Problem detection in real-time systems by trace analysis

Mathieu Côté

Laboratoire DORSAL

mathieu.cote@polymtl.ca

POLYTECHNIQUE
MONTRÉAL

13 mai 2015

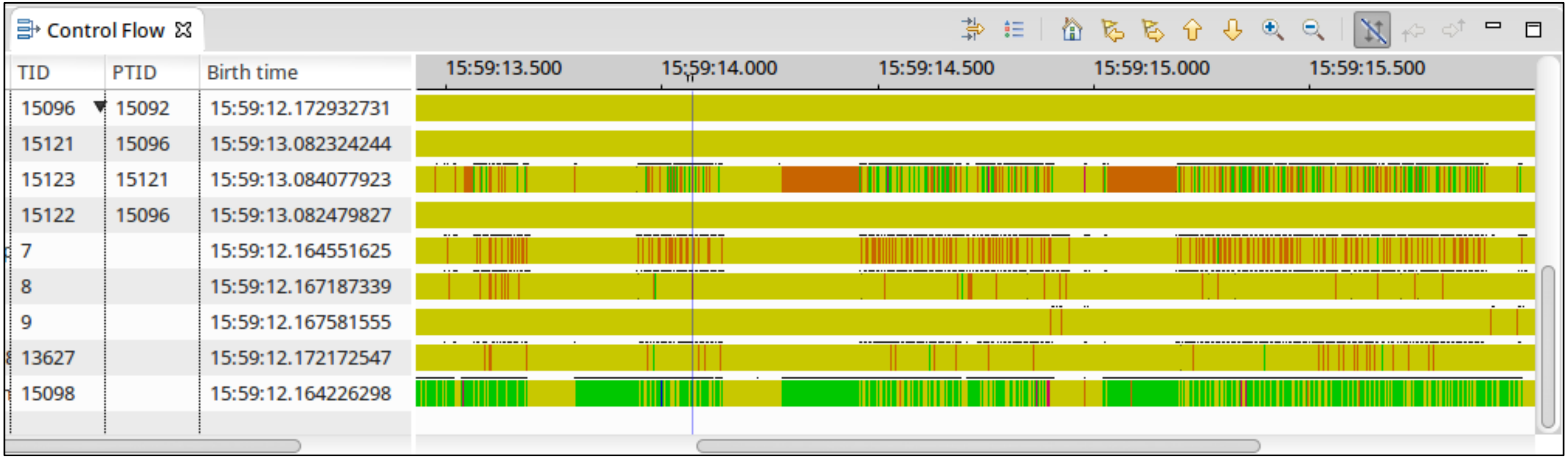
Outline

- Introduction
- Literature review
- Modeling
- Views
- Results
- Conclusion

Introduction : problematic

Introduction
Literature
Modeling
Views
Results
Conclusion

Music player trace in Trace Compass



TID	PTID	Birth time
15096	15092	15:59:12.172932731
15121	15096	15:59:13.082324244
15123	15121	15:59:13.084077923
15122	15096	15:59:13.082479827
7		15:59:12.164551625
8		15:59:12.167187339
9		15:59:12.167581555
13627		15:59:12.172172547
15098		15:59:12.164226298

Figure 1 : Multiple executions of an audio player

Introduction : problematic

Introduction
Literature
Modeling
Views
Results
Conclusion

Advantages of tracing real-time systems

- Low **overhead**
- Low **jitter**
- Access to **specific** information (priority, scheduling policy, etc.)

What is missing?

- Real-time **specific** user tools
- Show **useful** data

Introduction : goals

1. Develop a **model** to define real-time task **executions** in a trace
2. Identify common **problems** in real-time systems and useful **information** to analyze them
3. Develop a method to analyze the **trace segment** corresponding to an execution to identify if the execution presents a **problem**

Introduction : definition

- Real-time task : execution time, deadline, period (optional)
- Execution : periodic, sporadic
- Hard/soft real-time

PREEMPT_RT

- Priority inheritance for mutex in kernel
- Reduce non-preemptive sections in kernel

Scheduling policies

- Normal
  - SCHED_OTHER : standard
  - SCHED_BATCH
  - SCHED_IDLE
- Real-time
  - SCHED_FIFO
  - SCHED_RR : with time quantum
  - SCHED_DEADLINE : Global Earliest Deadline First, highest user controllable priority

Scheduling policies

- SCHED_FIFO and SCHED_RR
  - A deadline can be missed even if there was a valid scheduling to respect all deadlines
- SCHED_DEADLINE
  - No deadline will be missed if there is a valid scheduling

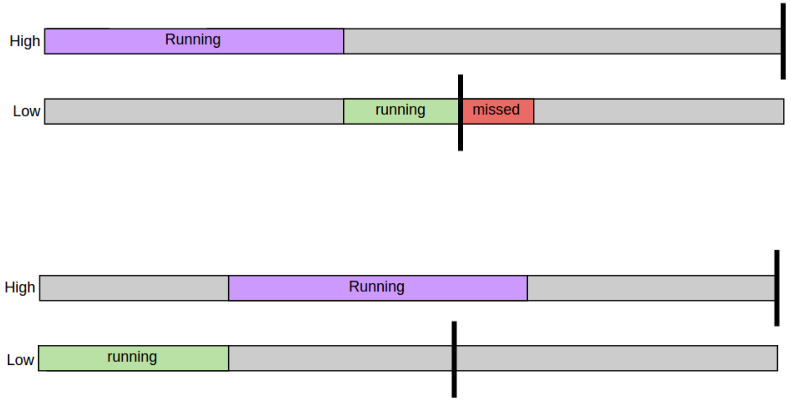


Figure 2 : Deadline missed

Scheduling policies

- SCHED_FIFO and SCHED_RR
  - The highest priority task will always execute if it is able to
- SCHED_DEADLINE
  - If there is a missed deadline, it can be on a highest priority task (for the user, because there is no priority set)

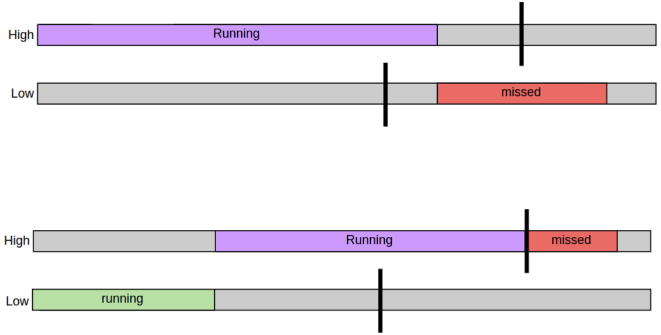


Figure 3 : Highest priority

Priority inversion

The high priority task is blocked by the low priority task that is preempted because the medium priority task is running.

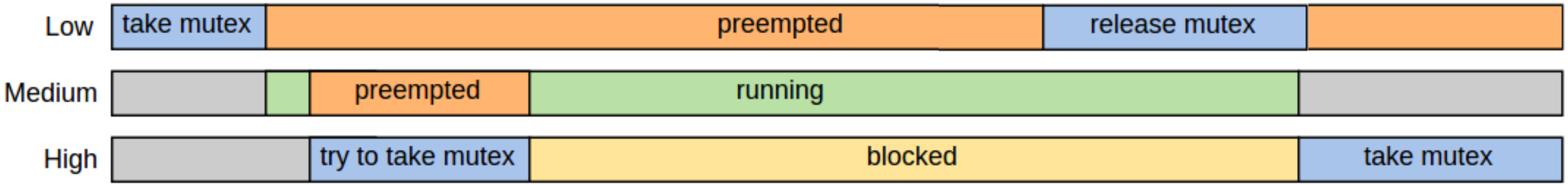


Figure 4 : Priority inversion

Priority inversion

Priority ceiling protocol

- Better if the high priority task accesses the resource more often than the low priority task, because it is faster and has fewer context switches, but it can give an unnecessary high priority to the lower task

