ERICSSON

The great troubleshooting encounter:

CDT meets Trace Compass

EclipseCon, March 2015

Marc Khouzam

Marc-André Laperle

ABOUT US

- Marc Khouzam
  - Software Developer at Ericsson since 1998
  - CDT project co-lead, focusing on Debugging
  - Working with CDT since 2009

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

AGENDA

- A bit of background: Debug and Tracing
- CDT Debug and Trace Compass integration
  - An integration in 4 parts
- Conclusion

ERICSSON

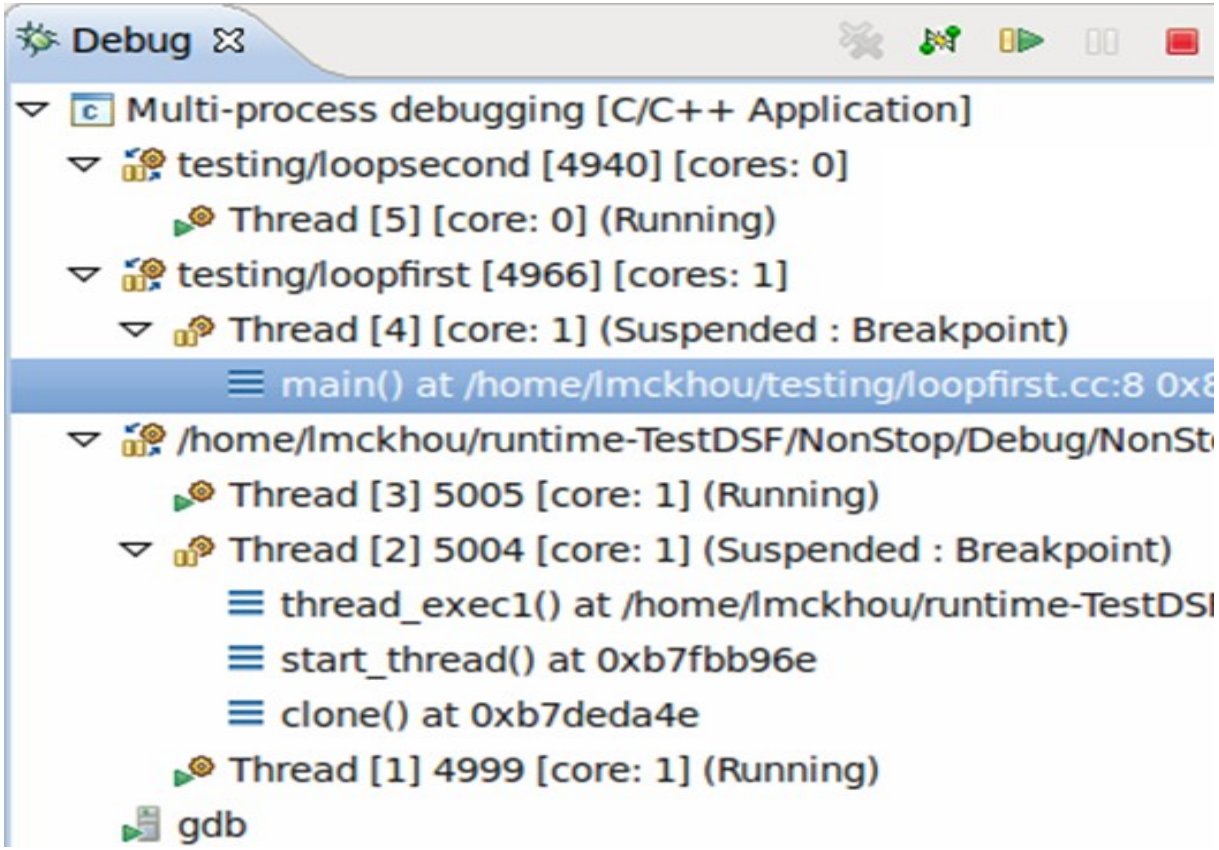
A LITTLE BACKGROUND:

ADVANCED DEBUGGING

NON-STOP DEBUGGING

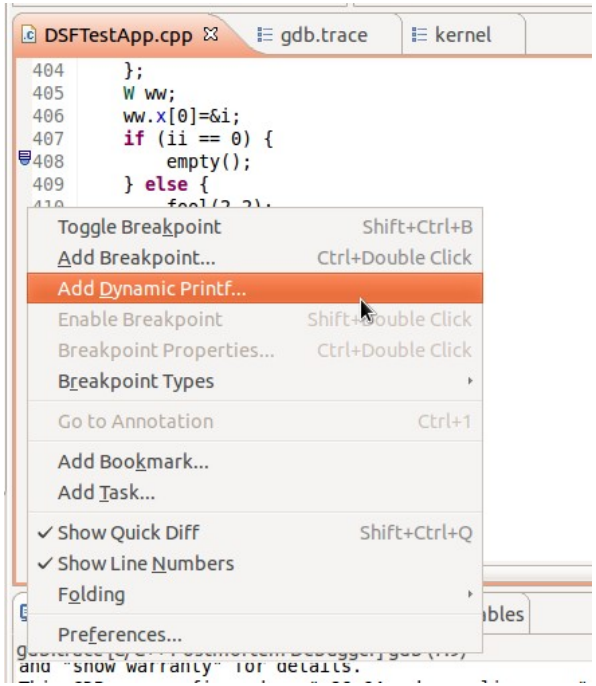
› Program continues execution while suspending some threads

› Reduced intrusiveness



DYNAMIC-PRINTF

- Sometimes traces are necessary
- Printf without recompiling or redeploying!



OS AWARENESS

› Access to system information while debugging

- Message Queues
- Loaded Kernel Modules
- Sockets
- Process Groups
- File Descriptors
- All Processes
- Semaphores
- All Threads
- Shared Memory Segments

OS resources | Kernel modules

Fetched at 11:37:46

Name	Size	Num uses	Dependencies	Status	Address
aes_generic	26863	1	aes_i586,	Live	f816f000
aes_i586	7268	2	-	Live	f80c8000
agpgart	31724	2	nvidia,intel_agp,	Live	f80cb000
arc4	1153	2	-	Live	f84b7000
binfmt_misc	6587	1	-	Live	f80bf000
bitblit	4707	1	fbcon,	Live	f81a9000
cfg80211	126528	3	iwlagn,iwlcore,mac80211,	Live	f830b000
cifs	248735	4	-	Live	f8393000
dm_crypt	11331	1	-	Live	f803b000
e1000e	119856	0	-	Live	f8127000
fbcon	35102	71	-	Live	f81f4000
font	7557	1	fbcon,	Live	f81d6000
hid	67032	1	usbhid,	Live	f80fa000

ERICSSON

MORE BACKGROUND:

TRACING WITH TRACE COMPASS

TRACE COMPASS

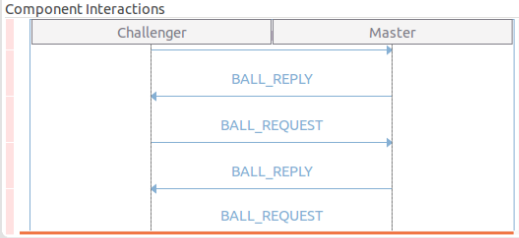
File Window Help

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

Statistics | **Sequence Diagram**

Component Interactions



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

Call Stack

Function	Depth	Entry time	Exit time	Duration
master_player-4715 (seq				

10:53:22.702 — 10:53:22.704 — 10:53:22.706

processBall() 4 4 proc 4 4 processBall() 4 4 processBall()

Ball Game View

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

10:53:22.702 — 10:53:22.704 — 10:53:22.706

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 | Properties | Bookmarks

Selection Start	10:53:22.703 391 532
Selection End	10:53:22.704 133 222
Window Span	000.006 063 508

6 / 0 — 10:53:22.701 281 155 — 10:53:22.707 344 663

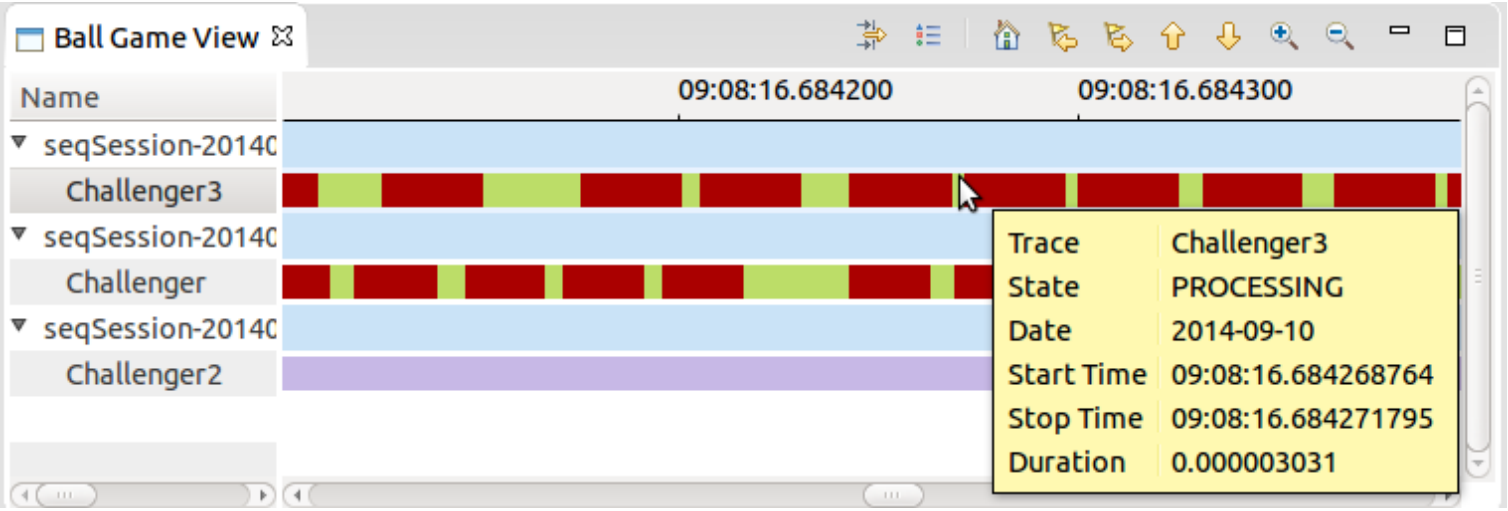
2435 / 0 — 10:53:22.597 002 449 — 10:53:27.538 216 624

