph
Charts
Tables
User-created
External
Analysis scripts
(JSON output)

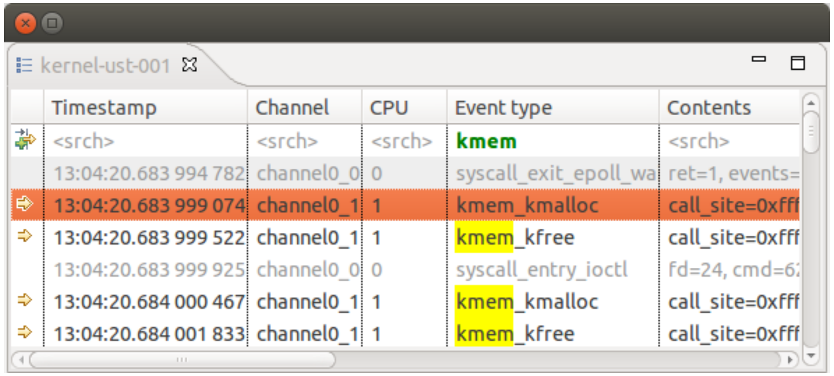
COMMON FEATURES

- Events Table


seqSession-20140610-093407/kernel
seqSession-20140610-093407/ust/pid/master_player-4715-20140610-093408

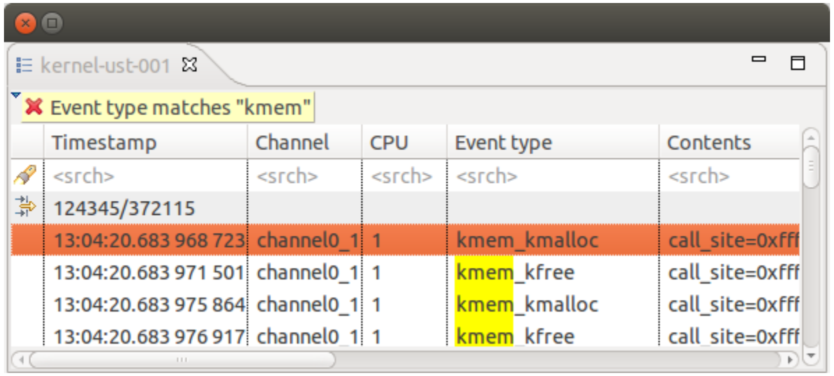
Timestamp	Channel	Type	Content
<srch>	<srch>	<srch>	<srch>
2014-06-10 09:36:12.670 038 881	channel0_3	kmem_kmalloc	call_site=0xffffffffa02d6b3e, ptr=0xffff8803ca901000, bytes_req=708, bytes_alloc=1024, gfp_
2014-06-10 09:36:12.670 041 650	channel0_3	kmem_kfree	call_site=0xffffffffa02d6bc8, ptr=0xffff8803ca901000
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

COMMON FEATURES



kernel-ust-001

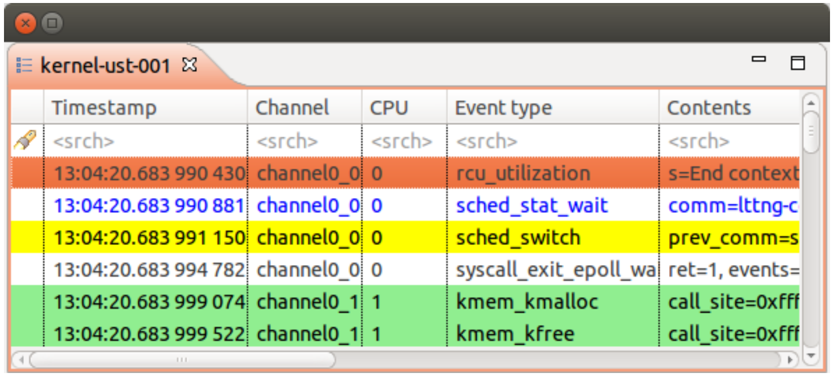
Timestamp	Channel	CPU	Event type	Contents
<srch>	<srch>	<srch>	**kmem**	<srch>
13:04:20.683 994 782	channel0_0	0	syscall_exit_epoll_wa	ret=1, events=
13:04:20.683 999 074	channel0_1	1	kmem_kmalloc	call_site=0xfff
13:04:20.683 999 522	channel0_1	1	kmem_kfree	call_site=0xfff
13:04:20.683 999 925	channel0_0	0	syscall_entry_ioctl	fd=24, cmd=6:
13:04:20.684 000 467	channel0_1	1	kmem_kmalloc	call_site=0xfff
13:04:20.684 001 833	channel0_1	1	kmem_kfree	call_site=0xfff



kernel-ust-001

Event type matches "kmem"

Timestamp	Channel	CPU	Event type	Contents
<srch>	<srch>	<srch>	<srch>	<srch>
124345/372115				
13:04:20.683 968 723	channel0_1	1	kmem_kmalloc	call_site=0xfff
13:04:20.683 971 501	channel0_1	1	kmem_kfree	call_site=0xfff
13:04:20.683 975 864	channel0_1	1	kmem_kmalloc	call_site=0xfff
13:04:20.683 976 917	channel0_1	1	kmem_kfree	call_site=0xfff



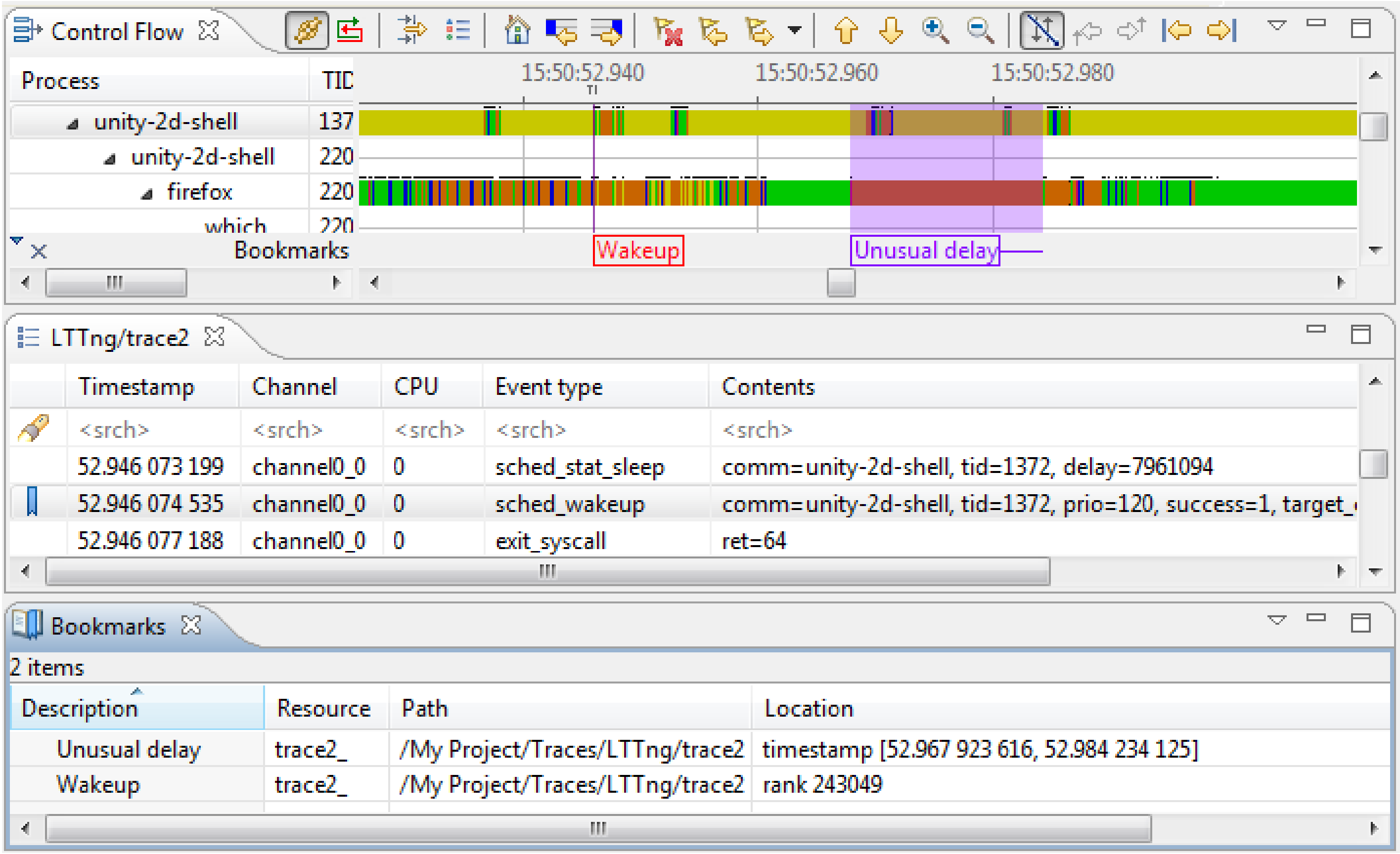
kernel-ust-001

Timestamp	Channel	CPU	Event type	Contents
<srch>	<srch>	<srch>	<srch>	<srch>
13:04:20.683 990 430	channel0_0	0	rcu_utilization	s=End context
13:04:20.683 990 881	channel0_0	0	sched_stat_wait	comm=lttng-c
13:04:20.683 991 150	channel0_0	0	sched_switch	prev_comm=s
13:04:20.683 994 782	channel0_0	0	syscall_exit_epoll_wa	ret=1, events=
13:04:20.683 999 074	channel0_1	1	kmem_kmalloc	call_site=0xfff
13:04:20.683 999 522	channel0_1	1	kmem_kfree	call_site=0xfff