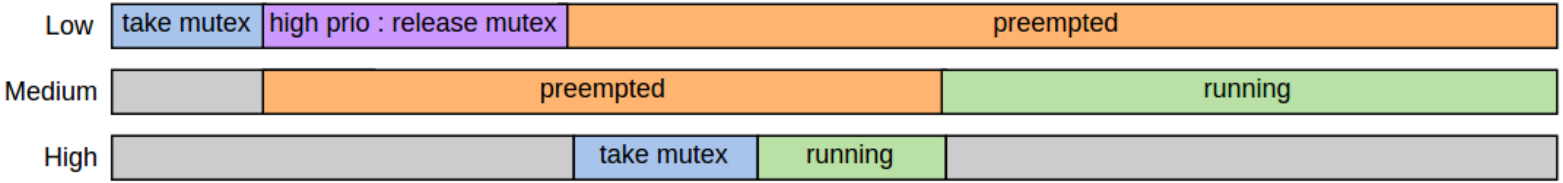
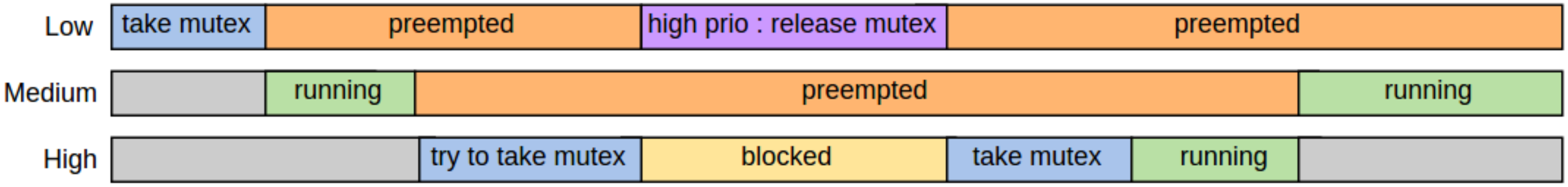


Figure 5 : Priority ceiling protocol

Priority inversion

Priority inheritance

- Better if the low priority task accesses the resource more often



Low: take mutex | preempted | high prio : release mutex | preempted

Medium: running | preempted | running

High: try to take mutex | blocked | take mutex | running

Figure 6 : Priority inheritance

Literature review

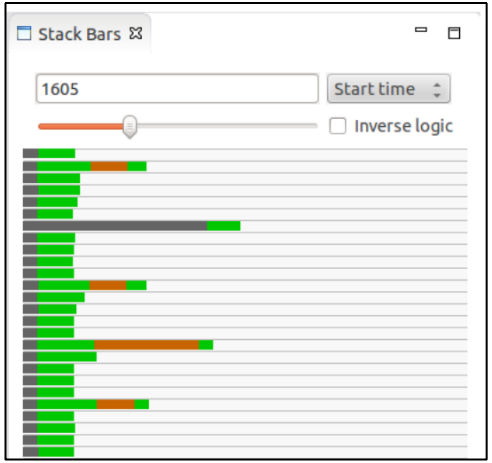
Linux low-latency tracing for multicore hard real-time systems (Beamonte, 2013)

- LTTng-UST modification to **reduce** the added **latency**
- Demonstrated **low latency** tracing with LTTng

Literature review

Real-time Linux analysis using low-impact tracer (Rajotte, 2014)

- Recreate the task states using kernel events
- Compare executions of a task
- Sort the executions by running time
- Limitations
    - Threads need to have different priorities
    - Model is fixed
    - Not working with SCHED_DEADLINE
    - Manual analysis to find problems
    - Problems when more than one processor



1605 Start time

☐ Inverse logic

Figure 7 : Original stackbars view

Modeling

Advantage of using only kernel events

- No need to modify the application source code to add tracepoints manually

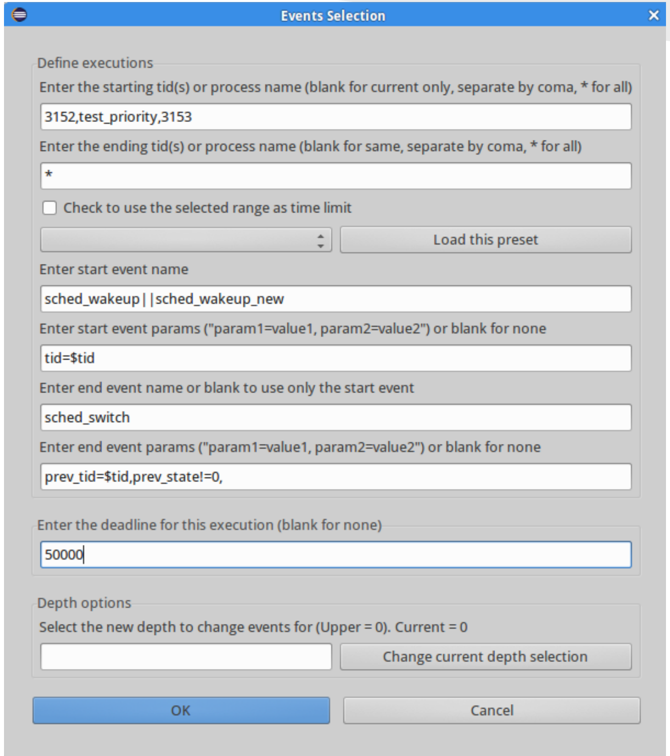
Modeling

- Identify executions automatically and then let the users choose between some valid models
  - Define a support ratio
  - Find all event types that are more frequent than the ratio
  - Increase the episode sizes using the fact that the sub-episodes must also be supported
  - Difficulties :
    - Using only event types
    - Execution time and memory usage
    - Many possible resulting models

Modeling : method

State machine

- User identifies :
  - an execution or
  - events that define the start and the end (name, parameters with operations, etc.)
  - TIDs for start and end
  - Presets for common cases



Define executions

Enter the starting tid(s) or process name (blank for current only, separate by coma, * for all)

3152,test_priority,3153

Enter the ending tid(s) or process name (blank for same, separate by coma, * for all)

*

□ Check to use the selected range as time limit

Enter start event name

sched_wakeup||sched_wakeup_new

Enter start event params ("param1=value1, param2=value2") or blank for none

tid=$tid

Enter end event name or blank to use only the start event

sched_switch

Enter end event params ("param1=value1, param2=value2") or blank for none

prev_tid=$tid,prev_state!=0,

Enter the deadline for this execution (blank for none)

50000

Depth options

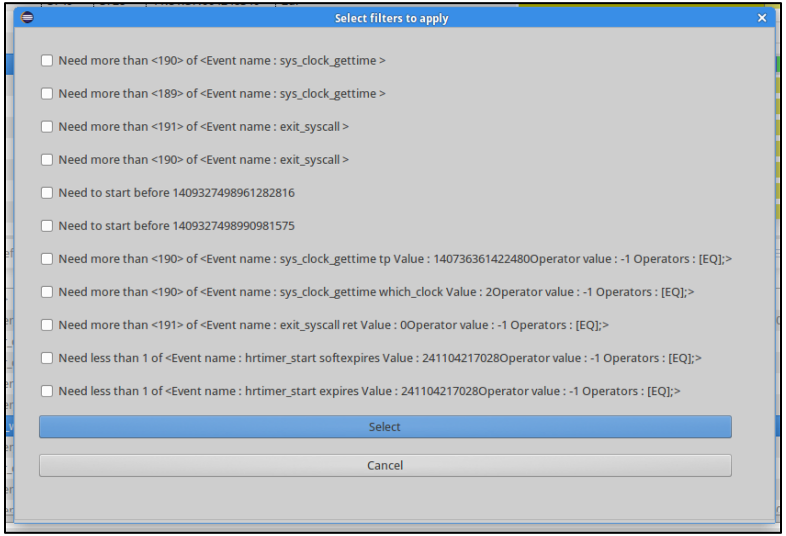
Select the new depth to change events for (Upper = 0). Current = 0

Figure 8 : Dialog to define model

Modeling : method

State machine

- Remove execution
- Add execution
- Define an execution as invalid and recalculate
  - Will suggest some modifications to the model based on differences between valid and invalid executions
  - The user can select the ones he wants to apply



□ Need more than <190> of <Event name : sys_clock_gettime >

□ Need more than <189> of <Event name : sys_clock_gettime >

□ Need more than <191> of <Event name : exit_syscall >

□ Need more than <190> of <Event name : exit_syscall >

□ Need to start before 1409327498961282816

□ Need to start before 1409327498990981575

□ Need more than <190> of <Event name : sys_clock_gettime tp Value : 140736361422480Operator value : -1 Operators : [EQ];>