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

COMMON FEATURES

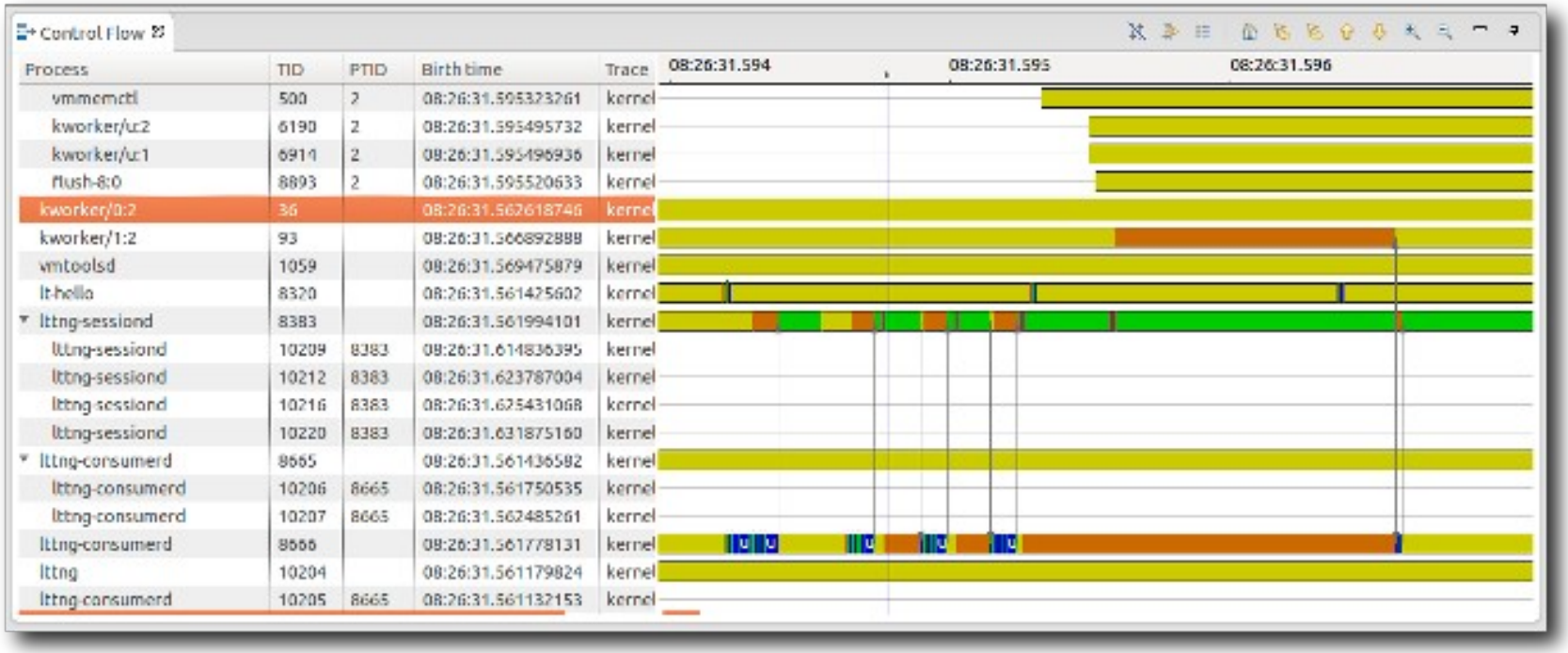
› Data-driven state system and views
- XML description of state changes to convert trace events to states
- XML description of view representation of the computed state system
- Can be created without changing source code or recompiling

› For example: 50 lines of XML created the view below



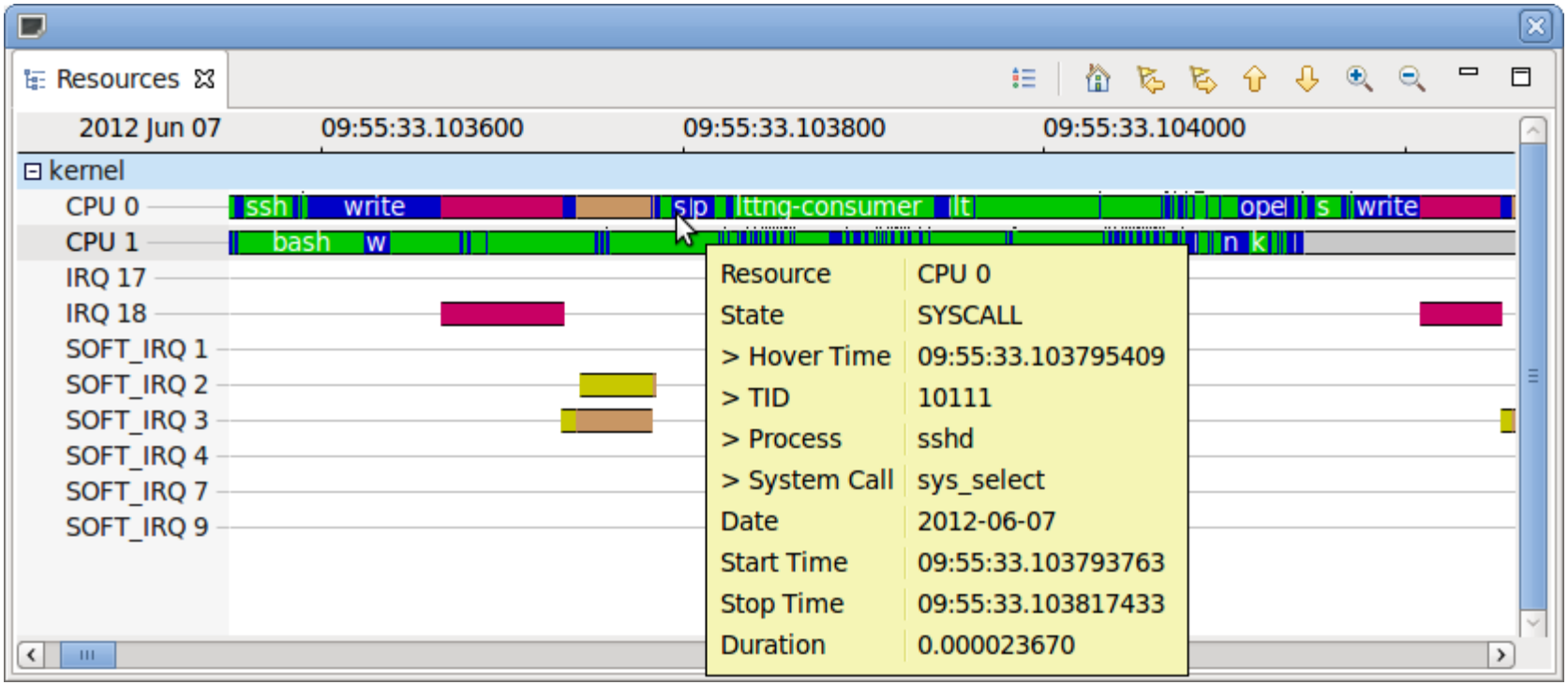
CONTROL FLOW VIEW

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



RESOURCES VIEW

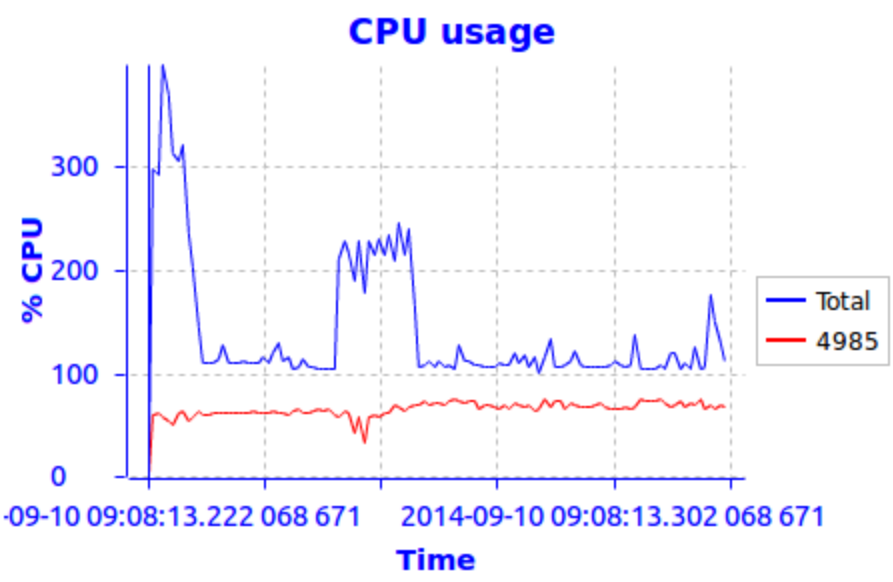
› Displays system resource states (color-coded) over time