- Searching
- Filtering
- Highlighting

COMMON FEATURES

› Trace annotation (bookmarks) and markers



Process	TID
unity-2d-shell	137
unity-2d-shell	220
firefox	220
which	220

LTTng/trace2

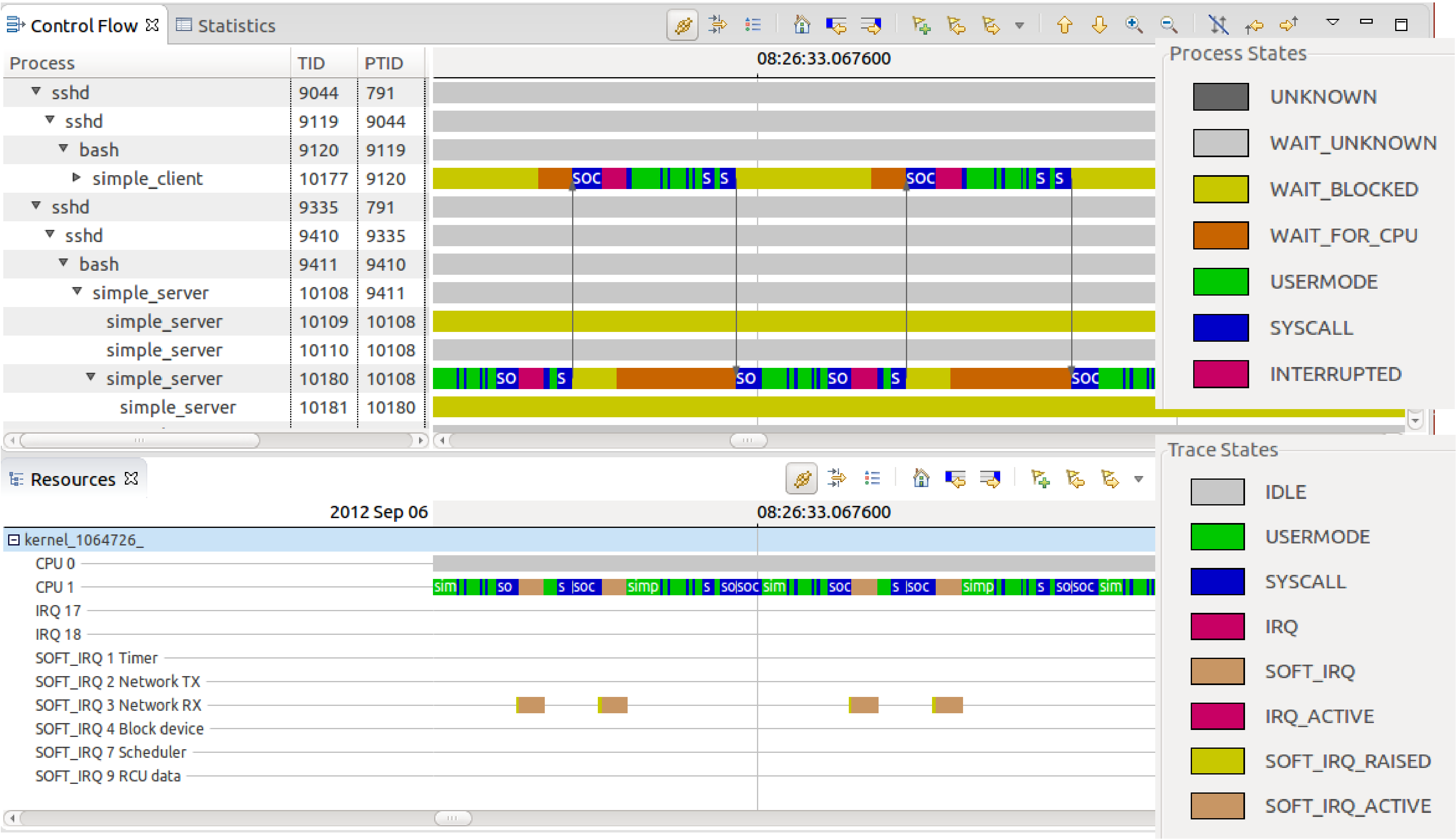
Timestamp	Channel	CPU	Event type	Contents
<srch>	<srch>	<srch>	<srch>	<srch>
52.946 073 199	channel0_0	0	sched_stat_sleep	comm=unity-2d-shell, tid=1372, delay=7961094
52.946 074 535	channel0_0	0	sched_wakeup	comm=unity-2d-shell, tid=1372, prio=120, success=1, target_
52.946 077 188	channel0_0	0	exit_syscall	ret=64

Bookmarks

2 items

Description	Resource	Path	Location
Unusual delay	trace2_	/My Project/Traces/LTTng/trace2	timestamp [52.967 923 616, 52.984 234 125]
Wakeup	trace2_	/My Project/Traces/LTTng/trace2	rank 243049

STATEFUL ANALYSES



Control Flow | Statistics

Process	TID	PTID
▼ sshd	9044	791
▼ sshd	9119	9044
▼ bash	9120	9119
▶ simple_client	10177	9120
▼ sshd	9335	791
▼ sshd	9410	9335
▼ bash	9411	9410
▼ simple_server	10108	9411
simple_server	10109	10108
simple_server	10110	10108
▼ simple_server	10180	10108
simple_server	10181	10180

08:26:33.067600

Process States

■	UNKNOWN
□	WAIT_UNKNOWN
🟨	WAIT_BLOCKED
🟧	WAIT_FOR_CPU
🟩	USERMODE
🟦	SYSCALL
🟥	INTERRUPTED

Resources

2012 Sep 06 | 08:26:33.067600

⊟ kernel_1064726_

- CPU 0
- CPU 1
- IRQ 17
- IRQ 18
- SOFT_IRQ 1 Timer
- SOFT_IRQ 2 Network TX
- SOFT_IRQ 3 Network RX
- SOFT_IRQ 4 Block device
- SOFT_IRQ 7 Scheduler
- SOFT_IRQ 9 RCU data

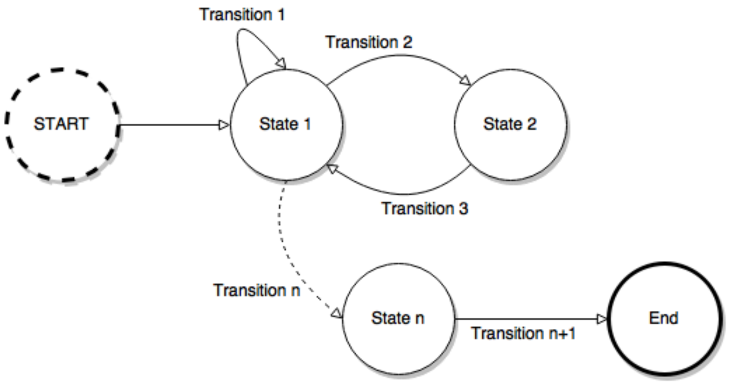
Trace States

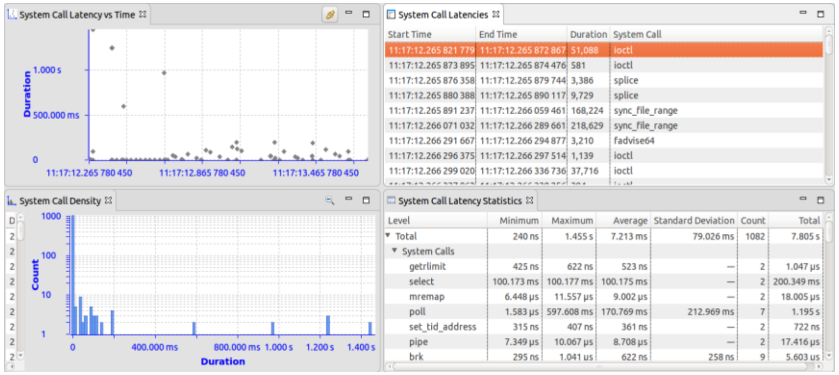
□	IDLE
🟩	USERMODE
🟦	SYSCALL
🟥	IRQ
🟫	SOFT_IRQ
🟥	IRQ_ACTIVE
🟨	SOFT_IRQ_RAISED
🟫	SOFT_IRQ_ACTIVE