□ Need more than <190> of <Event name : sys_clock_gettime which_clock Value : 2Operator value : -1 Operators : [EQ];>

□ Need more than <191> of <Event name : exit_syscall ret Value : 0Operator value : -1 Operators : [EQ];>

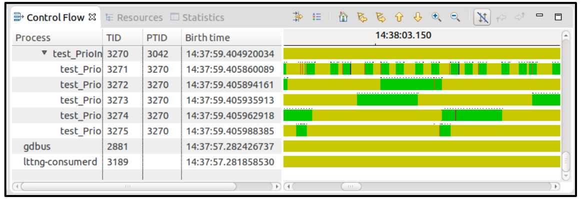
□ Need less than 1 of <Event name : hrtimer_start softexpires Value : 241104217028Operator value : -1 Operators : [EQ];>

□ Need less than 1 of <Event name : hrtimer_start expires Value : 241104217028Operator value : -1 Operators : [EQ];>

Figure 9 : Dialog to select modifications to apply

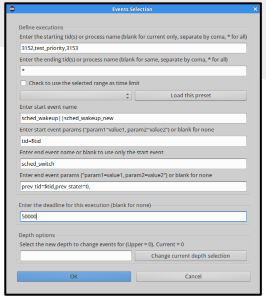
Overview

1) Control Flow View



Process	TID	PTID	Birth time
▼ test_PrioIn	3270	3042	14:37:59.404920034
test_Prio	3271	3270	14:37:59.405860089
test_Prio	3272	3270	14:37:59.405894161
test_Prio	3273	3270	14:37:59.405935913
test_Prio	3274	3270	14:37:59.405962918
test_Prio	3275	3270	14:37:59.405988385
gdbus	2881		14:37:57.282426737
lttng-consumerd	3189		14:37:57.281858530

2) Define executions



Define executions

Enter the starting tid(s) or process name (blank for current only, separate by coma, * for all)

3152,test_priority,3153

Enter the ending tid(s) or process name (blank for same, separate by coma, * for all)

*

□ Check to use the selected range as time limit

Enter start event name

sched_wakeup||sched_wakeup_new

Enter start event params ("param1=value1, param2=value2") or blank for none

tid=$tid

Enter end event name or blank to use only the start event

sched_switch

Enter end event params ("param1=value1, param2=value2") or blank for none

prev_tid=$tid,prev_state!=0,

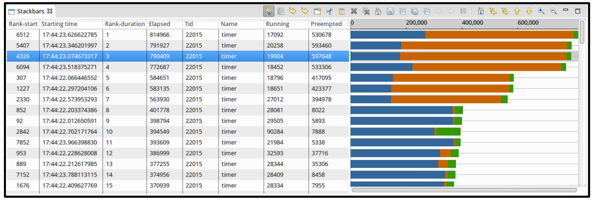
Enter the deadline for this execution (blank for none)

50000

Depth options

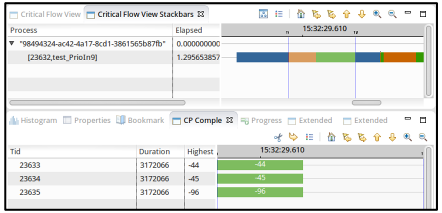
Select the new depth to change events for (Upper = 0). Current = 0

3) Stackbars View



Rank-start	Starting time	Rank-duration	Elapsed	Tid	Name	Running	Preempted
6512	17:44:23.626622785	1	814966	22015	timer	17092	530678
5407	17:44:23.346201997	2	791927	22015	timer	20258	593460
4326	17:44:23.074673317	3	790409	22015	timer	19904	597648
6094	17:44:23.518375271	4	772687	22015	timer	18452	533306
307	17:44:22.066446552	5	584651	22015	timer	18796	417095
1227	17:44:22.297204106	6	583135	22015	timer	18651	423377
2330	17:44:22.573953293	7	563930	22015	timer	27012	394978
852	17:44:22.203374386	8	401778	22015	timer	28081	8022
92	17:44:22.012650591	9	398794	22015	timer	29505	5893
2842	17:44:22.702171764	10	394549	22015	timer	90284	7888
7852	17:44:23.966398830	11	393609	22015	timer	21984	5338
953	17:44:22.228628008	12	386999	22015	timer	32593	37716
889	17:44:22.212617985	13	377255	22015	timer	28344	35306
7152	17:44:23.788113115	14	374956	22015	timer	28409	8458
1676	17:44:22.409627769	15	370939	22015	timer	28334	7955

4) Critical Flow View with CP Complement view



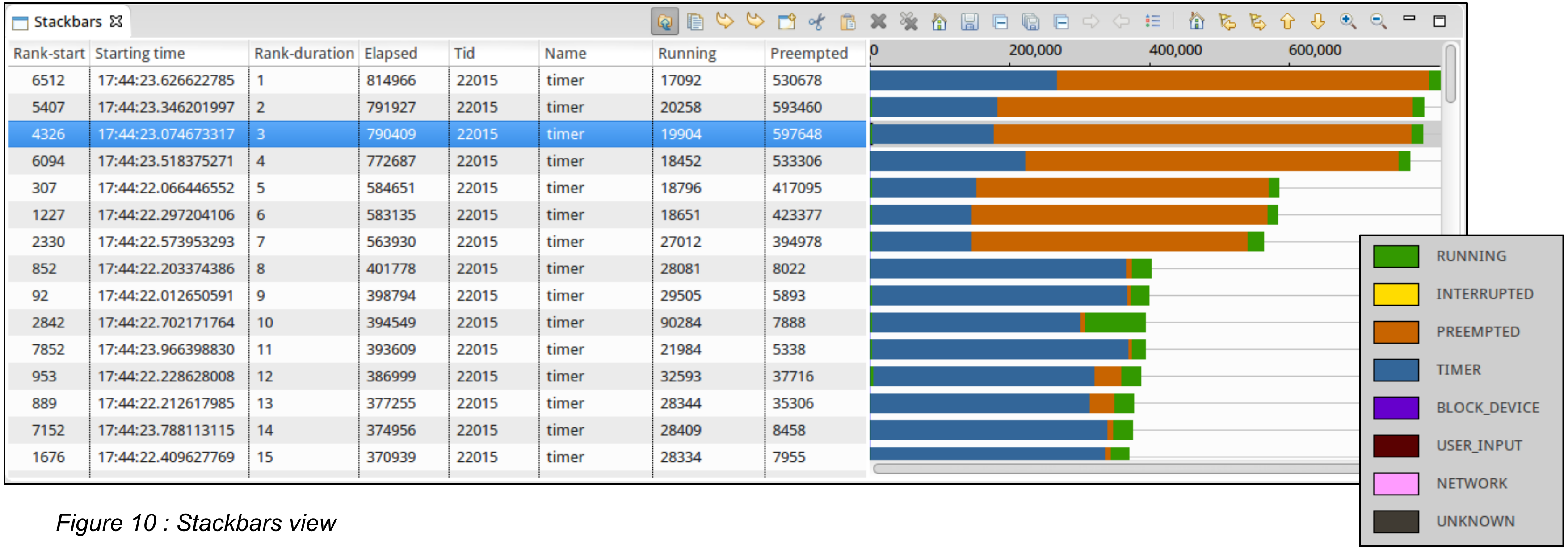
Process	Elapsed
▼ "98494324-ac42-4a17-8cd1-3861565b87fb"	0.000000000
[23632,test_PrioIn9]	1.295653857