CPU USAGE VIEW

CPU Usage

TID	Process	%	Time
4983	master_player	0.508 %	507549 ns
5079	java	0.172 %	172449 ns
4980	challenger	0.469 %	469351 ns
4986	master_player	0.508 %	507759 ns
4985	master_player	66.658 %	66657523 ns
5074	sh	0.041 %	41494 ns
5077	java	0.138 %	137739 ns
5076	java	0.013 %	13099 ns
4847	java	0.020 %	20163 ns
4988	kworker/3:1	0.595 %	595313 ns
3837	gtk-window-deco	0.074 %	73961 ns
5045	java	0.029 %	28705 ns



AGENDA

- A bit of background: Debug and Tracing

- CDT Debug and Trace Compass integration

  1. Enhanced Post-mortem troubleshooting
  2. Debugging with Trace snapshots
  3. Tracing with the (Multicore) Visualizer
  4. GDB Traces with Trace Compass

- Conclusion

ERICSSON

ENHANCED POST-MORTEM TROUBLESHOOTING

POST-MORTEM DEBUG

- Use GDB to examine core file
- Variables, Registers, Memory

Debug

- DSFTestApp Core [C/C++ Postmortem Debugger]
  - DSFTestApp
    - Thread #1 28327 (Suspended : Container)
      - dup() at DSFTestApp.cpp:203 0x4016fd
      - main() at DSFTestApp.cpp:244 0x4018b1
  - gdb.7.9 (7.9)

Variables | Breakpoints | Modules

Name	Type	Value
argc	int	1
argv	char **	0x7fffffffdf18
b	int [5]	0x7fffffffdcc0
um	wchar_t	32767 L'翻'
array_small	int [4]	0x7fffffffdcf0
de	int	0

DSFTestApp.cpp

```
        thatIsABigArray[i] = i;
    }

    float myFloat[4];
    myFloat[0] = 4.6;
    myFloat[1] = 3.14;
    myFloat[2] = 6;
    myFloat[3] = (float)1/(float)3;

    dup();

    int x = 1;
    while(x < 5){
//      sleep(1);
```

Console | Memory | Registers

Name	Value	Descript
rbp	0x7fffffffde30	
rsp	0x7fffffffb1a0	
r8	4294967295	
r9	0	
r10	34	
r11	582	
r12	4199104	
r13	140737488346896	

POST-MORTEM TRACE

› Standard visualization of traces taken upon a crash

DSFTestApp.cpp | kernel | trace.DSFTestApp | kernel(2) | uid/44328/64-bit | kernel(3)

Timestamp	Channel	CPU	Event type	Contents	TID
<srch>	<srch>	<srch>	<srch>	<srch>	<srch
14:25:34.484 531 692	bob_3	3	exit_syscall	ret=140082361139200	
14:25:34.484 550 786	bob_3	3	sys_clone	clone_flags=0x411, newsp=0x7f6777affff0, parent_tid=0xffffffff, child_tid=0x7f6780aaff70	
14:25:34.484 560 208	bob_1	1	hrtimer_cancel	hrtimer=18446612143422552440	
14:25:34.484 562 884	bob_1	1	hrtimer_expire_entry	hrtimer=18446612143422552440, now=116853200707670, function=18446744072108273280	

Statistics

Global - kernel(3)

Level		Events total		Events in selection
kernel(3)		4,030,265		1
Event Types				
block_bio_backmerge	0 %	1,262	0 %	0
block_bio_complete	0.2 %	8,559	0 %	0
block_bio_frontmerg	0 %	2	0 %	0
block_bio_queue	0.3 %	12,865	0 %	0
block_bio_remap	0.2 %	8,598	0 %	0
block_dirty_buffer	10.8 %	437,616	0 %	0

Control Flow

Process	TID	PTID
init	1	
thunderbird	4092	1
Timer	4105	4092
dconf worker	4096	4092
gdbus	4097	4092
Gecko_IOThread	4098	4092
Socket Thread	4099	4092
JS GC Helper	4100	4092
JS Watchdog	4101	4092

14:25:34.500 14:25:34.550

Histogram | Bookmarks | Resources

2015 Feb 25, 14:25:34.490 14:25:34.500 14:25:34.510 14:25:34.520 14:25:34.530 14:25:34.540 14:25:34.550 14:25:34.560 14:25:34.570

kernel(3)
CPU 0
CPU 1
CPU 2
CPU 3
IRQ 9
IRQ 43
IRQ 44

CORE + TRACES

› Joint Debug/Tracing visualization for most flexibility

Debug

DSFTestApp Core [C/C++ Postmortem Debugger]
DSFTestApp
Thread #1 28327 (Suspended : Container)
dup() at DSFTestApp.cpp:203 0x4016fd
main() at DSFTestApp.cpp:244 0x4018b1
gdb.7.9 (7.9)

Variables

Name	Type	Value
argc	int	1
argv	char **	0x7fffffffdf18
b	int [5]	0x7fffffffdcc0
um	wchar_t	32767 L'翻'
array_small	int [4]	0x7fffffffdcf0
de	int	0
aba	int	0
shortarray	int [10]	0x7fffffffdc30

Control Flow

08:47:53.400

Process	TID	PTID
init	1	
upstart-udev-br	662	1
udevd	666	1
udevd	27575	666
udevd	27576	666
sshd	676	1
rsyslogd	696	1
rs:main Q:Reg	702	696

DSFTestApp.cpp

```
238     float myFloat[4];
239     myFloat[0] = 4.6;
240     myFloat[1] = 3.14;
241     myFloat[2] = 6;
242     myFloat[3] = (floa
243
244     dup();
245
246     int x = 1;
247     while(x < 5){
248 //      sleep(1);
```

kernel

Timestamp	Channel	CPU	Event type	Contents
<srch>	<srch>	<srch>	<srch>	<srch>
08:47:53.381 102 911	channel0_1	1	hrtimer_cancel	hrtimer=184466121
08:47:53.381 104 658	channel0_1	1	hrtimer_expire_entry	hrtimer=184466121
08:47:53.381 105 598	channel0_1	1	hrtimer_expire_exit	hrtimer=184466121
08:47:53.381 105 846	channel0_1	1	hrtimer_start	hrtimer=184466121
08:47:53.381 139 042	channel0_0	0	hrtimer_cancel	hrtimer=184466121
08:47:53.381 139 792	channel0_0	0	hrtimer_expire_entry	hrtimer=184466121

Statistics

Global - kernel

Level		Events total	Events i
block_bio_complete	0.1 %	8,351	0 %
block_bio_frontmerg	0 %	7	0 %
block_bio_queue	0.3 %	15,969	0 %
block_bio_remap	0.2 %	10,617	0 %
block_dirty_buffer	4.5 %	227,329	0 %

Histogram | Bookmarks | Resources

Selection Start 08:47:53.381 102 911
Selection End 08:47:53.381 102 911
Window Span 000.100 000 000