XML ANALYSIS & VIEWS

› Pattern analysis
  – Find a sequence of data within a trace

› Customize Trace Compass without adding code
  – Generate state systems
  – Do timing analysis
  – Define specialized views



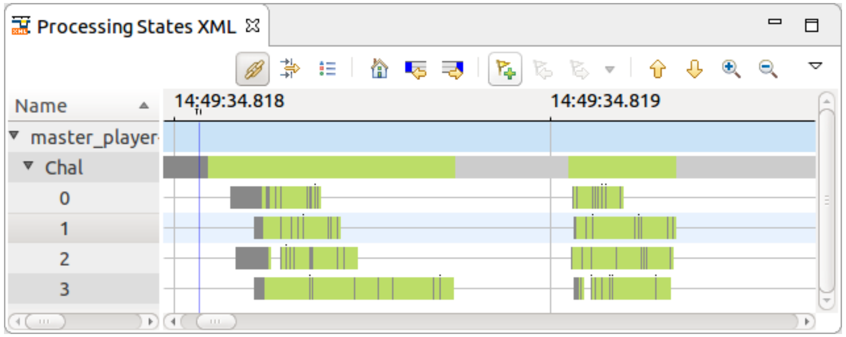


System Call Latencies

Start Time	End Time	Duration	System Call
11:17:12.265 821 779	11:17:12.265 872 867	51,088	ioctl
11:17:12.265 873 895	11:17:12.265 874 476	581	ioctl
11:17:12.265 876 358	11:17:12.265 879 744	3,386	splice
11:17:12.265 880 388	11:17:12.265 890 117	9,729	splice
11:17:12.265 891 237	11:17:12.266 059 461	168,224	sync_file_range
11:17:12.266 071 032	11:17:12.266 289 661	218,629	sync_file_range
11:17:12.266 291 667	11:17:12.266 294 877	3,210	fadvise64
11:17:12.266 296 375	11:17:12.266 297 514	1,139	ioctl
11:17:12.266 299 020	11:17:12.266 336 736	37,716	ioctl

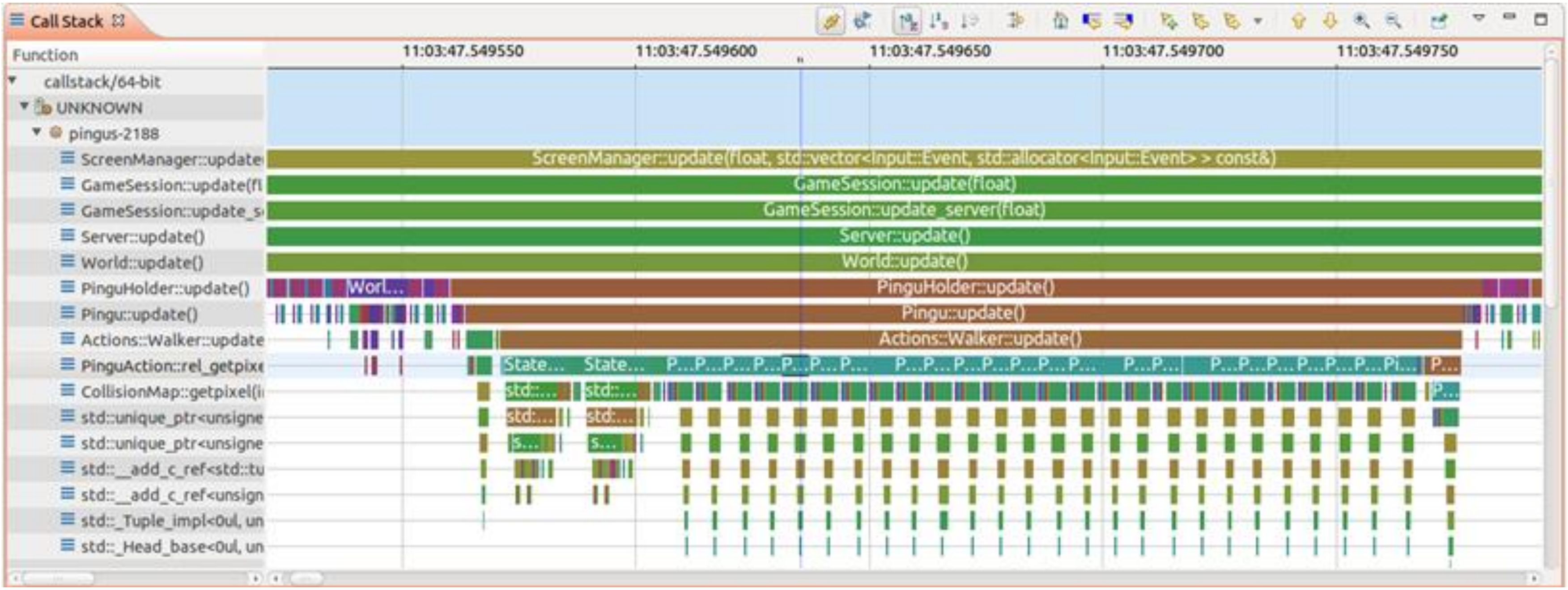
System Call Latency Statistics

Level	Minimum	Maximum	Average	Standard Deviation	Count	Total
▾ Total	240 ns	1.455 s	7.213 ms	79.026 ms	1082	7.805 s
▾ System Calls						
getrlimit	425 ns	622 ns	523 ns	—	2	1.047 µs
select	100.173 ms	100.177 ms	100.175 ms	—	2	200.349 ms
mremap	6.448 µs	11.557 µs	9.002 µs	—	2	18.005 µs
poll	1.583 µs	597.608 ms	170.769 ms	212.969 ms	7	1.195 s
set_tid_address	315 ns	407 ns	361 ns	—	2	722 ns
pipe	7.349 µs	10.067 µs	8.708 µs	—	2	17.416 µs
brk	295 ns	1.041 us	622 ns	258 ns	9	5.603 us



CALL STACK VIEW

- › Extensible view to display of call stacks over time
- › LTTng-UST and finstrument-functions of GCC



TRACE CORRELATION

- Trace Compass can open multiple traces together to view it as one
  - This is called an Experiment
- Useful for
  - Traces coming from multiple nodes
  - Traces from applications written in different languages
  - Different layers (network, etc.)
- Traces can be synchronized by time
  - Manually
  - Automatic algorithm (extensible)

BUILT-IN TRACE TYPES

- Linux Tracing Toolkit - LTTng (UST, Kernel)
- Text & XML Logs (custom parsers)
- Common Trace Format – CTF
  - application, kernel, HW, bare metal, etc.
- Packet Capture
- Best Trace Format - BTF
- GDB Trace Points

TIMING ANALYSIS

- Real-time systems
- We have two metrics to analyse
  - what is the data and when did it come
- Timing is as important as data
- Measure time between a start and end state
  - Simple: Start and end event
  - Often: State Machine to determine start and end
- Represent execution times, latencies, latency chains etc.

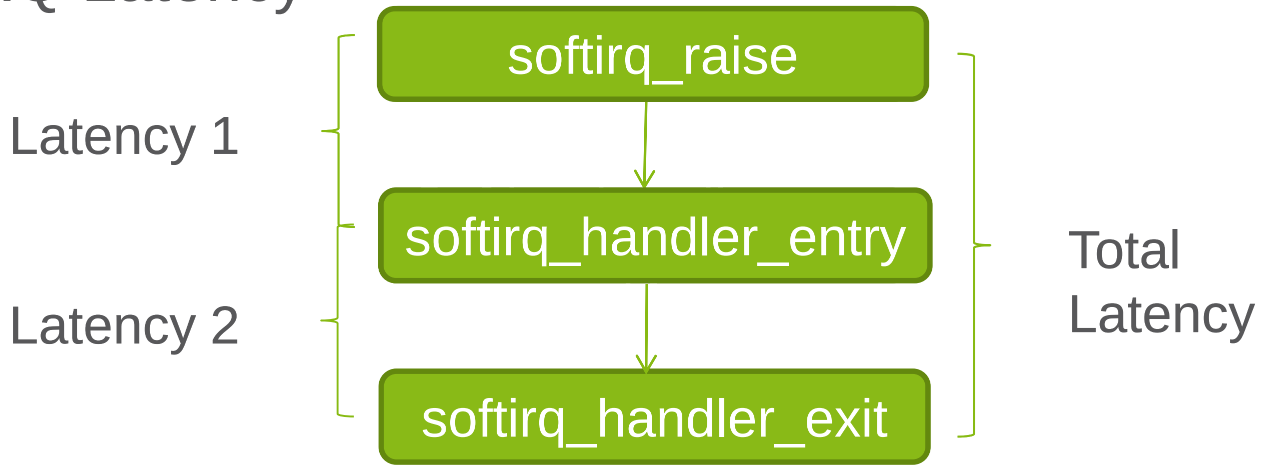
TIMING ANALYSIS

- **Locate timing problems**
  - **Missed deadlines**
  - **Potential missed deadline (find problem before it occurs)**
- Analyze timing problems
  - Find root cause and solution
  - Solve difficult to debug sporadic problems