Tid	Duration	Highest
23633	3172066	-44
23634	3172066	-45
23635	3172066	-96

5) Other views

Views

Stackbars view

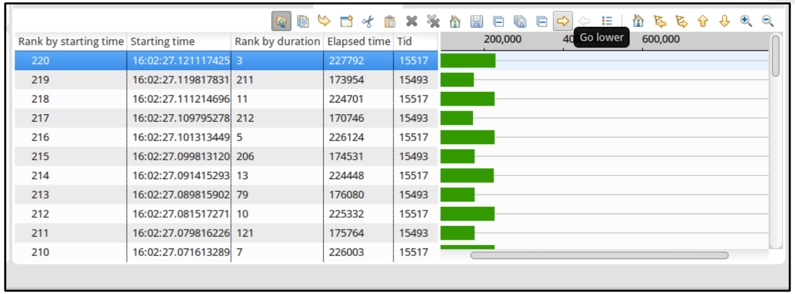


Rank-start	Starting time	Rank-duration	Elapsed	Tid	Name	Running	Preempted
6512	17:44:23.626622785	1	814966	22015	timer	17092	530678
5407	17:44:23.346201997	2	791927	22015	timer	20258	593460
4326	17:44:23.074673317	3	790409	22015	timer	19904	597648
6094	17:44:23.518375271	4	772687	22015	timer	18452	533306
307	17:44:22.066446552	5	584651	22015	timer	18796	417095
1227	17:44:22.297204106	6	583135	22015	timer	18651	423377
2330	17:44:22.573953293	7	563930	22015	timer	27012	394978
852	17:44:22.203374386	8	401778	22015	timer	28081	8022
92	17:44:22.012650591	9	398794	22015	timer	29505	5893
2842	17:44:22.702171764	10	394549	22015	timer	90284	7888
7852	17:44:23.966398830	11	393609	22015	timer	21984	5338
953	17:44:22.228628008	12	386999	22015	timer	32593	37716
889	17:44:22.212617985	13	377255	22015	timer	28344	35306
7152	17:44:23.788113115	14	374956	22015	timer	28409	8458
1676	17:44:22.409627769	15	370939	22015	timer	28334	7955

Figure 10 : Stackbars view

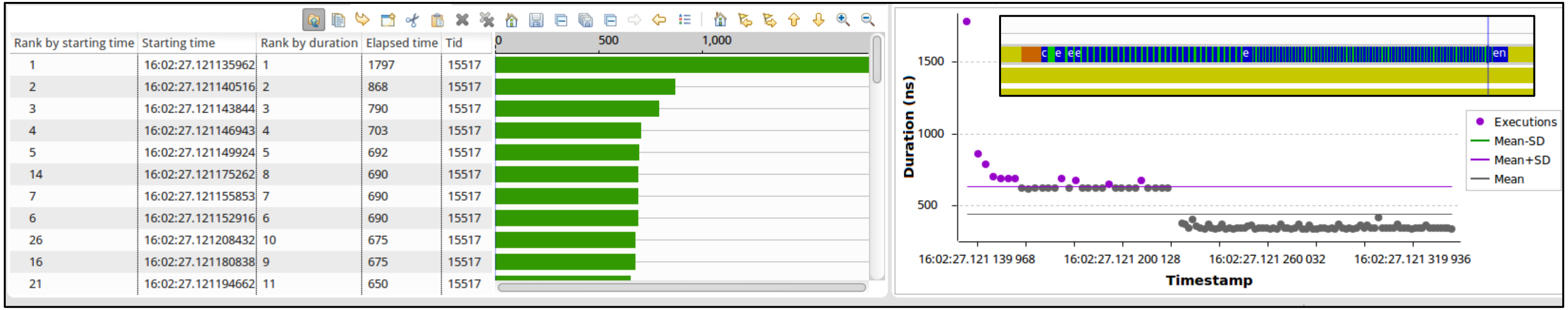
Views

- Supports
    - Thread pool
    - Nested executions



Rank by starting time	Starting time	Rank by duration	Elapsed time	Tid
220	16:02:27.121117425	3	227792	15517
219	16:02:27.119817831	211	173954	15493
218	16:02:27.111214696	11	224701	15517
217	16:02:27.109795278	212	170746	15493
216	16:02:27.101313449	5	226124	15517
215	16:02:27.099813120	206	174531	15493
214	16:02:27.091415293	13	224448	15517
213	16:02:27.089815902	79	176080	15493
212	16:02:27.081517271	10	225332	15517
211	16:02:27.079816226	121	175764	15493
210	16:02:27.071613289	7	226003	15517

Figure 11 : Task on multiple threads



Rank by starting time	Starting time	Rank by duration	Elapsed time	Tid
1	16:02:27.121135962	1	1797	15517
2	16:02:27.121140516	2	868	15517
3	16:02:27.121143844	3	790	15517
4	16:02:27.121146943	4	703	15517
5	16:02:27.121149924	5	692	15517
14	16:02:27.121175262	8	690	15517
7	16:02:27.121155853	7	690	15517
6	16:02:27.121152916	6	690	15517
26	16:02:27.121208432	10	675	15517
16	16:02:27.121180838	9	675	15517
21	16:02:27.121194662	11	650	15517

Figure 12 : Nested executions

Views

Time View

- View of duration by starting timestamp
- Synced with other views

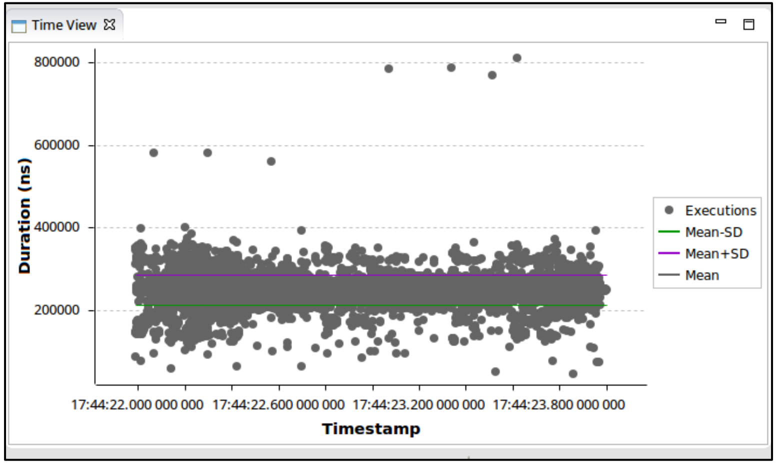
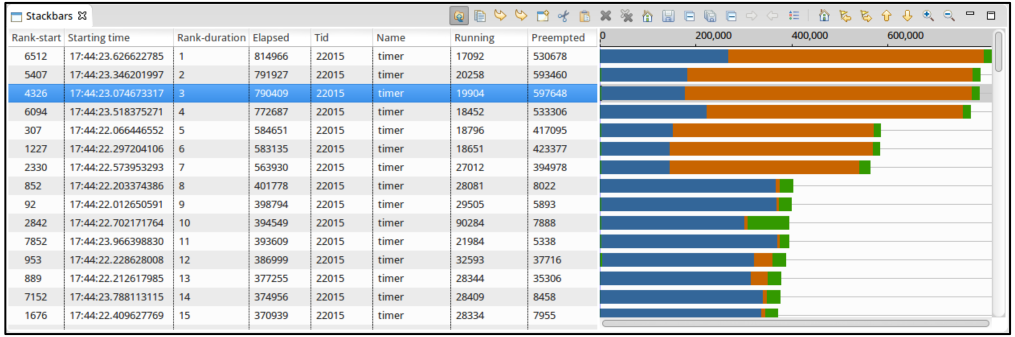


Figure 13 : Time view



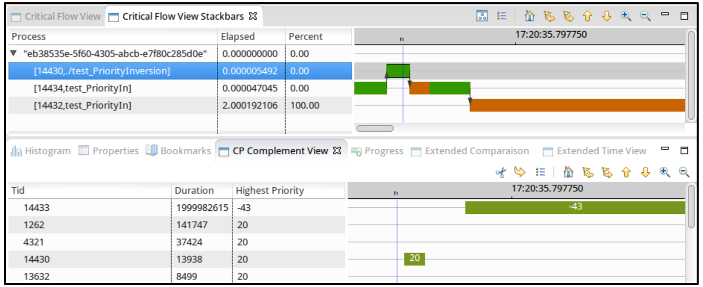
Rank-start	Starting time	Rank-duration	Elapsed	Tid	Name	Running	Preempted
6512	17:44:23.626622785	1	814966	22015	timer	17092	530678
5407	17:44:23.346201997	2	791927	22015	timer	20258	593460
4326	17:44:23.074673317	3	790409	22015	timer	19904	597648
6094	17:44:23.518375271	4	772687	22015	timer	18452	533306
307	17:44:22.066446552	5	584651	22015	timer	18796	417095
1227	17:44:22.297204106	6	583135	22015	timer	18651	423377
2330	17:44:22.573953293	7	563930	22015	timer	27012	394978
852	17:44:22.203374386	8	401778	22015	timer	28081	8022
92	17:44:22.012650591	9	398794	22015	timer	29505	5893
2842	17:44:22.702171764	10	394549	22015	timer	90284	7888
7852	17:44:23.966398830	11	393609	22015	timer	21984	5338
953	17:44:22.228628008	12	386999	22015	timer	32593	37716
889	17:44:22.212617985	13	377255	22015	timer	28344	35306
7152	17:44:23.788113115	14	374956	22015	timer	28409	8458
1676	17:44:22.409627769	15	370939	22015	timer	28334	7955