227
0
08:47:53.381 102 911
08:47:53.481 102 911

187754
0
08:47:53.381 102 911
08:48:20.414 944 384

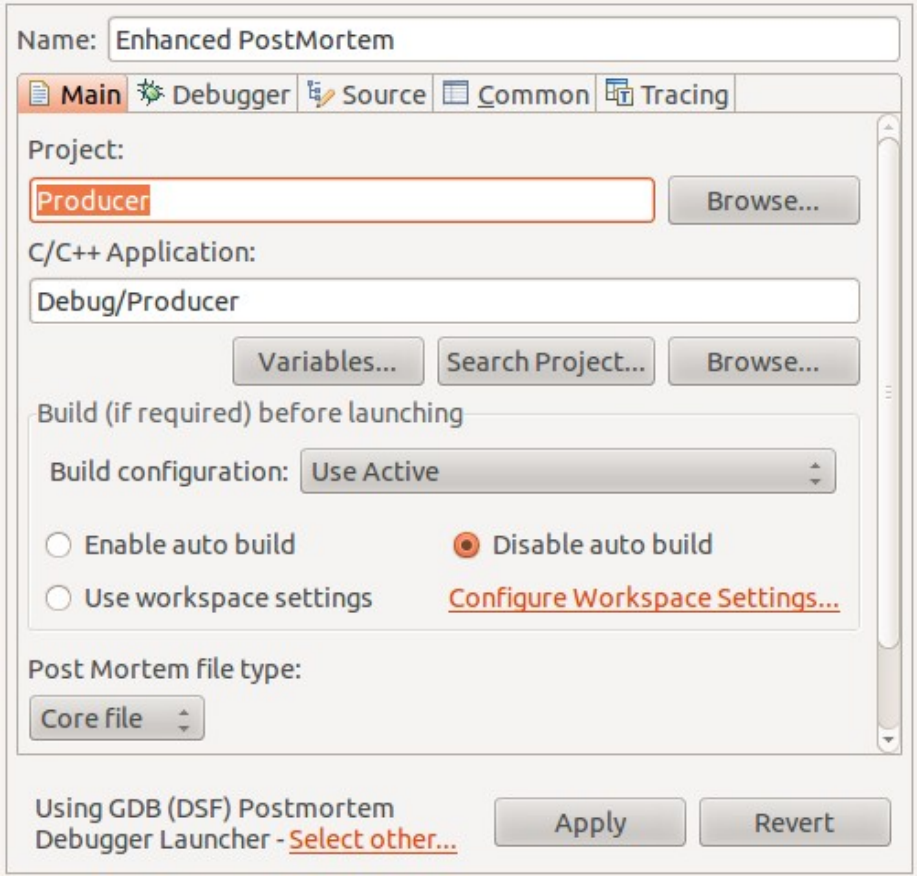
kernel

SYSTEM SETUP

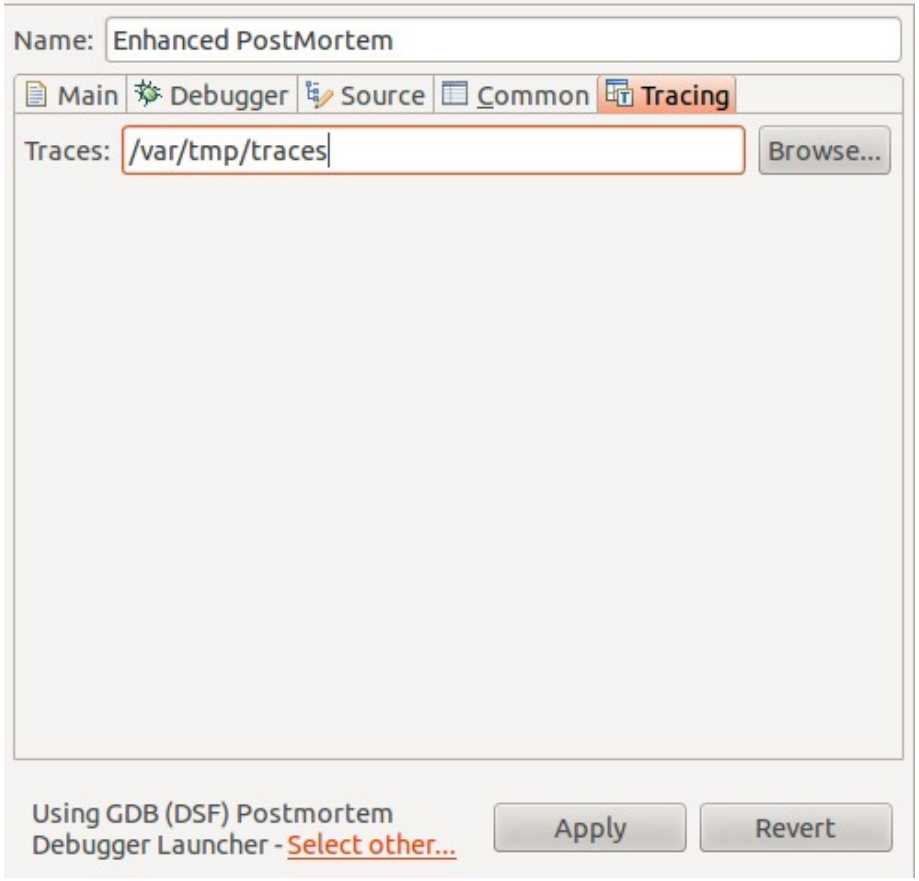
1. Enable Tracing e.g., LTTng, UST, etc
2. Register crash handler with Linux kernel (man core)
3. Crash Handler collects/stores traces as well as core file

DEBUG LAUNCH

1) Use Post-Mortem launch



2) Specify location of Traces



CORE + TRACES

Debug

- DSFTestApp Core [C/C++ Postmortem Debugger]
  - DSFTestApp
    - Thread #1 28327 (Suspended : Container)
      - dup() at DSFTestApp.cpp:203 0x4016fd
      - main() at DSFTestApp.cpp:244 0x4018b1
  - gdb.7.9 (7.9)

Variables

Name	Type	Value
argc	int	1
argv	char **	0x7fffffffdf18
b	int [5]	0x7fffffffdcc0
um	wchar_t	32767 L'翻'
array_small	int [4]	0x7fffffffdcf0
de	int	0
aba	int	0
shortarray	int [10]	0x7fffffffdc30

Control Flow

08:47:53.400

Process	TID	PTID
init	1	
upstart-udev-br	662	1
udevd	666	1
udevd	27575	666
udevd	27576	666
sshd	676	1
rsyslogd	696	1
rs:main Q:Reg	702	696

DSFTestApp.cpp

```
238     float myFloat[4];
239     myFloat[0] = 4.6;
240     myFloat[1] = 3.14;
241     myFloat[2] = 6;
242     myFloat[3] = (floa
243
244     dup();
245
246     int x = 1;
247     while(x < 5){
248 //      sleep(1);
```

kernel

Timestamp	Channel	CPU	Event type	Contents
<srch>	<srch>	<srch>	<srch>	<srch>
08:47:53.381 102 911	channel0_1	1	hrtimer_cancel	hrtimer=1844661214
08:47:53.381 104 658	channel0_1	1	hrtimer_expire_entry	hrtimer=1844661214
08:47:53.381 105 598	channel0_1	1	hrtimer_expire_exit	hrtimer=1844661214
08:47:53.381 105 846	channel0_1	1	hrtimer_start	hrtimer=1844661214
08:47:53.381 139 042	channel0_0	0	hrtimer_cancel	hrtimer=1844661214
08:47:53.381 139 792	channel0_0	0	hrtimer_expire_entry	hrtimer=1844661214

Statistics

Global - kernel

Level		Events total	Events i
block_bio_complete	0.1 %	8,351	0 %
block_bio_frontmerg	0 %	7	0 %
block_bio_queue	0.3 %	15,969	0 %
block_bio_remap	0.2 %	10,617	0 %
block_dirty_buffer	4.5 %	227,329	0 %

Histogram | Bookmarks | Resources

Selection Start 08:47:53.381 102 911
Selection End 08:47:53.381 102 911
Window Span 000.100 000 000

227
0
08:47:53.381 102 911
08:47:53.481 102 911

187754
0
08:47:53.381 102 911
08:48:20.414 944 384

kernel

AGENDA

- A bit of background: Debug and Tracing

- CDT Debug and Trace Compass integration

  1. Enhanced Post-mortem troubleshooting
  2. **Debugging with Trace snapshots**
  3. Tracing with the (Multicore) Visualizer
  4. GDB Traces with Trace Compass