EXAMPLE

- Soft IRQ Latency

Parameter: CPU ID, IRQ #


softirq_raise

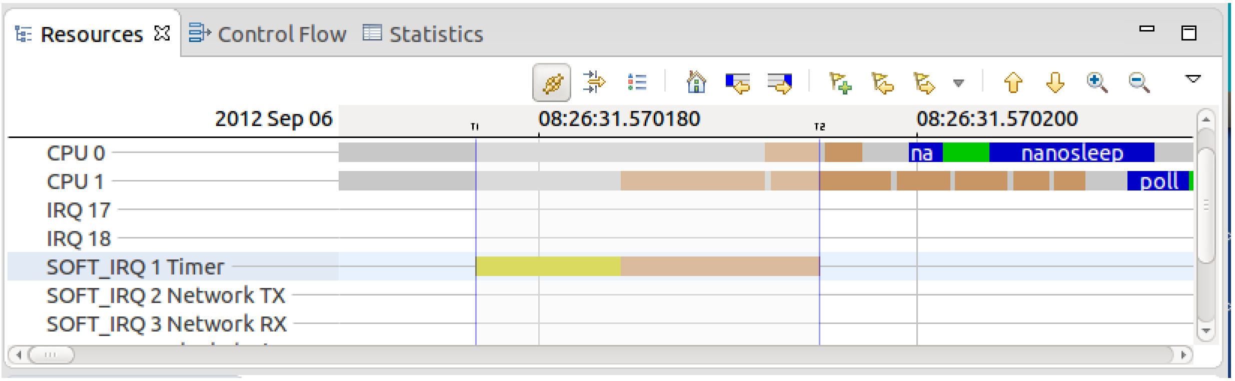
softirq_handler_entry

softirq_handler_exit

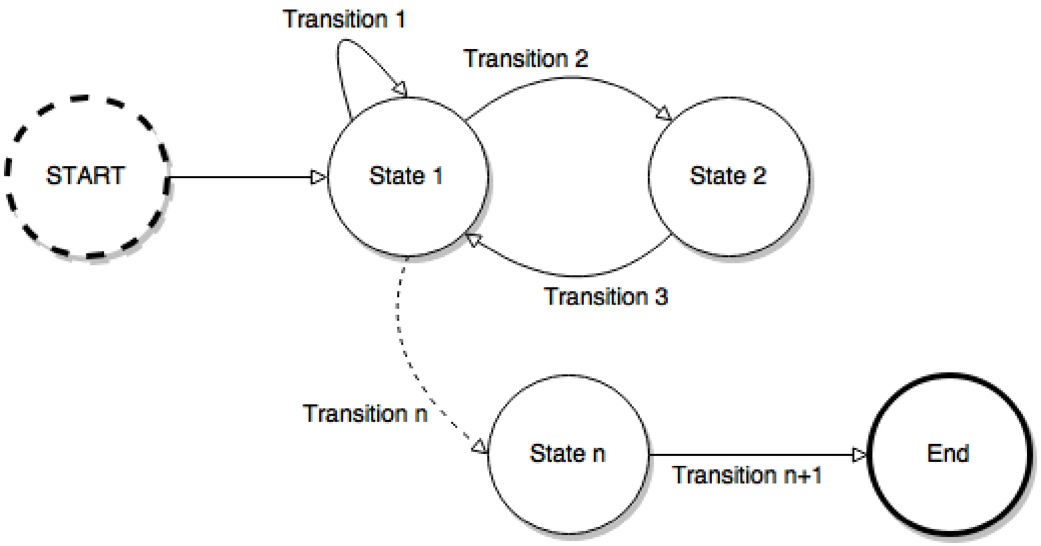
Latency 1

Latency 2

Total Latency



GENERALIZATION



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

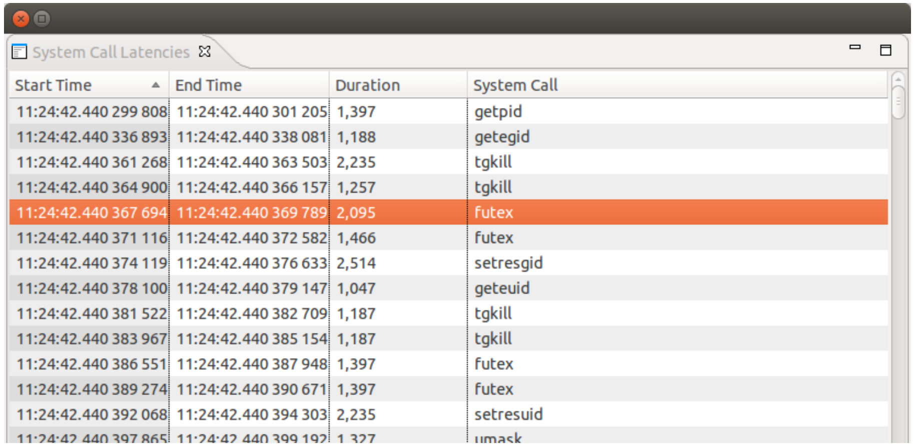
YOUR TIMING ANALYSIS

- Define a state machine for timing analysis
  - Implementation in Java as Trace Compass extension
  - Data-driven pattern matching (in XML)
    - Defining timing analyses on-the-fly
- Store in a built-in segment store
- Visualize data in various supplied views

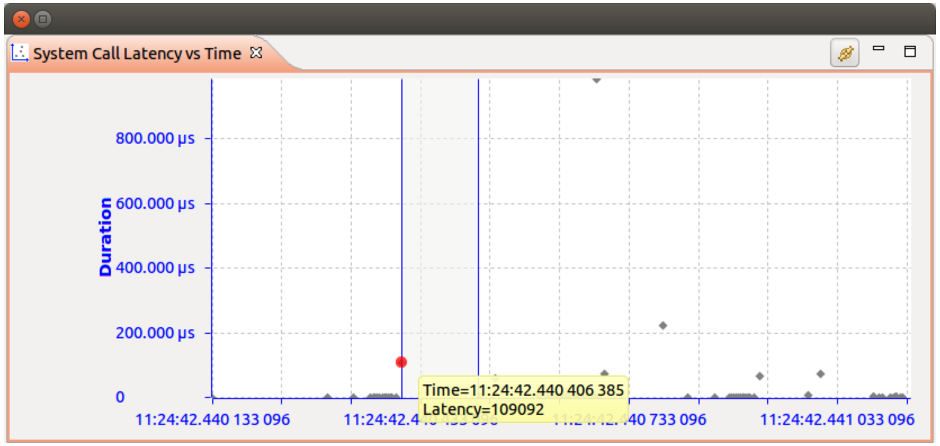
VISUALIZATION

- Table
  - Get raw data
  - Explore data
  - Sorting, highlighting, filtering

- Scatter Chart
  - Latency vs Time
  - Have a big picture of the current range



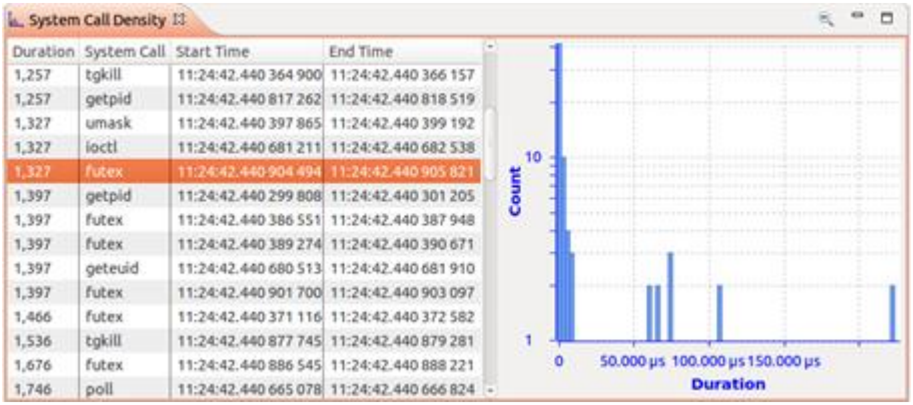
Start Time	End Time	Duration	System Call
11:24:42.440 299 808	11:24:42.440 301 205	1,397	getpid
11:24:42.440 336 893	11:24:42.440 338 081	1,188	getegid
11:24:42.440 361 268	11:24:42.440 363 503	2,235	tgkill
11:24:42.440 364 900	11:24:42.440 366 157	1,257	tgkill
11:24:42.440 367 694	11:24:42.440 369 789	2,095	futex
11:24:42.440 371 116	11:24:42.440 372 582	1,466	futex
11:24:42.440 374 119	11:24:42.440 376 633	2,514	setresgid
11:24:42.440 378 100	11:24:42.440 379 147	1,047	geteuid
11:24:42.440 381 522	11:24:42.440 382 709	1,187	tgkill
11:24:42.440 383 967	11:24:42.440 385 154	1,187	tgkill
11:24:42.440 386 551	11:24:42.440 387 948	1,397	futex
11:24:42.440 389 274	11:24:42.440 390 671	1,397	futex
11:24:42.440 392 068	11:24:42.440 394 303	2,235	setresuid
11:24:42.440 397 865	11:24:42.440 399 192	1,327	umask