Figure 14 : Stackbars view

Views

CP Complement View

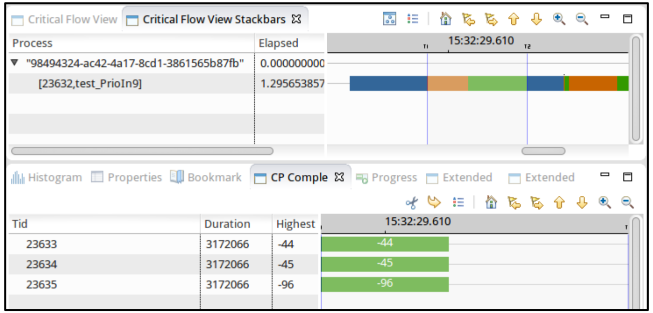
- Show the priority of all running threads during preemption period of any thread in the critical path



Process	Elapsed	Percent
▼ "eb38535e-5f60-4305-abcb-e7f80c285d0e"	0.000000000	0.00
[14430,./test_PriorityInversion]	0.000005492	0.00
[14434,test_PriorityIn]	0.000047045	0.00
[14432,test_PriorityIn]	2.000192106	100.00

Tid	Duration	Highest Priority
14433	1999982615	-43
1262	141747	20
4321	37424	20
14430	13938	20
13632	8499	20

Figure 15 : CP Complement view



Process	Elapsed
▼ "98494324-ac42-4a17-8cd1-3861565b87fb"	0.000000000
[23632,test_PrioIn9]	1.295653857

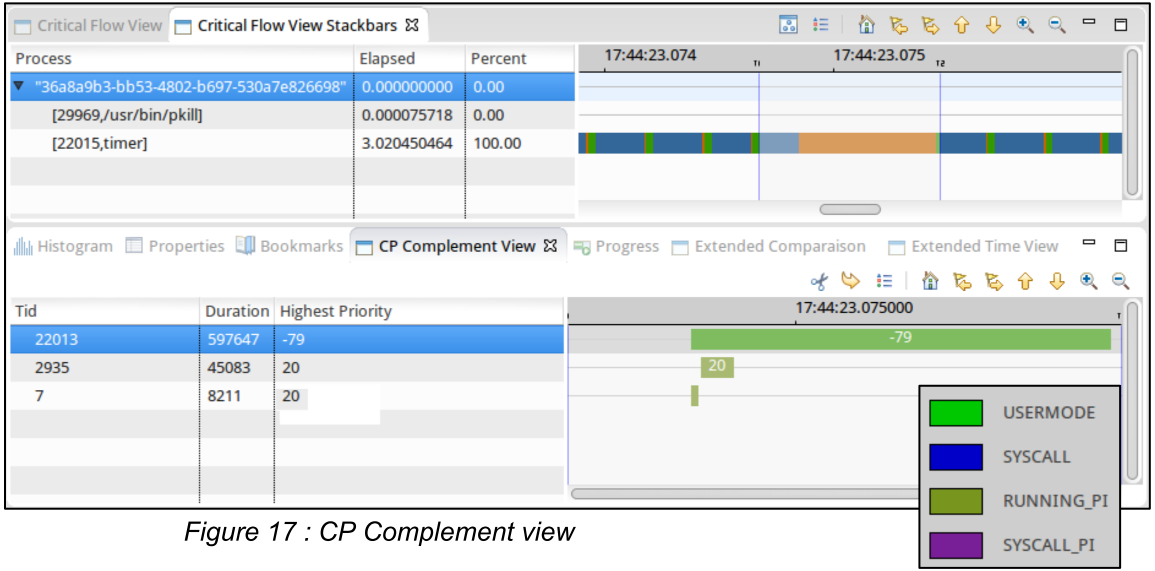
Tid	Duration	Highest
23633	3172066	-44
23634	3172066	-45
23635	3172066	-96

Figure 16 : CP Complement view

Views

CP Complement View

- Detect priority inversion



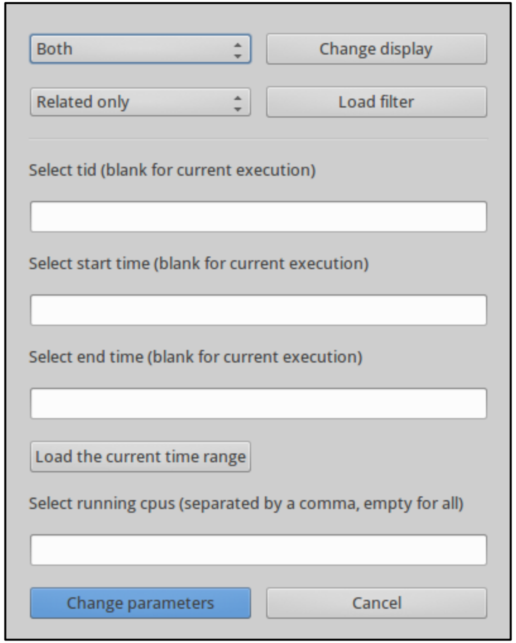
Critical Flow View | Critical Flow View Stackbars

Process	Elapsed	Percent
▼ "36a8a9b3-bb53-4802-b697-530a7e826698"	0.000000000	0.00
[29969,/usr/bin/pkill]	0.000075718	0.00
[22015,timer]	3.020450464	100.00

Histogram | Properties | Bookmarks | CP Complement View | Progress | Extended Comparaison | Extended Time View

Tid	Duration	Highest Priority
22013	597647	-79
2935	45083	20
7	8211	20

Figure 17 : CP Complement view



Both | Change display

Related only | Load filter

Select tid (blank for current execution)

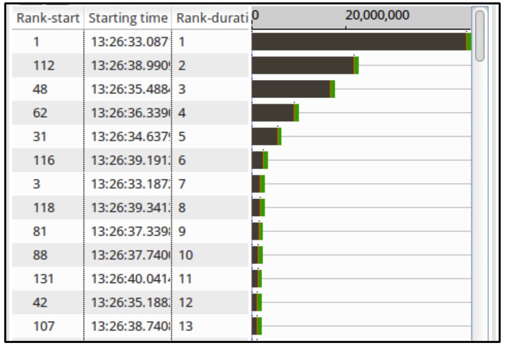
Select start time (blank for current execution)

Select end time (blank for current execution)

Select running cpus (separated by a comma, empty for all)

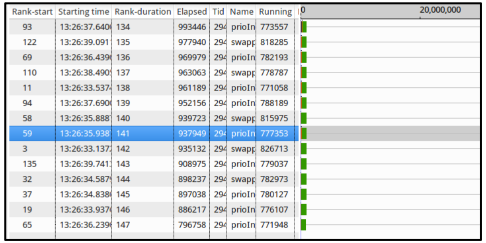
Example

Find out why some executions take more time



Rank-start	Starting time	Rank-durati
1	13:26:33.087	1
112	13:26:38.990	2
48	13:26:35.488	3
62	13:26:36.339	4
31	13:26:34.637	5
116	13:26:39.191	6
3	13:26:33.187	7
118	13:26:39.341	8
81	13:26:37.339	9
88	13:26:37.740	10
131	13:26:40.041	11
42	13:26:35.188	12
107	13:26:38.740	13