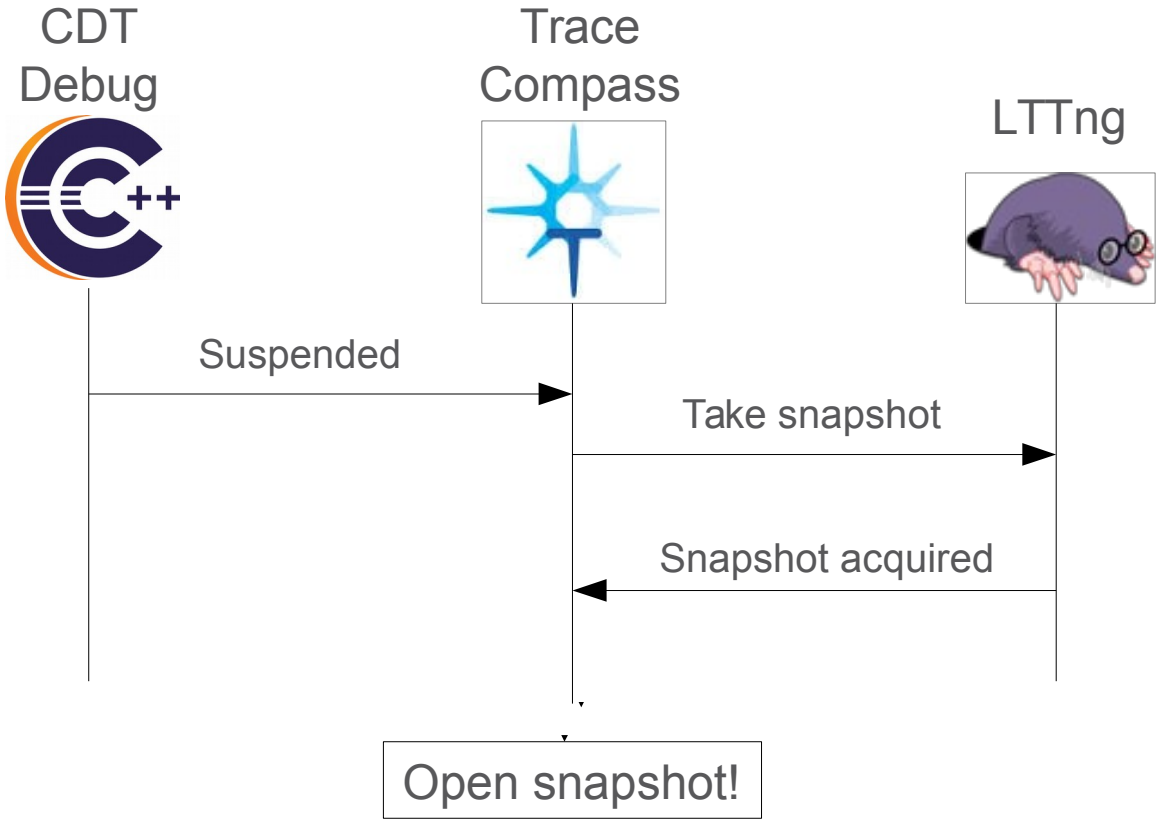
- Conclusion

ERICSSON

DEBUGGING WITH TRACE SNAPSHOTS

DEBUGGING WITH TRACE SNAPSHOTS

› Acquire snapshot and open on suspended debugger

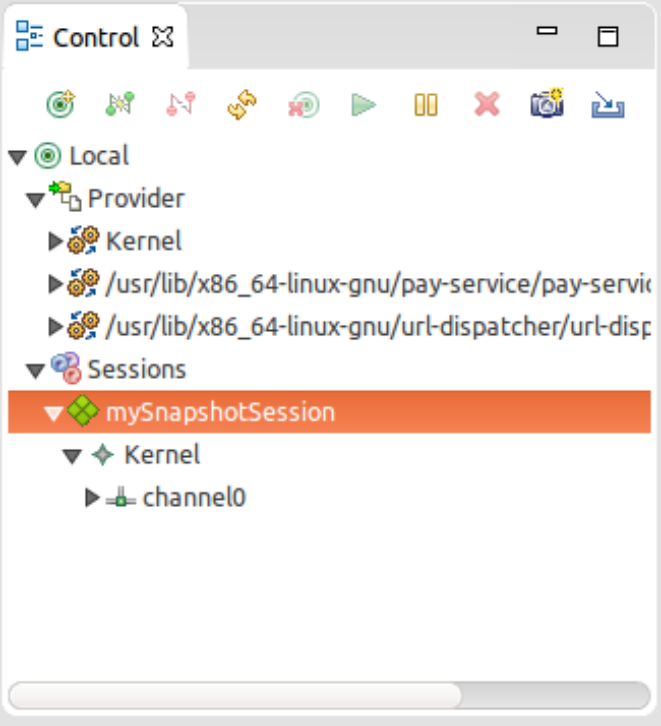


DEBUGGING WITH TRACE SNAPSHOTS

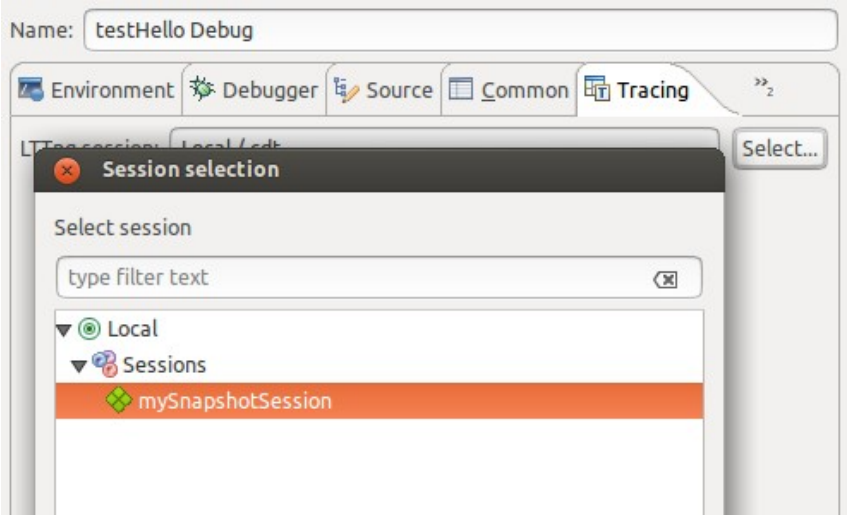
- Advantages:
  - Very low overhead
  - Minimal disk usage
- Disadvantage:
  - Limited data available (as big as buffer allows)

THE PROTOTYPE

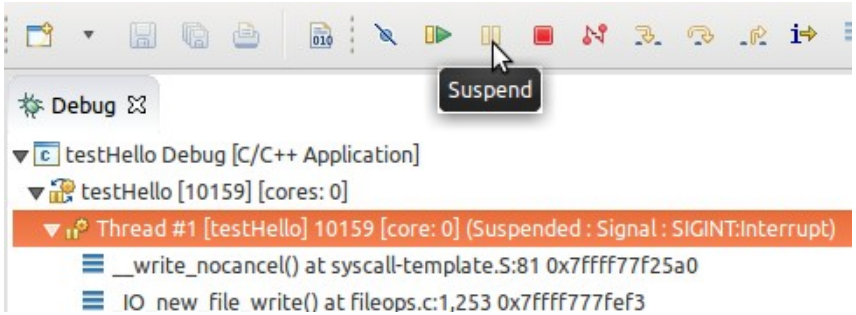
1) Create a tracing session



2) Select session in Debug configuration



3) Suspend (or hit a breakpoint)



THE PROTOTYPE

Debug

testHello [10159] [cores: 0]

Thread #1 [testHello] 10159 [core: 0] (Suspended : Signal : SIGINT:Interrupt)

__write_nocancel() at syscall-template.S:81 0x7ffff77f25a0

_IO_new_file_write() at fileops.c:1,253 0x7ffff777fef3

new_do_write() at fileops.c:530 0x7ffff77813cc

_IO_new_do_write() at fileops.c:503 0x7ffff77813cc

Histogram

Selection Start 22:45:16.112 427 037

Selection End 22:45:16.112 427 037

Window Span 000.000 050 378

1

0

22:45:16.112 400 170 22:45:16.112 450 548

1706

mySnapshotSession-20150302-223735/snapshot-1-20150302-224516-1/kernel

mySnapshotSession-20150302-223735/snapshot-1-20150302-224516-1/kernel

Timestamp	Channel	CPU	Event type	Contents
<srch>	<srch>	<srch>	<srch>	<srch>
22:45:16.112 426 367	channel0_1	1	sched_wakeup	comm=kworker/1:0, tid=29
22:45:16.112 427 037	channel0_1	1	sched_switch	prev_comm=testHello, prev
22:45:16.112 428 049	channel0_1	1	sched_switch	prev_comm=kworker/1:0, p
22:45:16.112 428 601	channel0_3	3	sched_wakeup	comm=testHello, tid=1015
22:45:16.112 429 073	channel0_1	1	sched_wakeup	comm=kworker/1:0, tid=29
22:45:16.112 429 892	channel0_1	1	sched_switch	prev_comm=testHello, prev

testHello.cpp

```
#include <iostream>
#include <unistd.h>
using namespace std;

int main() {
    while (true) {
        cout << "!!!Hello World!!!" << endl;
    }
    return 0;
}
```

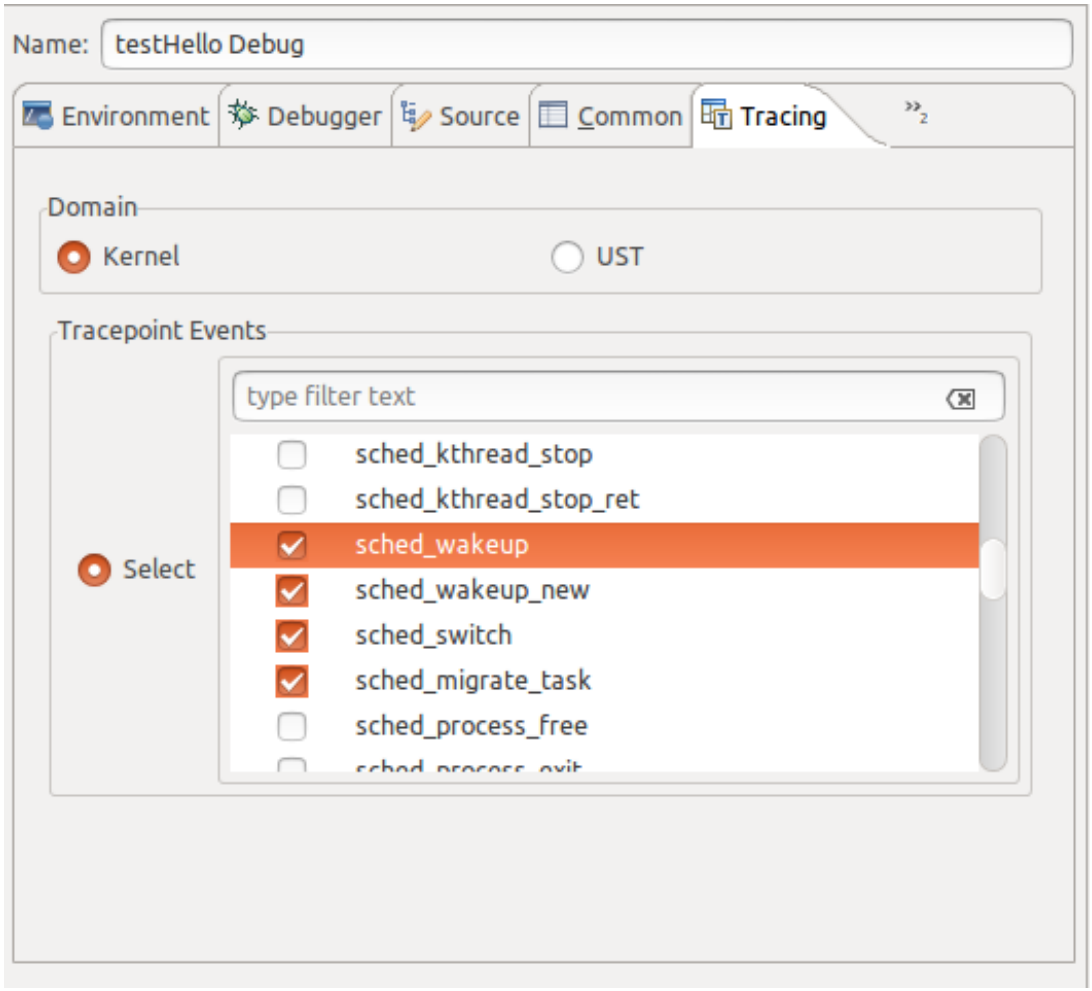
Console Tasks Problems Executables Memory Control Control Flow

22:45:16.112420

Process	TID	PTID	Birth time	Trace
java	10155		22:45:15.918050669	mySnapshotSession-2
java	10158		22:45:15.988630127	mySnapshotSession-2
testHello	10159		22:45:16.076495261	mySnapshotSession-2
soffice.bin	16920		22:45:16.168158193	mySnapshotSession-2
kworker/3:1	23670		22:45:16.088200769	mySnapshotSession-2
java	24507		22:45:16.103509665	mySnapshotSession-2
java	24512		22:45:16.189503760	mySnapshotSession-2