VISUALIZATION

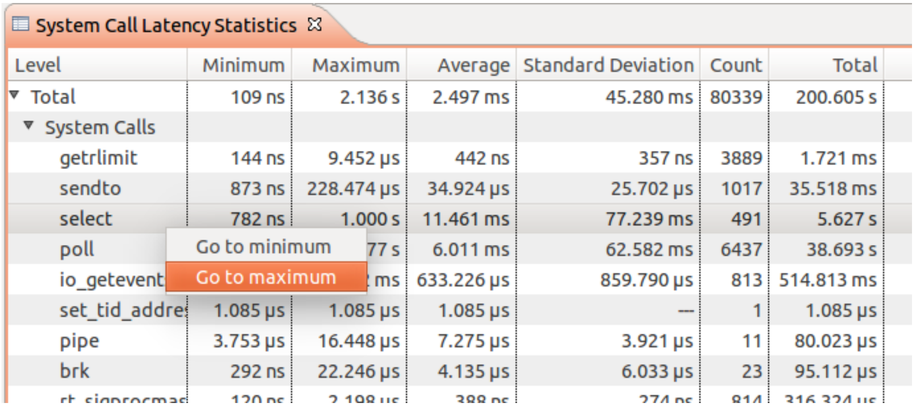
- Distribution Chart
  - Find outliers and modes easily

- Statistics
  - Min, max, average etc.
  - Find worst offenders
  - Find worst possible offender combination



System Call Density

Duration	System Call	Start Time	End Time
1,257	tgkill	11:24:42.440 364 900	11:24:42.440 366 157
1,257	getpid	11:24:42.440 817 262	11:24:42.440 818 519
1,327	umask	11:24:42.440 397 865	11:24:42.440 399 192
1,327	ioctl	11:24:42.440 681 211	11:24:42.440 682 538
1,327	futex	11:24:42.440 904 494	11:24:42.440 905 821
1,397	getpid	11:24:42.440 299 808	11:24:42.440 301 205
1,397	futex	11:24:42.440 386 551	11:24:42.440 387 948
1,397	futex	11:24:42.440 389 274	11:24:42.440 390 671
1,397	geteuid	11:24:42.440 680 513	11:24:42.440 681 910
1,397	futex	11:24:42.440 901 700	11:24:42.440 903 097
1,466	futex	11:24:42.440 371 116	11:24:42.440 372 582
1,536	tgkill	11:24:42.440 877 745	11:24:42.440 879 281
1,676	futex	11:24:42.440 886 545	11:24:42.440 888 221
1,746	poll	11:24:42.440 665 078	11:24:42.440 666 824



System Call Latency Statistics

Level	Minimum	Maximum	Average	Standard Deviation	Count	Total
▾ Total	109 ns	2.136 s	2.497 ms	45.280 ms	80339	200.605 s
▾ System Calls						
getrlimit	144 ns	9.452 µs	442 ns	357 ns	3889	1.721 ms
sendto	873 ns	228.474 µs	34.924 µs	25.702 µs	1017	35.518 ms
select	782 ns	1.000 s	11.461 ms	77.239 ms	491	5.627 s
poll		77 s	6.011 ms	62.582 ms	6437	38.693 s
io_getevent		ms	633.226 µs	859.790 µs	813	514.813 ms
set_tid_addre	1.085 µs	1.085 µs	1.085 µs	---	1	1.085 µs
pipe	3.753 µs	16.448 µs	7.275 µs	3.921 µs	11	80.023 µs
brk	292 ns	22.246 µs	4.135 µs	6.033 µs	23	95.112 µs
rt_sigprocmas	120 ns	2.198 µs	388 ns	274 ns	814	316.324 µs

TIMING ANALYSIS

- Locate timing problems
  - Missed deadlines
  - Potential missed deadline (find problem before it occurs)
- **Analyze timing problems**
  - **Find root cause and solution**
  - **Solve difficult to debug sporadic problems**

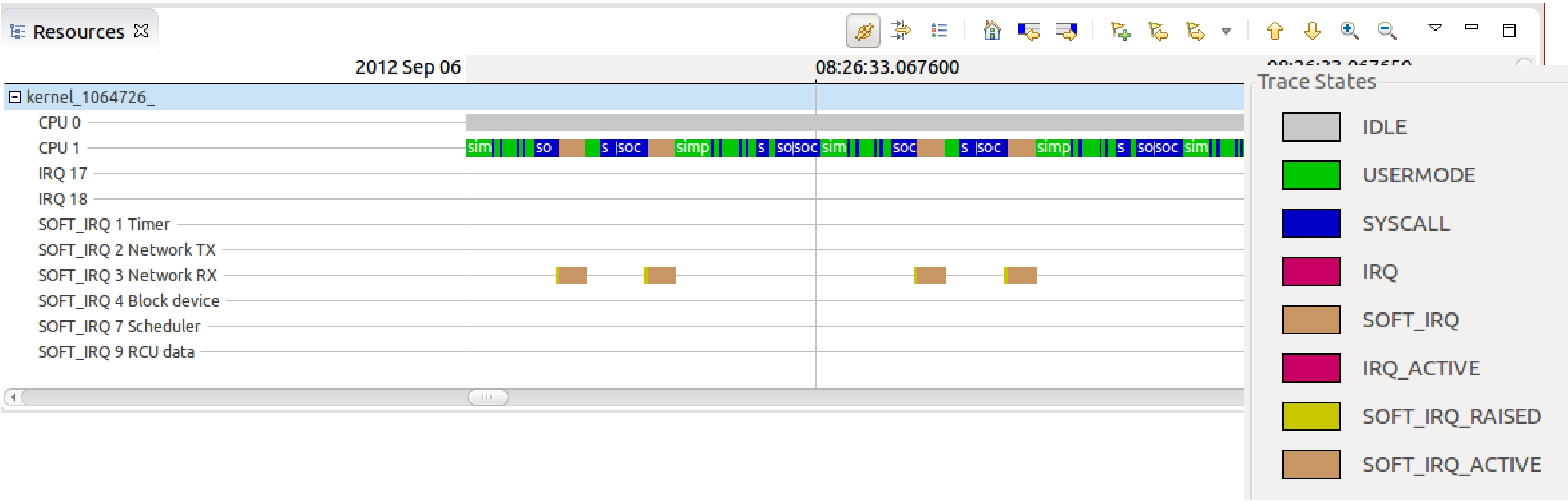
EXAMPLE ROOT CAUSES

- System overload
- System misconfiguration (e.g. wrong priorities of tasks)
- Priority inversion
  - Lower priority task is blocking higher priority task (indirectly)
- Blocked threads, starvation, deadlock
- Slow code