Figure 18 : Problematic executions



Rank-start	Starting time	Rank-duration	Elapsed	Tid	Name	Running
93	13:26:37.640	134	993446	294	prioIn	773557
122	13:26:39.091	135	977940	294	swapp	818285
69	13:26:36.439	136	969979	294	prioIn	782193
110	13:26:38.490	137	963063	294	swapp	778787
11	13:26:33.537	138	961189	294	prioIn	771058
94	13:26:37.690	139	952156	294	prioIn	788189
58	13:26:35.888	140	939723	294	swapp	815975
59	13:26:35.938	141	937949	294	prioIn	777353
3	13:26:33.137	142	935132	294	swapp	826713
135	13:26:39.741	143	908975	294	prioIn	779037
32	13:26:34.587	144	898237	294	swapp	782973
37	13:26:34.838	145	897038	294	prioIn	780127
19	13:26:33.937	146	886217	294	prioIn	776107
65	13:26:36.239	147	796758	294	prioIn	771948

Figure 19 : Normal executions

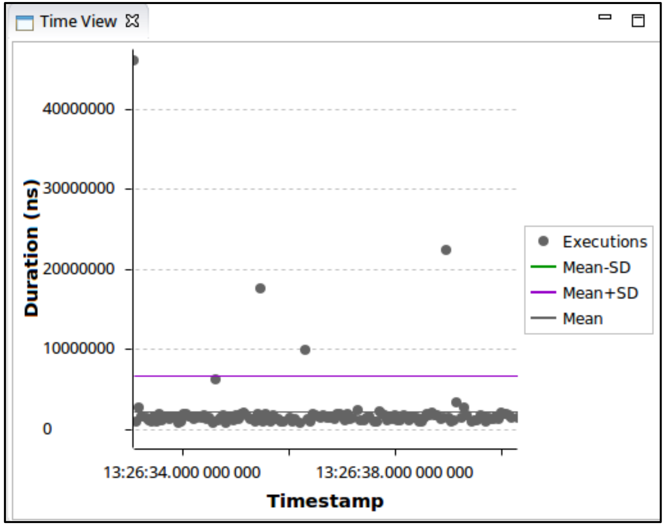
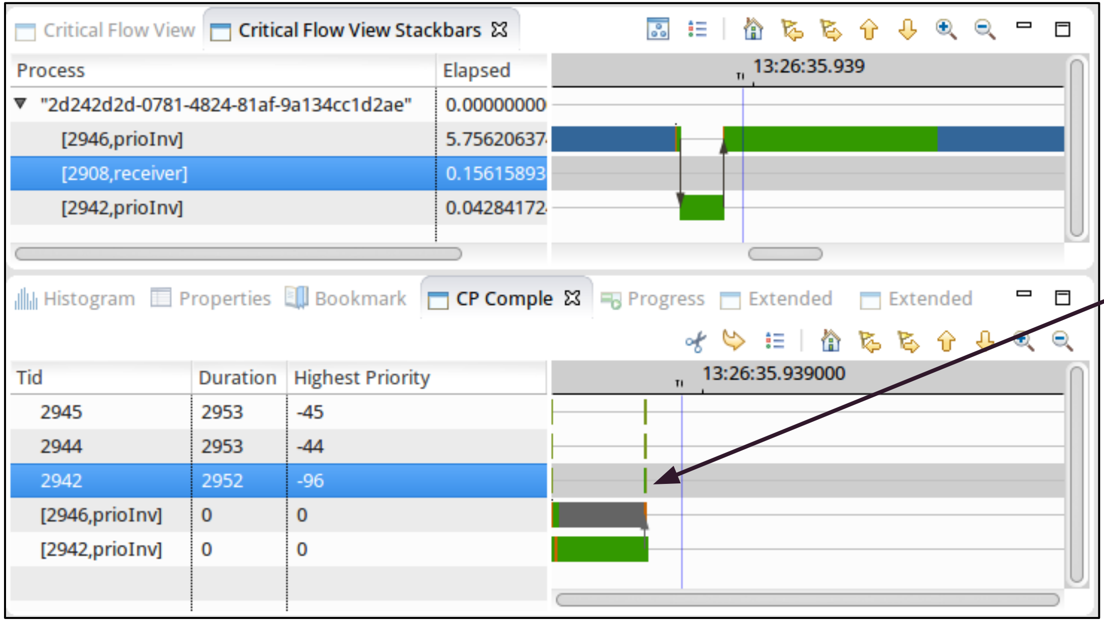


Figure 20 : Time View

Example

Introduction
Literature
Modeling
Views
Results
Conclusion

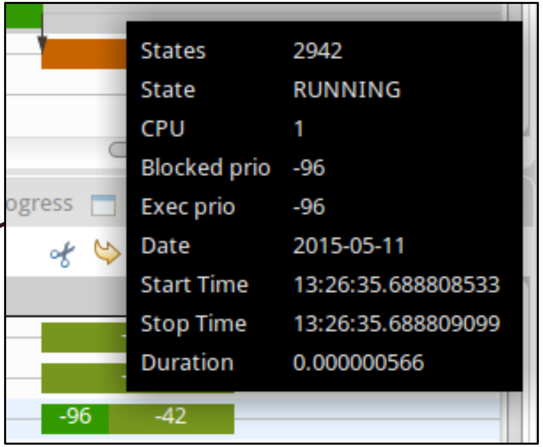
Normal execution



Process	Elapsed
"2d242d2d-0781-4824-81af-9a134cc1d2ae"	0.00000000
[2946,prioInv]	5.75620637
[2908,receiver]	0.15615893
[2942,prioInv]	0.04284172

Tid	Duration	Highest Priority
2945	2953	-45
2944	2953	-44
2942	2952	-96
[2946,prioInv]	0	0
[2942,prioInv]	0	0

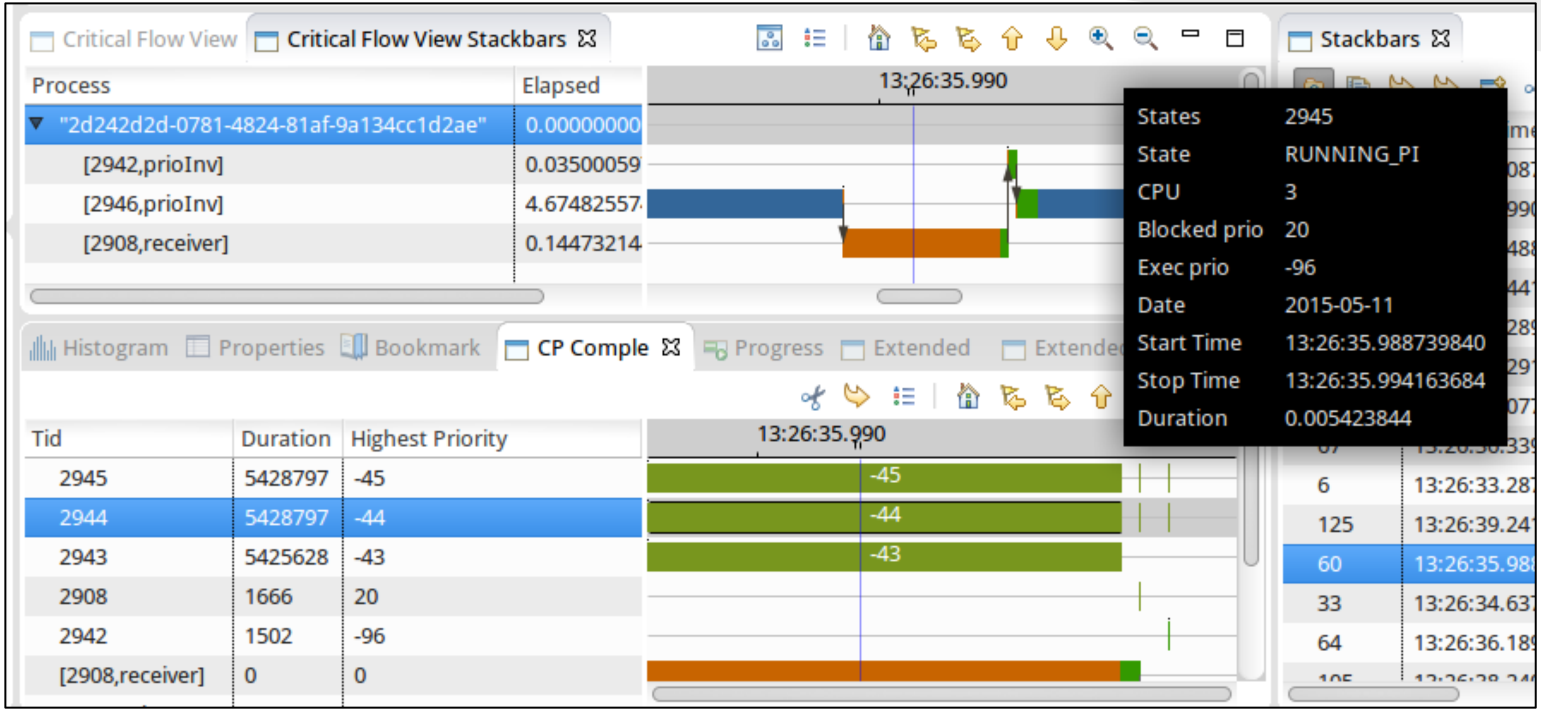
Figure 21 : CP Complement of a normal execution