FUTURE IMPROVEMENTS

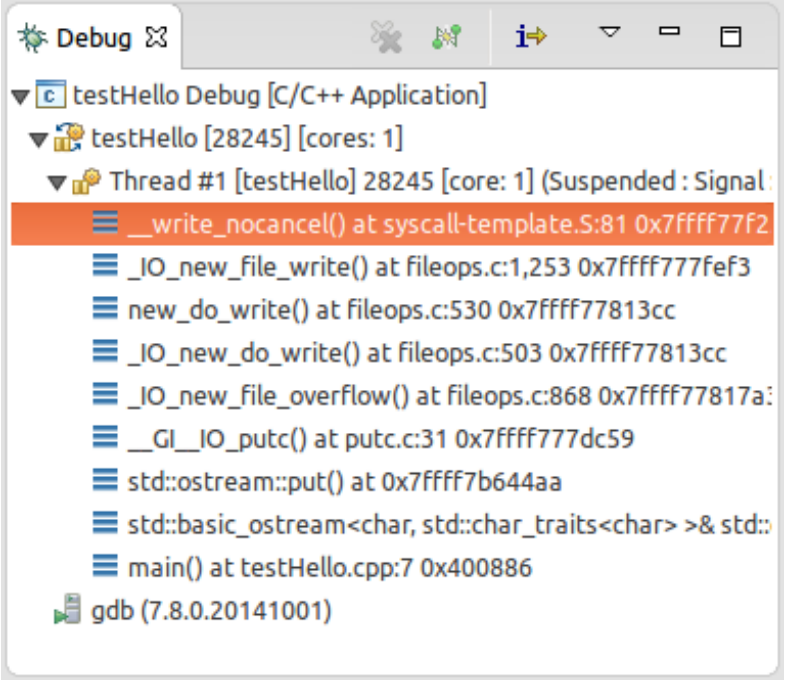
- Configure session from Debug configuration
  - Choose tracer
  - Choose trace points
  - Tracer specific options
  - Persisted



FUTURE IMPROVEMENTS

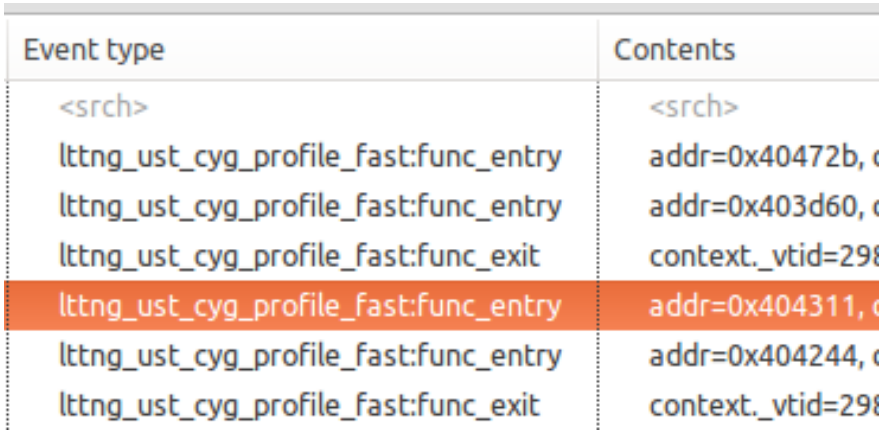
› Callstacks of the last few seconds

Current stack frames (GDB)



+

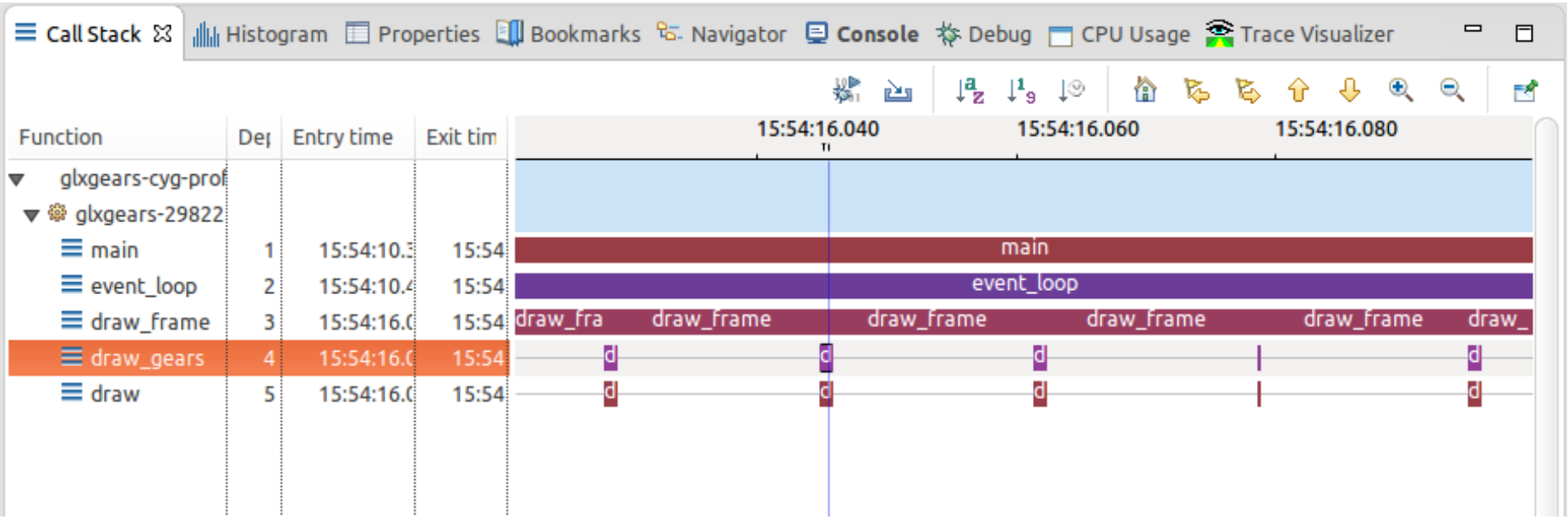
Previous events with function entry and exit (LTTng snaphot)



Event type	Contents
<srch>	<srch>
lttng_ust_cyg_profile_fast:func_entry	addr=0x40472b,
lttng_ust_cyg_profile_fast:func_entry	addr=0x403d60,
lttng_ust_cyg_profile_fast:func_exit	context._vtid=298
lttng_ust_cyg_profile_fast:func_entry	addr=0x404311,
lttng_ust_cyg_profile_fast:func_entry	addr=0x404244,
lttng_ust_cyg_profile_fast:func_exit	context._vtid=298

FUTURE IMPROVEMENTS

Result (example)



Callstack can be visualized moments before suspend!

AGENDA

- A bit of background: Debug and Tracing

- CDT Debug and Trace Compass integration

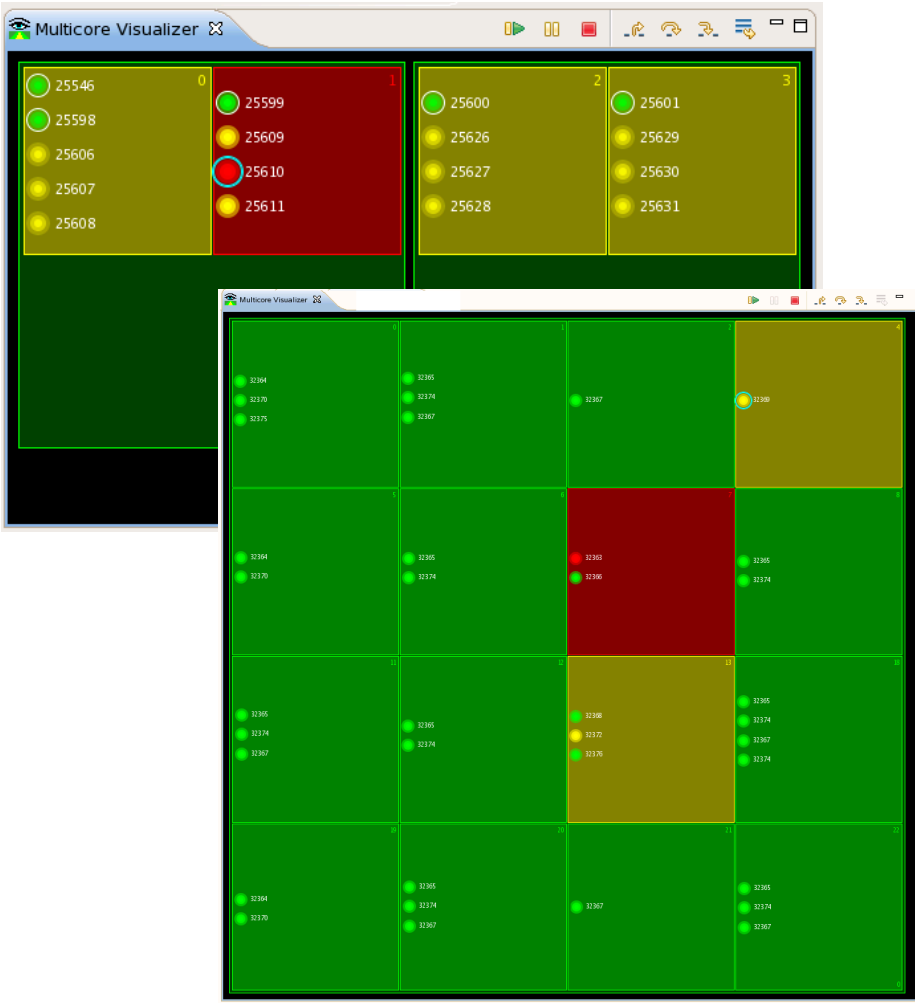
  1. Enhanced Post-mortem troubleshooting
  2. Debugging with Trace snapshots
  3. **Tracing with the (Multicore) Visualizer**
  4. GDB Traces with Trace Compass