RESOURCES VIEW

› Displays resources states (color-coded) over time

  – CPUs, IRQs, SoftIRQs



Resources

2012 Sep 06

08:26:33.067600

08:26:33.067650

⊟ kernel_1064726_

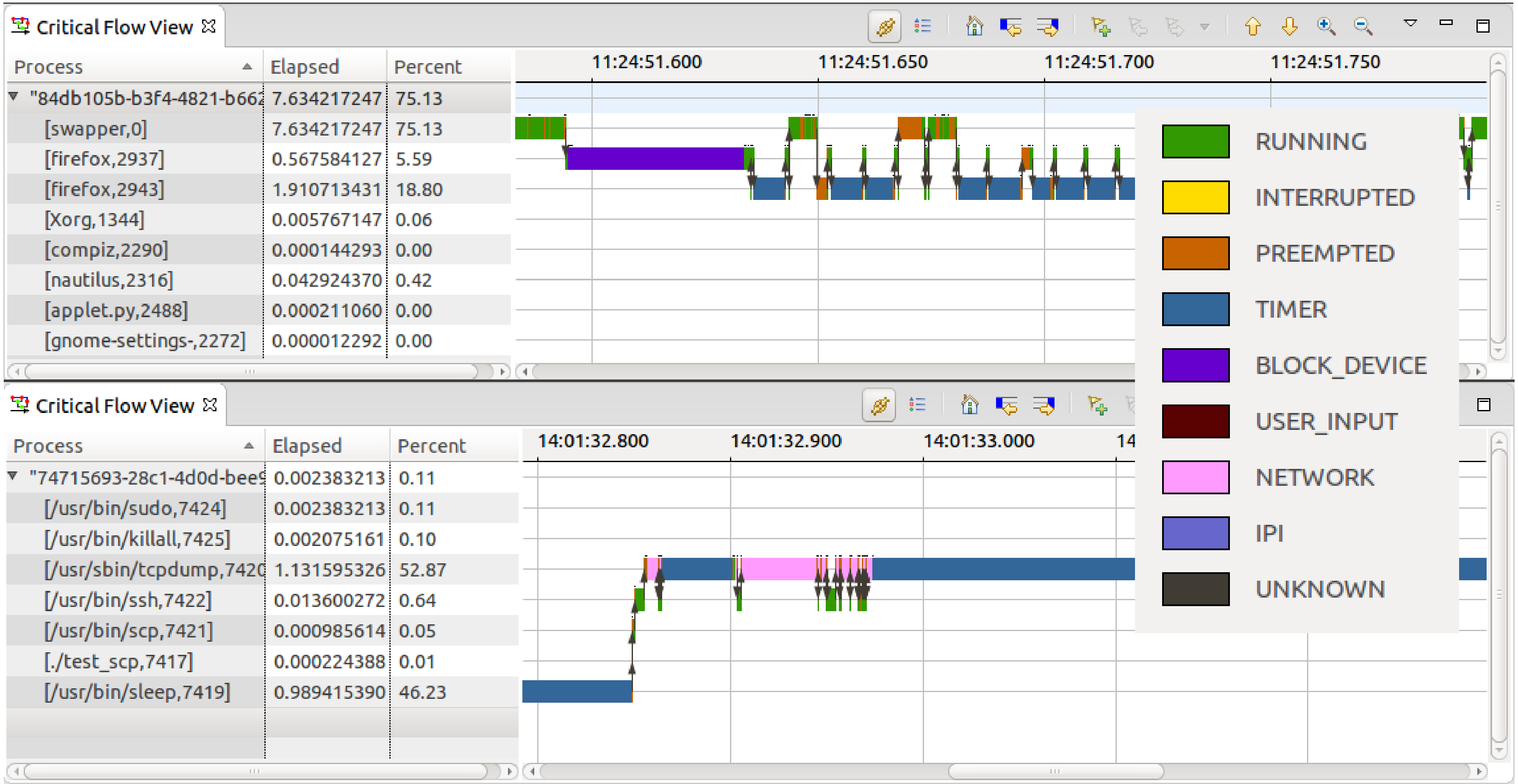
- CPU 0
- CPU 1
- IRQ 17
- IRQ 18
- SOFT_IRQ 1 Timer
- SOFT_IRQ 2 Network TX
- SOFT_IRQ 3 Network RX
- SOFT_IRQ 4 Block device
- SOFT_IRQ 7 Scheduler
- SOFT_IRQ 9 RCU data

Trace States

⬜	IDLE
🟩	USERMODE
🟦	SYSCALL
🟥	IRQ
🟫	SOFT_IRQ
🟥	IRQ_ACTIVE
🟨	SOFT_IRQ_RAISED
🟫	SOFT_IRQ_ACTIVE

CRITICAL PATH

- › Displays of system wait chains for given process

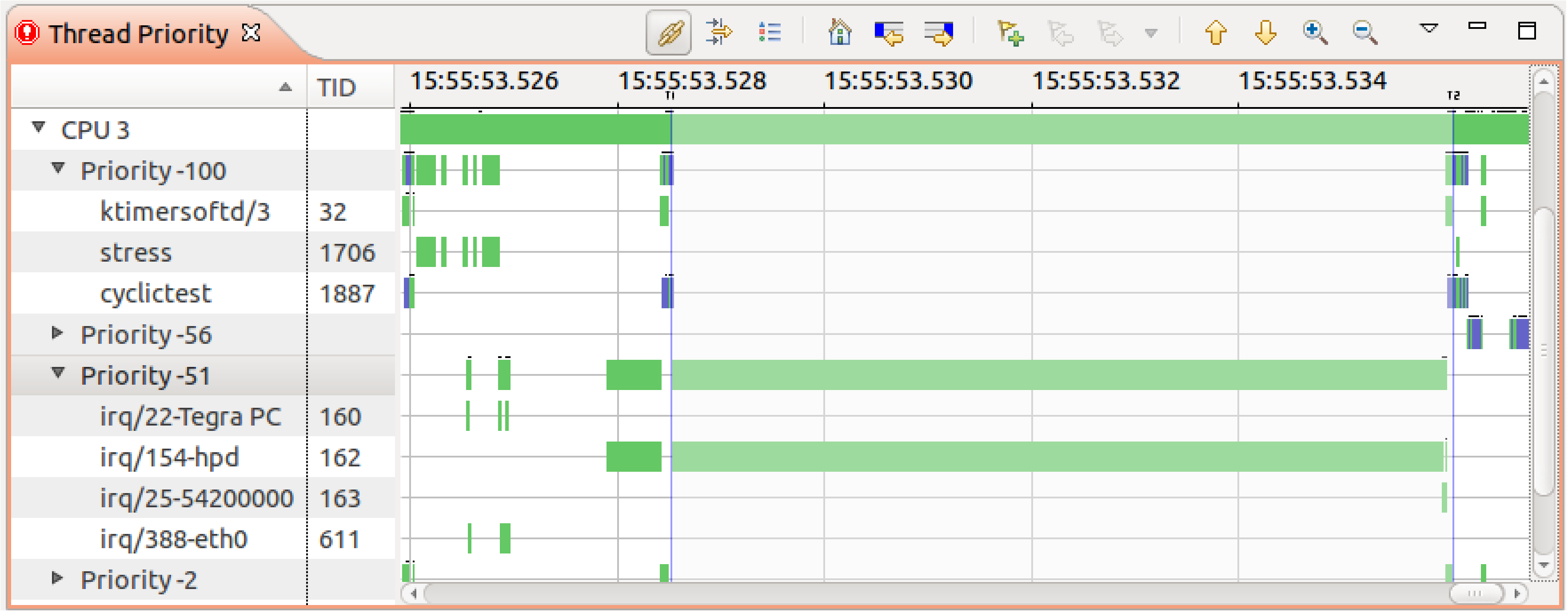


Process	Elapsed	Percent
▾ "84db105b-b3f4-4821-b662	7.634217247	75.13
[swapper,0]	7.634217247	75.13
[firefox,2937]	0.567584127	5.59
[firefox,2943]	1.910713431	18.80
[Xorg,1344]	0.005767147	0.06
[compiz,2290]	0.000144293	0.00
[nautilus,2316]	0.042924370	0.42
[applet.py,2488]	0.000211060	0.00
[gnome-settings-,2272]	0.000012292	0.00

Process	Elapsed	Percent
▾ "74715693-28c1-4d0d-bee9	0.002383213	0.11
[/usr/bin/sudo,7424]	0.002383213	0.11
[/usr/bin/killall,7425]	0.002075161	0.10
[/usr/sbin/tcpdump,7420	1.131595326	52.87
[/usr/bin/ssh,7422]	0.013600272	0.64
[/usr/bin/scp,7421]	0.000985614	0.05
[./test_scp,7417]	0.000224388	0.01
[/usr/bin/sleep,7419]	0.989415390	46.23

PRIORITY VIEW

- Group processes per CPU and priority
- Quickly find priority inversion or misconfigured task priorities
- Note: View not mainlined yet – Prototype!

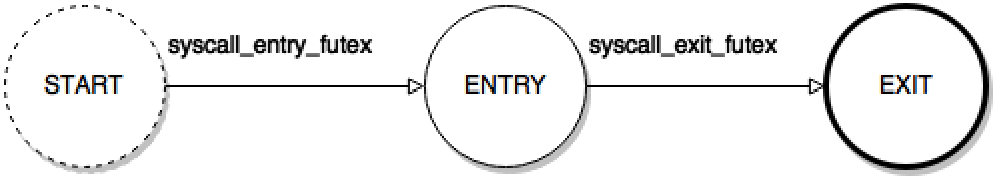


Thread Priority

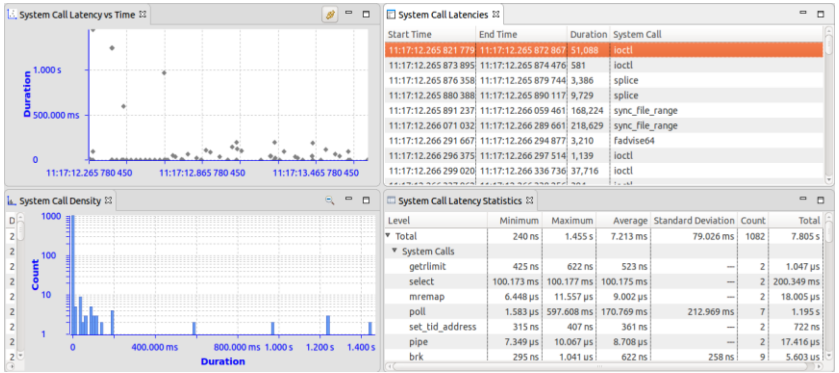
	TID
CPU 3	
Priority -100	
ktimersoftd/3	32
stress	1706
cyclictest	1887
Priority -56	
Priority -51	
irq/22-Tegra PC	160
irq/154-hpd	162
irq/25-54200000	163
irq/388-eth0	611
Priority -2	