States	2942
State	RUNNING
CPU	1
Blocked prio	-96
Exec prio	-96
Date	2015-05-11
Start Time	13:26:35.688808533
Stop Time	13:26:35.688809099
Duration	0.000000566

Figure 22 : There was priority inheritance

Example



Process	Elapsed
▼ "2d242d2d-0781-4824-81af-9a134cc1d2ae"	0.00000000
[2942,prioInv]	0.03500059
[2946,prioInv]	4.67482557
[2908,receiver]	0.14473214

States	2945
State	RUNNING_PI
CPU	3
Blocked prio	20
Exec prio	-96
Date	2015-05-11
Start Time	13:26:35.988739840
Stop Time	13:26:35.994163684
Duration	0.005423844

Tid	Duration	Highest Priority
2945	5428797	-45
2944	5428797	-44
2943	5425628	-43
2908	1666	20
2942	1502	-96
[2908,receiver]	0	0

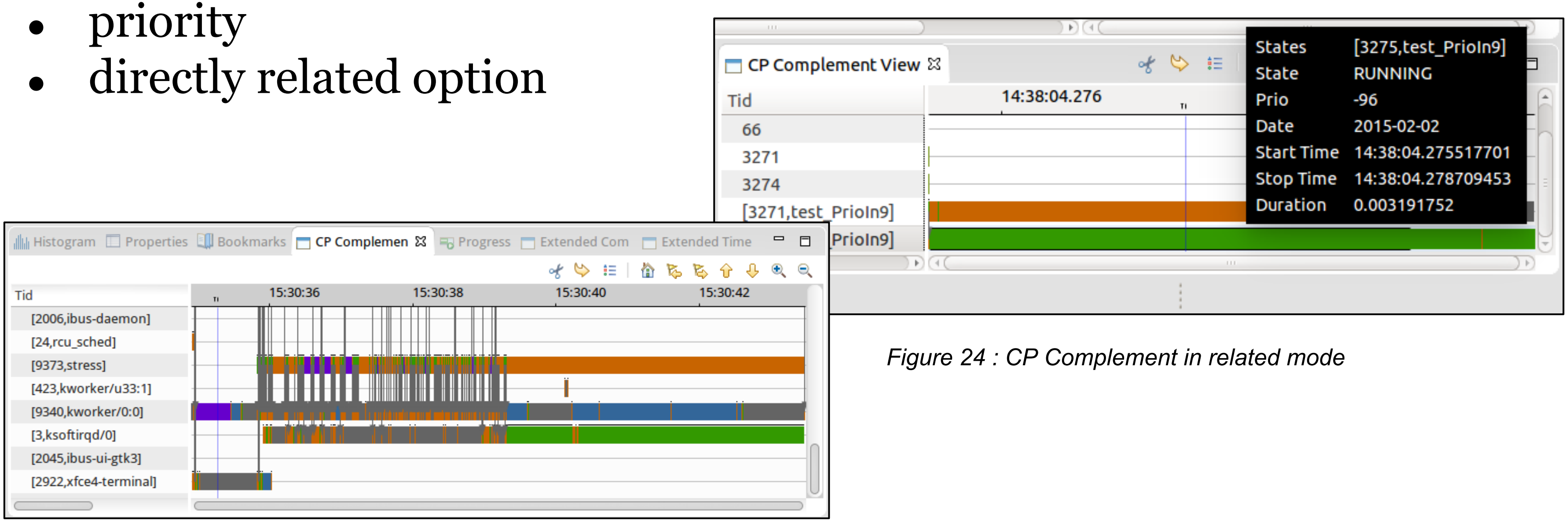
6	13:26:33.28
125	13:26:39.24
60	13:26:35.98
33	13:26:34.63
64	13:26:36.18

Figure 23 : CP Complement of a problematic execution

Other results

Output the dependencies during an execution

- priority
- directly related option



Tid
66
3271
3274
[3271,test_PrioIn9]
PrioIn9]

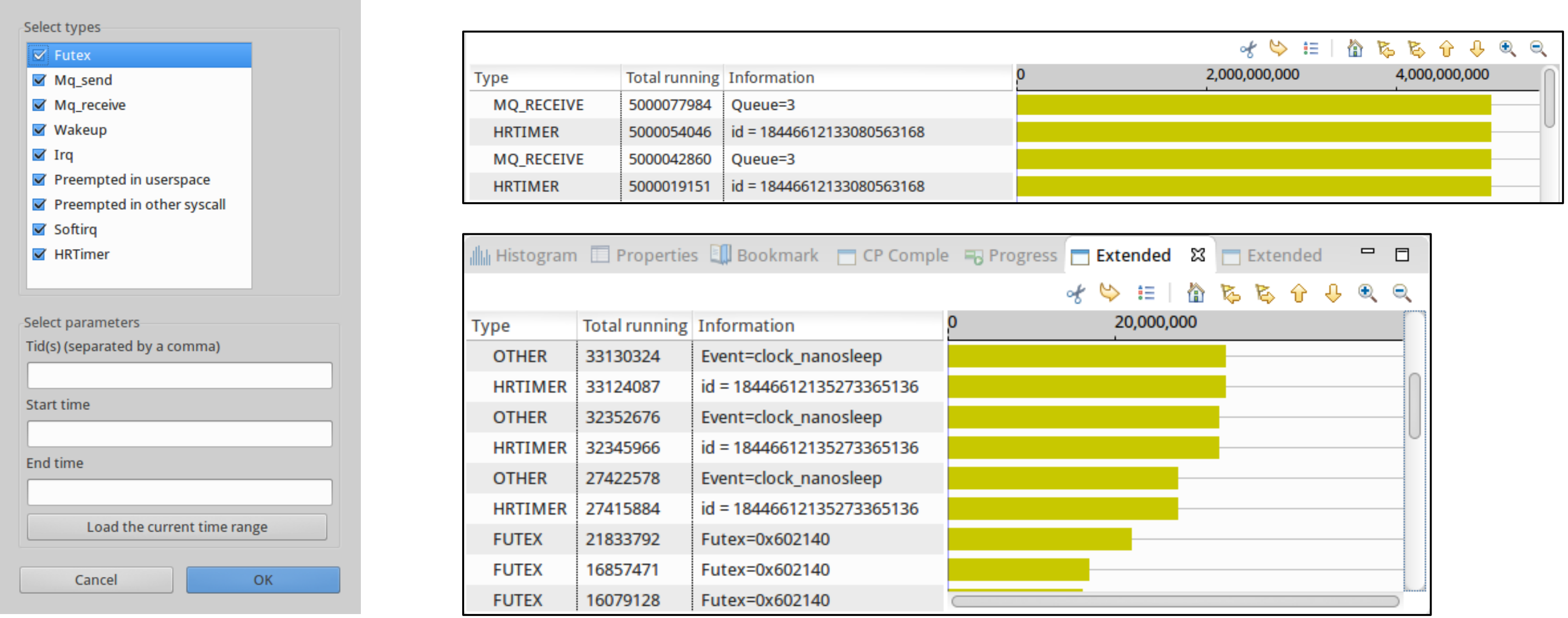
States	[3275,test_PrioIn9]
State	RUNNING
Prio	-96
Date	2015-02-02
Start Time	14:38:04.275517701
Stop Time	14:38:04.278709453
Duration	0.003191752