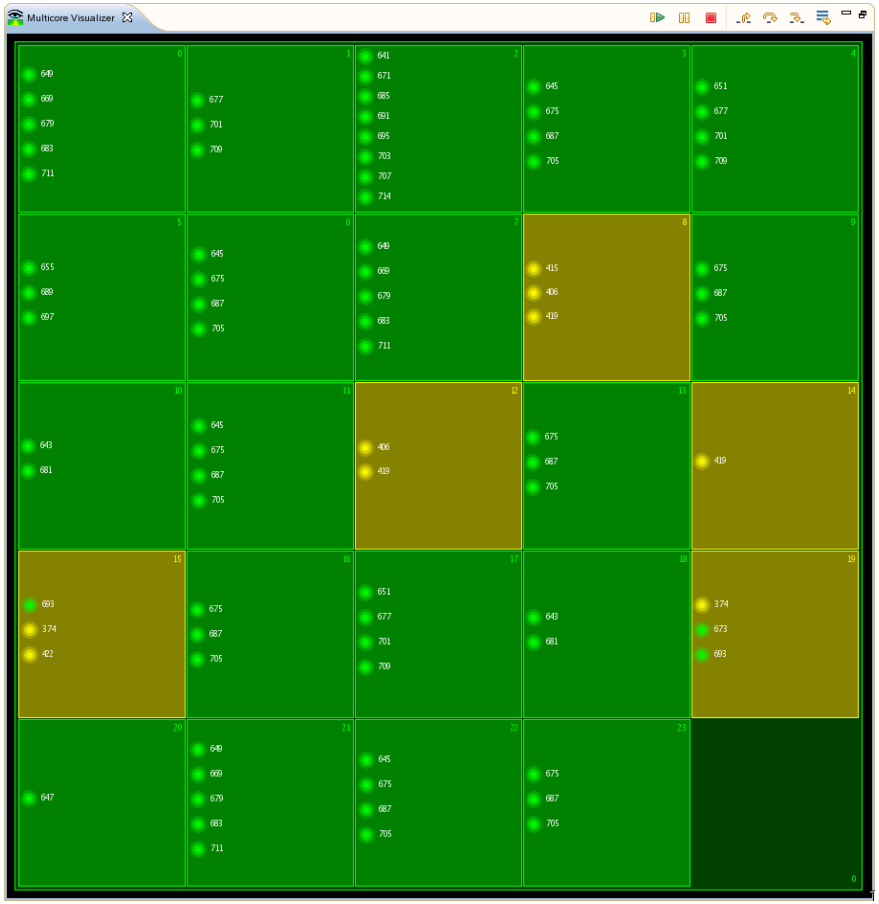
- Conclusion

ERICSSON

TRACE (MULTICORE) VISUALIZER

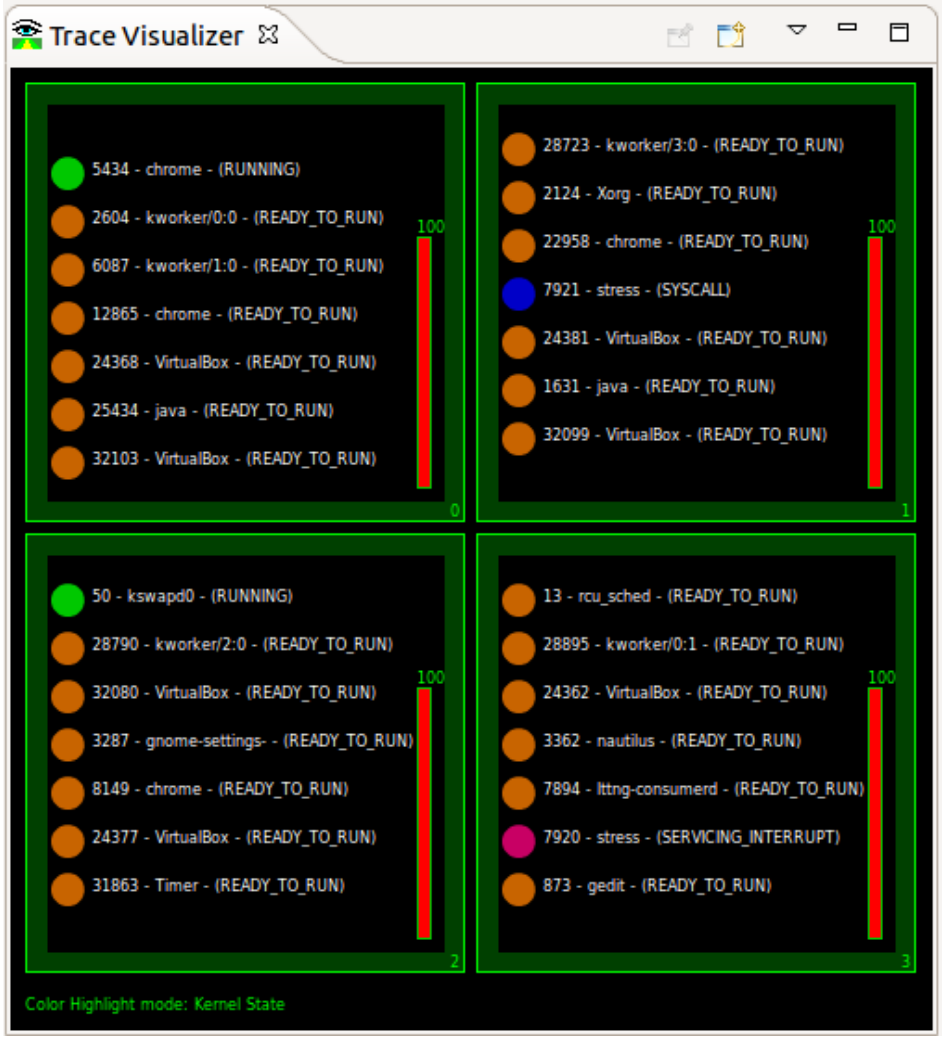
MULTICORE VISUALIZER





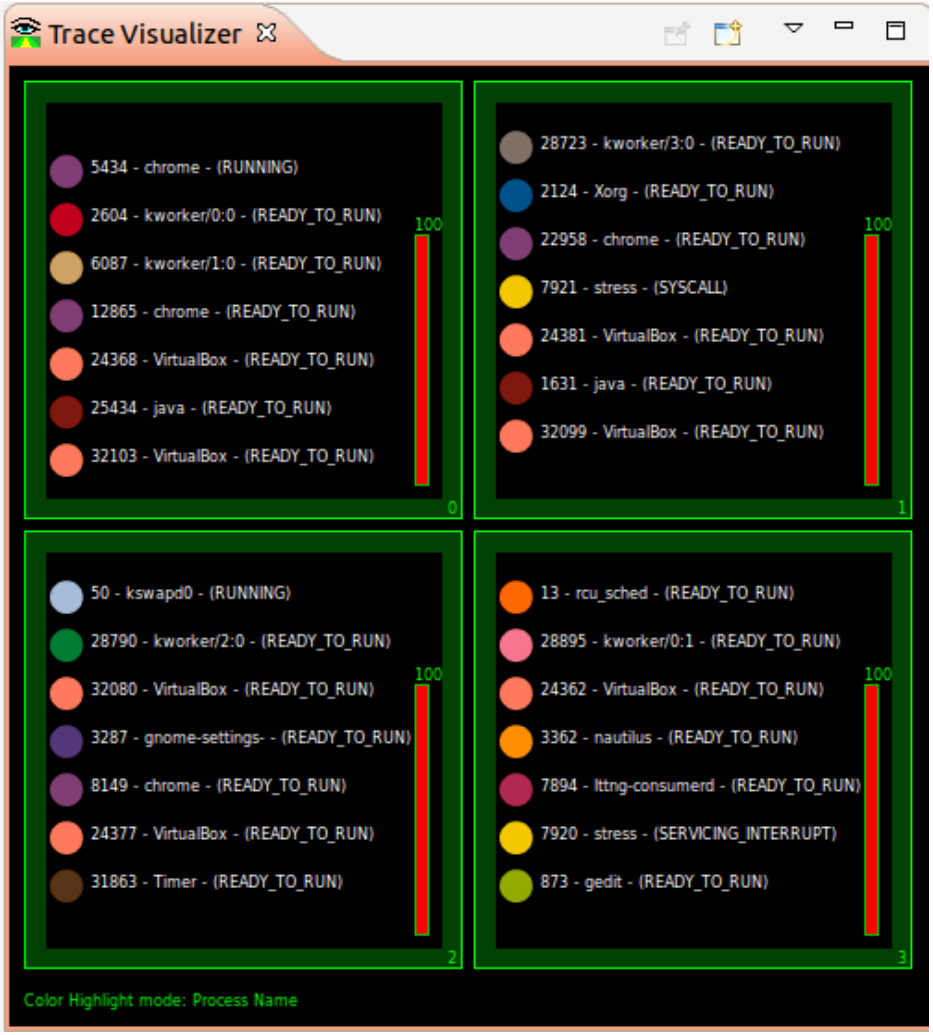
TRACE VISUALIZER

- Show all threads *except* sleeping
  - All of them *could* run
- Coloured by kernel state
- CPU Usage
- We can have a better grasp of level of overload
- Which processes are affected by the overload?