FUTEX ANALYSIS

- Find contention at the Kernel level using LTTng



- Realized as XML pattern analysis



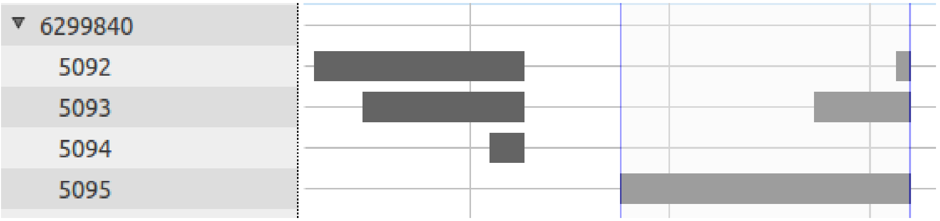
Start Time	End Time	Duration	System Call
11:17:12.265 821 779	11:17:12.265 872 867	51,088	ioctl
11:17:12.265 873 895	11:17:12.265 874 476	581	ioctl
11:17:12.265 876 358	11:17:12.265 879 744	3,386	splice
11:17:12.265 880 388	11:17:12.265 890 117	9,729	splice
11:17:12.265 891 237	11:17:12.266 059 461	168,224	sync_file_range
11:17:12.266 071 032	11:17:12.266 289 661	218,629	sync_file_range
11:17:12.266 291 667	11:17:12.266 294 877	3,210	fadvise64
11:17:12.266 296 375	11:17:12.266 297 514	1,139	ioctl
11:17:12.266 299 020	11:17:12.266 336 736	37,716	ioctl

Level	Minimum	Maximum	Average	Standard Deviation	Count	Total
▾ Total	240 ns	1.455 s	7.213 ms	79.026 ms	1082	7.805 s
▾ System Calls						
getrlimit	425 ns	622 ns	523 ns	—	2	1.047 µs
select	100.173 ms	100.177 ms	100.175 ms	—	2	200.349 ms
mremap	6.448 µs	11.557 µs	9.002 µs	—	2	18.005 µs
poll	1.583 µs	597.608 ms	170.769 ms	212.969 ms	7	1.195 s
set_tid_address	315 ns	407 ns	361 ns	—	2	722 ns
pipe	7.349 µs	10.067 µs	8.708 µs	—	2	17.416 µs
brk	295 ns	1.041 µs	622 ns	258 ns	9	5.603 µs

- Count of simultaneous waits

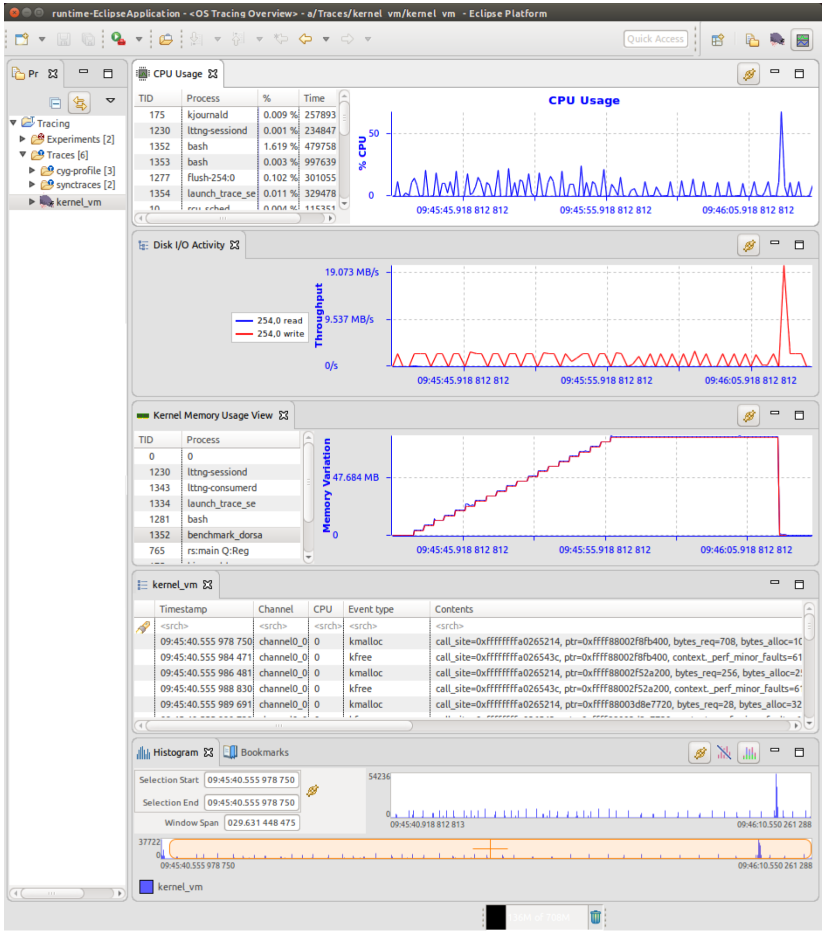
- Show all in timing analysis views

- Uaddr vs Thread Gantt chart



OS TRACING OVERVIEW

- Overloaded resources
- CPU, Memory and IO Usage
- Counter-intuitive example, CPU usage too low:
  - Kernel memory usage is rising
    - Find the offending process
  - IO usage is high
    - Maybe it’s swaps
  - Too many seeks?
    - Low IO, low CPU, low memory usage and low bandwidth


runtime-EclipseApplication - <OS Tracing Overview> - a/Traces/kernel vm/kernel vm - Eclipse Platform

TID	Process	%	Time
175	kjournald	0.009 %	257893
1230	lttng-sessiond	0.001 %	234847
1352	bash	1.619 %	479758
1353	bash	0.003 %	997639
1277	flush-254:0	0.102 %	301055
1354	launch_trace_se	0.011 %	329478

TID	Process
0	0
1230	lttng-sessiond
1343	lttng-consumerd
1334	launch_trace_se
1281	bash
1352	benchmark_dorsa
765	rs:main Q:Reg

Timestamp	Channel	CPU	Event type	Contents
<srch>	<srch>	<srch>	<srch>	<srch>
09:45:40.555 978 750	channel0_0	0	kmalloc	call_site=0xffffffffa0265214, ptr=0xffff88002f8fb400, bytes_req=708, bytes_alloc=10
09:45:40.555 984 471	channel0_0	0	kfree	call_site=0xffffffffa026543c, ptr=0xffff88002f8fb400, context._perf_minor_faults=61
09:45:40.555 986 481	channel0_0	0	kmalloc	call_site=0xffffffffa0265214, ptr=0xffff88002f52a200, bytes_req=256, bytes_alloc=2
09:45:40.555 988 830	channel0_0	0	kfree	call_site=0xffffffffa026543c, ptr=0xffff88002f52a200, context._perf_minor_faults=6
09:45:40.555 989 691	channel0_0	0	kmalloc	call_site=0xffffffffa0265214, ptr=0xffff88003d8e7720, bytes_req=28, bytes_alloc=32

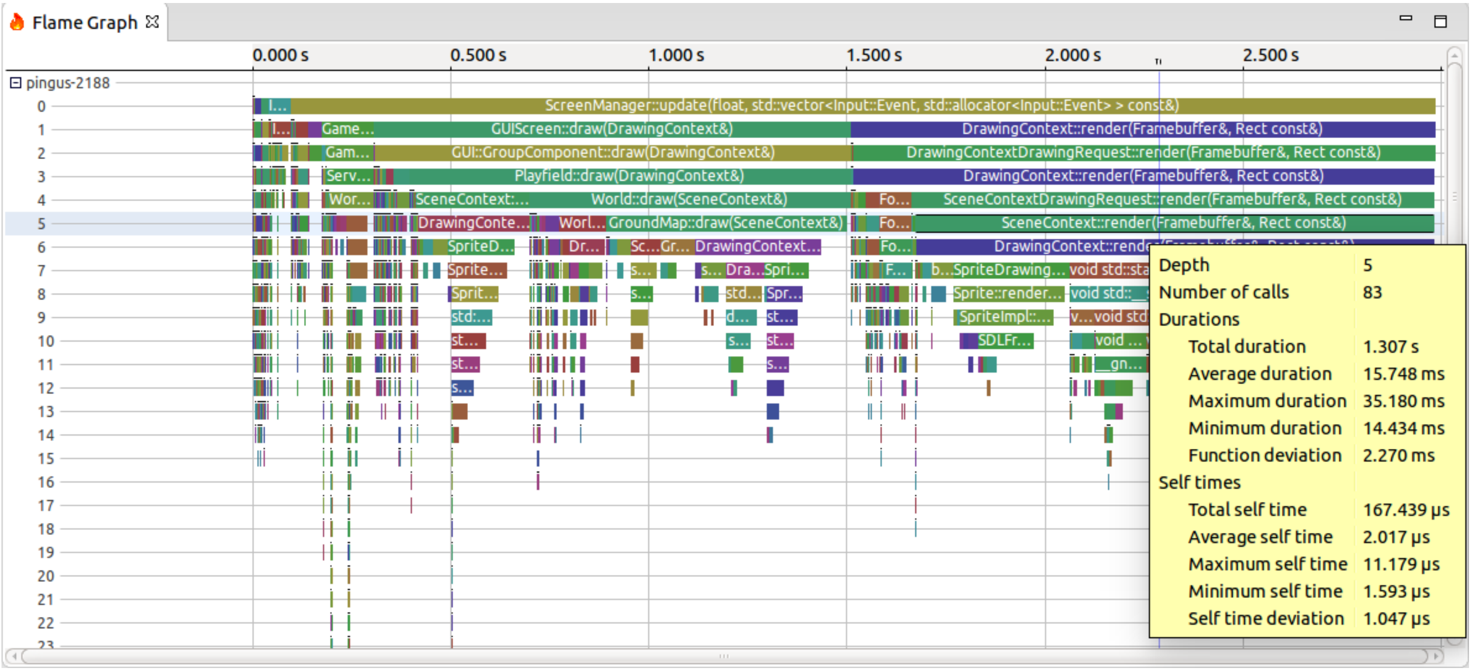
Selection Start 09:45:40.555 978 750

Selection End 09:45:40.555 978 750

Window Span 029.631 448 475

FLAME GRAPH VIEW

- Aggregation of function durations per call stack
- Highlights most time consuming execution path
- Find functions for performance optimization