Tid
[2006,ibus-daemon]
[24,rcu_sched]
[9373,stress]
[423,kworker/u33:1]
[9340,kworker/0:0]
[3,ksoftirqd/0]
[2045,ibus-ui-gtk3]
[2922,xfce4-terminal]

Figure 24 : CP Complement in related mode

Other results

Extended comparison view



Select types

- ☑ Futex
- ☑ Mq_send
- ☑ Mq_receive
- ☑ Wakeup
- ☑ Irq
- ☑ Preempted in userspace
- ☑ Preempted in other syscall
- ☑ Softirq
- ☑ HRTimer

Select parameters

Tid(s) (separated by a comma)

Start time

End time

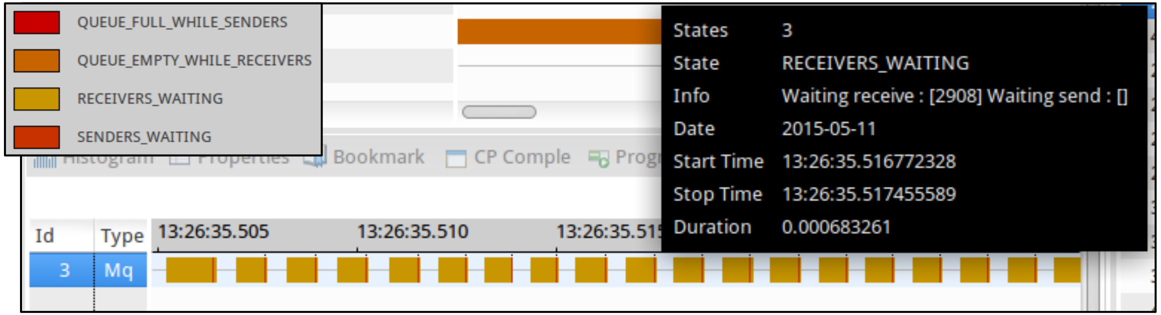
Type	Total running	Information
MQ_RECEIVE	5000077984	Queue=3
HRTIMER	5000054046	id = 18446612133080563168
MQ_RECEIVE	5000042860	Queue=3
HRTIMER	5000019151	id = 18446612133080563168

Type	Total running	Information
OTHER	33130324	Event=clock_nanosleep
HRTIMER	33124087	id = 18446612135273365136
OTHER	32352676	Event=clock_nanosleep
HRTIMER	32345966	id = 18446612135273365136
OTHER	27422578	Event=clock_nanosleep
HRTIMER	27415884	id = 18446612135273365136
FUTEX	21833792	Futex=0x602140
FUTEX	16857471	Futex=0x602140
FUTEX	16079128	Futex=0x602140

Figure 25 : Extended comparison view

Other results

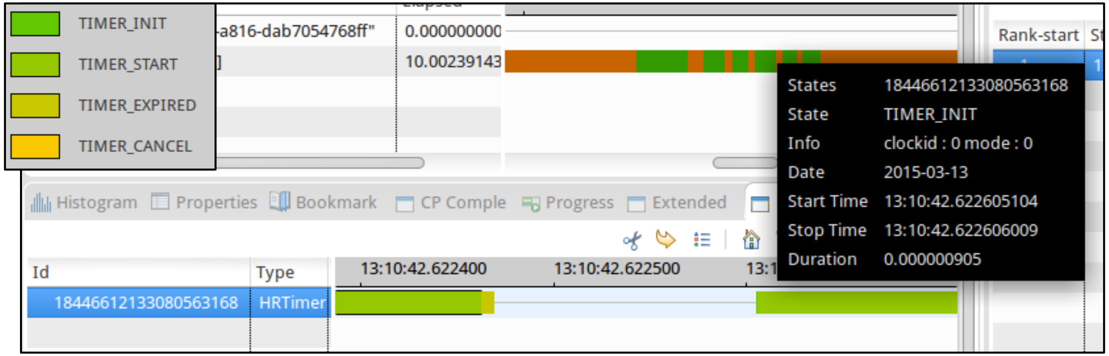
Extended time view : queue



🟥 QUEUE_FULL_WHILE_SENDERS
🟧 QUEUE_EMPTY_WHILE_RECEIVERS
🟨 RECEIVERS_WAITING
🟥 SENDERS_WAITING

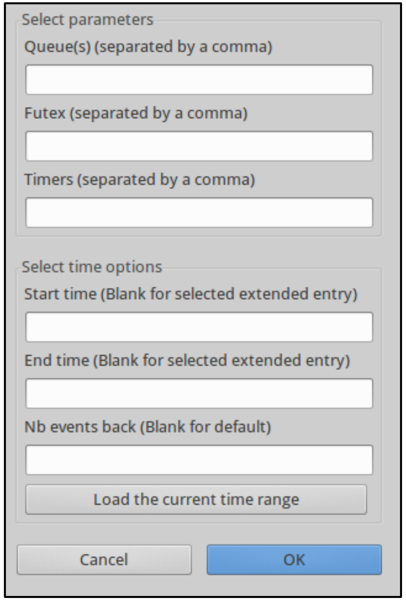
States	3
State	RECEIVERS_WAITING
Info	Waiting receive : [2908] Waiting send : []
Date	2015-05-11
Start Time	13:26:35.516772328
Stop Time	13:26:35.517455589
Duration	0.000683261

HRTimer



🟩 TIMER_INIT
🟩 TIMER_START
🟨 TIMER_EXPIRED
🟨 TIMER_CANCEL

States	18446612133080563168
State	TIMER_INIT
Info	clockid : 0 mode : 0
Date	2015-03-13
Start Time	13:10:42.622605104
Stop Time	13:10:42.622606009
Duration	0.000000905



Select parameters

Queue(s) (separated by a comma)

Futex (separated by a comma)

Timers (separated by a comma)

Select time options

Start time (Blank for selected extended entry)

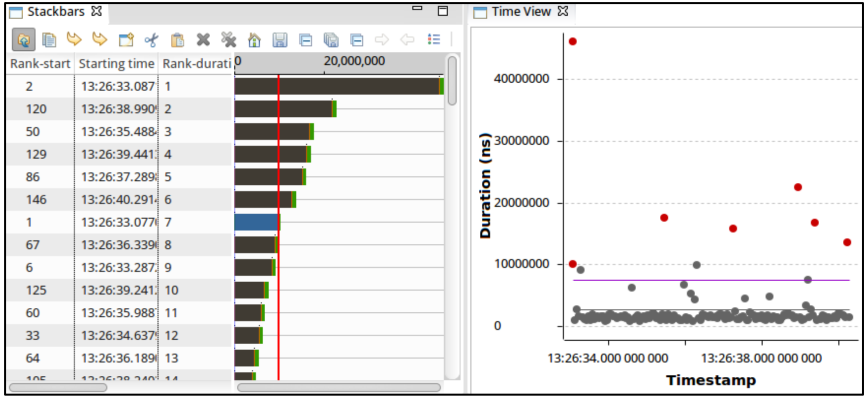
End time (Blank for selected extended entry)

Nb events back (Blank for default)

Figure 26 : Extended time view

Other results

- Deadline analysis
    - Tell which executions missed their deadlines
    - User input



Rank-start	Starting time	Rank-durati
2	13:26:33.087	1
120	13:26:38.990	2
50	13:26:35.488	3
129	13:26:39.441	4
86	13:26:37.289	5
146	13:26:40.291	6
1	13:26:33.077	7
67	13:26:36.339	8
6	13:26:33.287	9
125	13:26:39.241	10
60	13:26:35.988	11
33	13:26:34.637	12
64	13:26:36.189	13
105	13:26:38.240	14

Figure 27 : Deadline

Conclusion

- Future work
  - Modeling
    - Instrument complex real-time application in user-space and for each task, validate if it is possible to model only with kernel events
  - Analysis
    - Validate with real bugs
    - Add new analysis
- Questions?