Flame Graph
0.000 s
0.500 s
1.000 s
1.500 s
2.000 s
2.500 s
pingus-2188
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
ScreenManager::update(float, std::vector<Input::Event, std::allocator<Input::Event> > const&)
GUIScreen::draw(DrawingContext&)
DrawingContext::render(Framebuffer&, Rect const&)
GUI::GroupComponent::draw(DrawingContext&)
DrawingContextDrawingRequest::render(Framebuffer&, Rect const&)
Playfield::draw(DrawingContext&)
DrawingContext::render(Framebuffer&, Rect const&)
SceneContext:...
World::draw(SceneContext&)
SceneContextDrawingRequest::render(Framebuffer&, Rect const&)
DrawingConte...
Worl...
GroundMap::draw(SceneContext&)
SceneContext::render(Framebuffer&, Rect const&)
SpriteD...
DrawingContext...
b...SpriteDrawing...
Sprite::render...
SpriteImpl:...
SDLFr...

Depth	5
Number of calls	83
Durations	
Total duration	1.307 s
Average duration	15.748 ms
Maximum duration	35.180 ms
Minimum duration	14.434 ms
Function deviation	2.270 ms
Self times	
Total self time	167.439 µs
Average self time	2.017 µs
Maximum self time	11.179 µs
Minimum self time	1.593 µs
Self time deviation	1.047 µs

FUTURE DEVELOPMENT

- User-configurable periodic markers
- Custom charts
- Enhanced call graph analysis and views
- Call stack views using data-driven analysis
- Pin & clone of views
- Time based import of traces/experiments
- Scalable segment store
- Enhanced searching, filtering and highlighting in Gantt charts
- Data-driven analysis and view enhancements
- Cropping of traces
- Priority view
- …

REFERENCES

- Project pages
  - http://tracecompass.org
  - http://projects.eclipse.org/projects/tools.tracecompass

- Documentation
  - Trace Compass User Guide
  - Trace Compass Developer Guide

REFERENCES

- Linux Tracing Toolkit (LTTng)
  - http://lttng.org/
- Diagnostic and Monitoring Working Group
  - http://diamon.org/
- Common Trace Format (CTF)
  - http://diamon.org/ctf/
- Trace Research Project
  - http://hsdm.dorsal.polymtl.ca/

CONTACTS

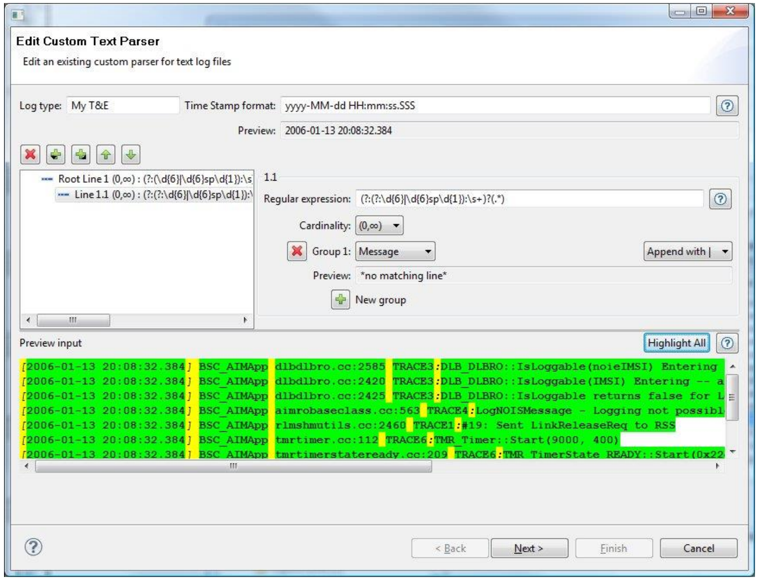
- Bernd.Hufmann@ericsson.com
- Mailing list
  - tracecompass-dev@eclipse.org
- IRC
  - oftc.net #tracecompass
- Mattermost
  - https://mattermost-test.eclipse.org/eclipse/channels/trace-compass

Q&A

ERICSSON

CUSTOM PARSERS

- Custom Text and XML Parsers
  - Line based parser with regex
  - XML based extracting data from XML elements and their attributes



Edit Custom Text Parser

Edit an existing custom parser for text log files

Log type: My T&E

Time Stamp format: yyyy-MM-dd HH:mm:ss.SSS

Preview: 2006-01-13 20:08:32.384

Root Line 1 (0,∞) : (?:(\d{6}|\d{6}sp\d{1}):\s

Line 1.1 (0,∞) : (?:(?:\d{6}|\d{6}sp\d{1}):\

1.1

Regular expression: (?:(?:\d{6}|\d{6}sp\d{1}):\s+)?(.*)

Cardinality: (0,∞)

Group 1: Message

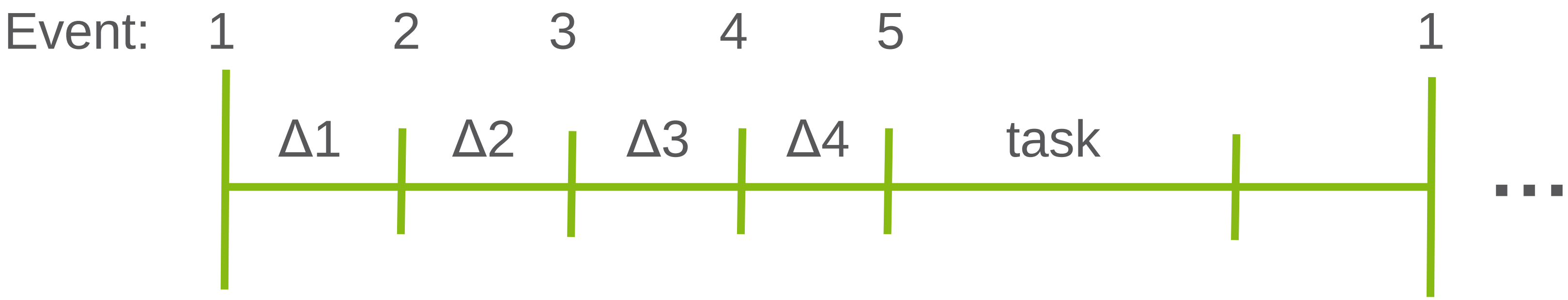
Append with |

Preview: *no matching line*

```
[2006-01-13 20:08:32.384] BSC_AIMApp dlbdlbro.cc:2585 TRACE3:DLB_DLBRO::IsLoggable(noieIMSI) Entering
[2006-01-13 20:08:32.384] BSC_AIMApp dlbdlbro.cc:2420 TRACE3:DLB_DLBRO::IsLoggable(IMSI) Entering -- a
[2006-01-13 20:08:32.384] BSC_AIMApp dlbdlbro.cc:2425 TRACE3:DLB_DLBRO::IsLoggable returns false for L
[2006-01-13 20:08:32.384] BSC_AIMApp aimrobaseclass.cc:563 TRACE4:LogNOISMessage - Logging not possibl
[2006-01-13 20:08:32.384] BSC_AIMApp rlmshmutils.cc:2460 TRACE1:#19: Sent LinkReleaseReq to RSS
[2006-01-13 20:08:32.384] BSC_AIMApp tmrtimer.cc:112 TRACE6:TMR_Timer::Start(9000, 400)
[2006-01-13 20:08:32.384] BSC_AIMApp tmrtimerstateready.cc:209 TRACE6:TMR_TimerState_READY::Start(0x22
```

EXAMPLE

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



- Latency = Δ1+ Δ2 + Δ3 + Δ4

  - Event 1: Timer expires
  - Event 2: Interrupt begins executing
  - Event 3: Interrupt handler marks the task to react
  - Event 4: Linux scheduler switches to the task
  - Event 5: Application task begins executing