TRACE VISUALIZER

- Colouring by process
- Sorting as improvement



TRACE COMPASS AND TRACE VISUALIZER

Project Explo
- DemoTracef
- DSFTestApp
- PostMortemTraces
- Remote
- StressTrace
  - Experiments [0]
  - Traces [3]
    - kernel
    - kernel(2)
    - kernel(3)
- TraceDSFTestApp

uid/44328/64-bit | kernel | kernel | TraceVis1-20150225-131319/kernel | kernel(2) | DSFTestApp.cpp | kernel(3)

Timestamp	Channel	CPU	Event type	Contents
<srch>	<srch>	<srch>	<srch>	<srch>
08:47:53.535 740 275	channel0_0	0	hrtimer_expire_exit	hrtimer=18446612140259789944
08:47:53.535 740 394	channel0_0	0	hrtimer_start	hrtimer=18446612140259789944, function=18446744072105967232, expires=3292188561031888, softexpires=3292188561031888

Control Flow

Process	TID	PTID	Birth time	Trace
unity-panel-ser	3481	1	08:47:53.384003494	kernel
dconf worker	3484	3481	08:47:53.384004425	kernel
gdbus	3487	3481	08:47:53.384005582	kernel
hud-service	3483	1	08:47:53.384006285	kernel
dconf worker	3485	3483	08:47:53.384007031	kernel

08:47:53.400 08:47:53.450 08:47:53.500

Statistics

Global - kernel

Level		Events total	Events
block_rq_requeue	0 %	379	0 %
block_sleeprq	0 %	6	0 %
block_touch_buffer	9.1 %	455,860	0 %
block_unplug	0 %	174	0 %
exit_syscall	8.1 %	402,723	0 %
hrtimer_cancel	1.8 %	93,137	100 %
hrtimer_expire_entry	1.3 %	67,849	0 %
hrtimer_expire_exit	1.3 %	67,847	0 %

Trace Visualizer

2604 - kworker/0:0 - (RUNNING)
28723 - kworker/3:0 - (READY_TO_RUN)
3362 - nautilus - (READY_TO_RUN)
7921 - stress - (READY_TO_RUN)
7925 - stress - (READY_TO_RUN)
12865 - chrome - (READY_TO_RUN)

28790 - kworker/2:0 - (READY_TO_RUN)
4032 - update-notifier - (READY_TO_RUN)
8149 - chrome - (READY_TO_RUN)
22958 - chrome - (READY_TO_RUN)
7922 - stress - (READY_TO_RUN)
7926 - stress - (RUNNING)
7927 - stress - (READY_TO_RUN)

5434 - chrome - (READY_TO_RUN)
50 - kswapd0 - (RUNNING)
32080 - VirtualBox - (READY_TO_RUN)
7306 - lttng-sessiond - (READY_TO_RUN)
7916 - stress - (READY_TO_RUN)
7923 - stress - (READY_TO_RUN)

3287 - gnome-settings- - (READY_TO_RUN)
1432 - chrome - (READY_TO_RUN)
7918 - stress - (READY_TO_RUN)
7920 - stress - (RUNNING)
7924 - stress - (READY_TO_RUN)
31863 - Timer - (READY_TO_RUN)

Color Highlight mode: Kernel State

Histogram | Properties | Bookmarks

Selection Start 08:47:53.536 098 611
Selection End 08:47:53.536 098 611
Window Span 000.195 312 500

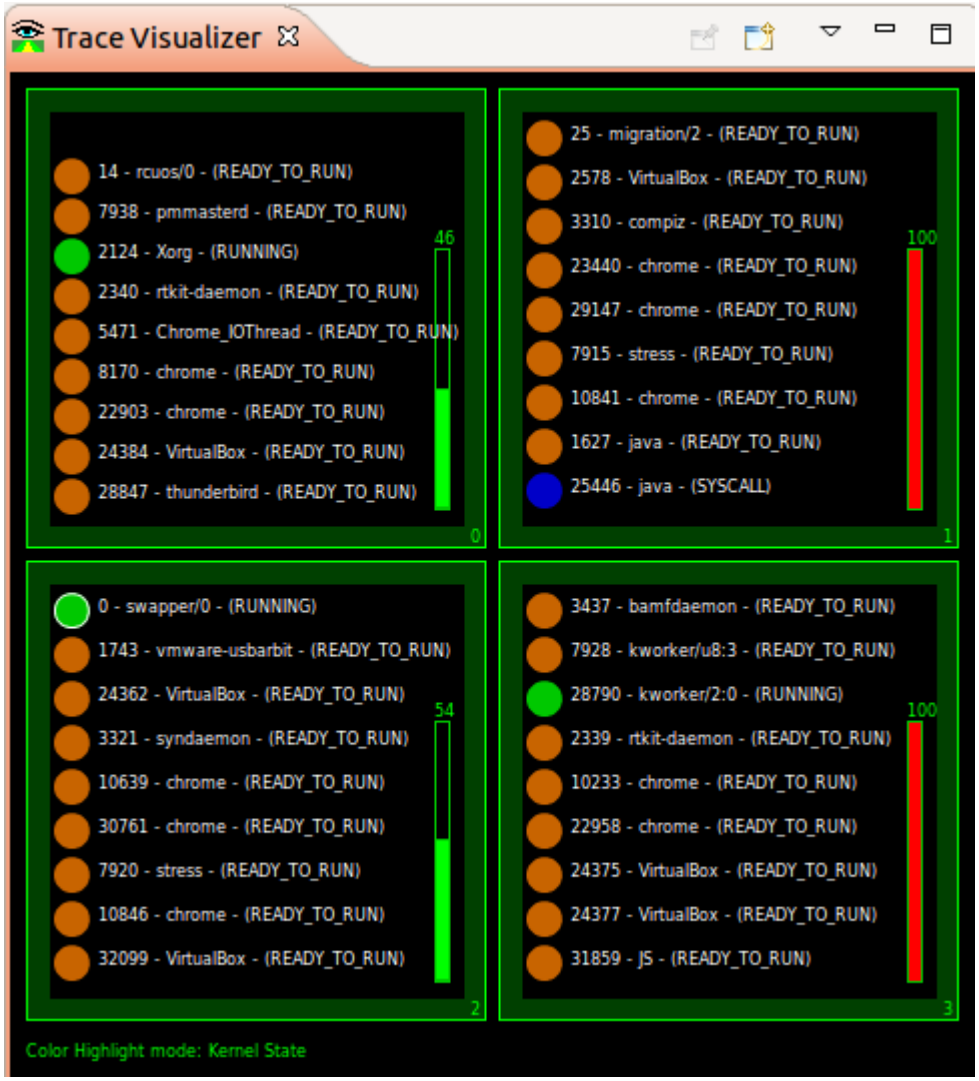
438
0
08:47:53.381 102 911
08:47:53.576 415 411

184659
0
08:47:53.381 102 911
08:48:20.414 944 384

kernel

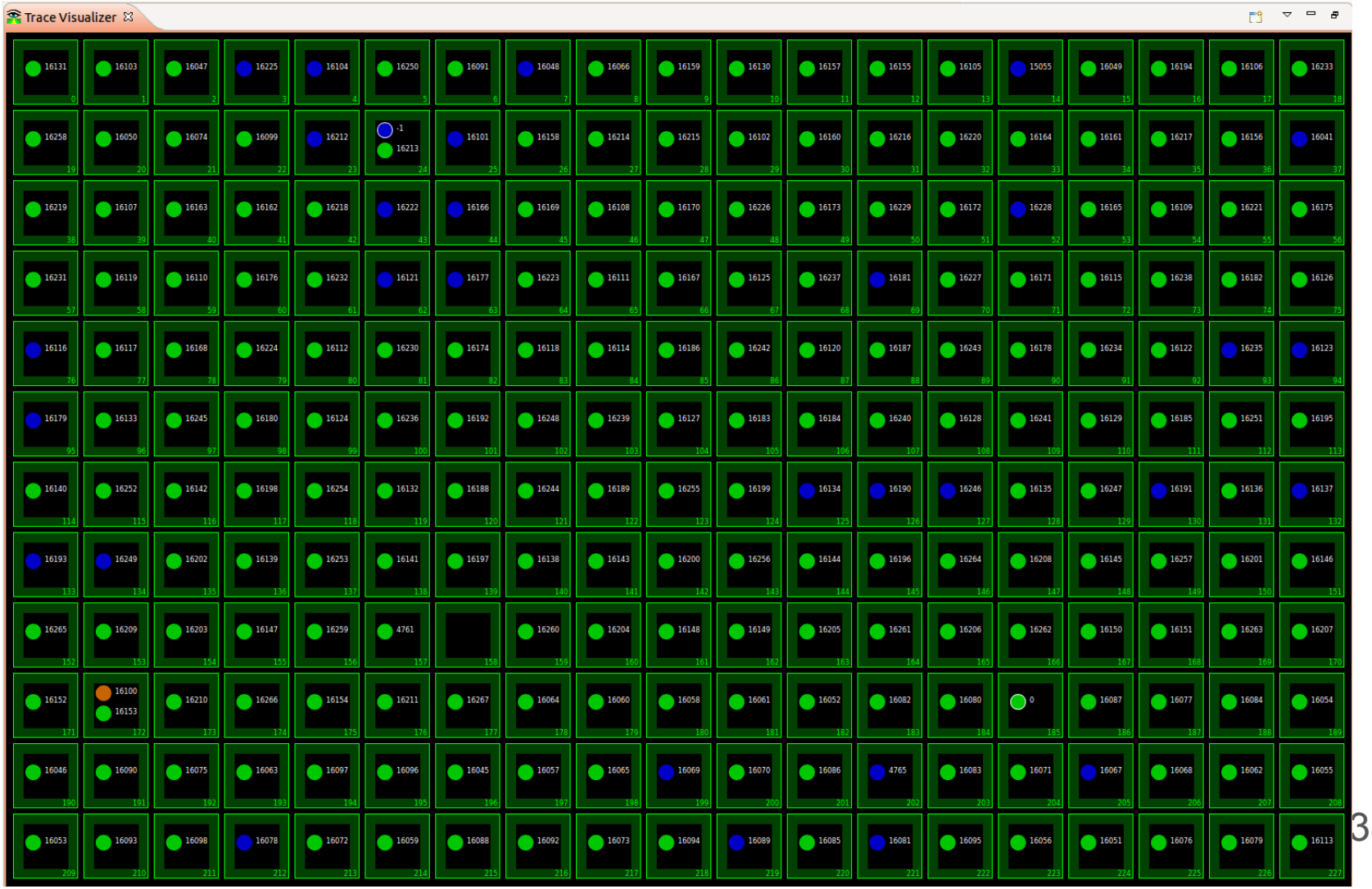
TRACE VISUALIZER

- Another example
- Notice partial CPU usage even with overload
- Could it be the Kernel using CPU?
- Could indicate even stronger overload



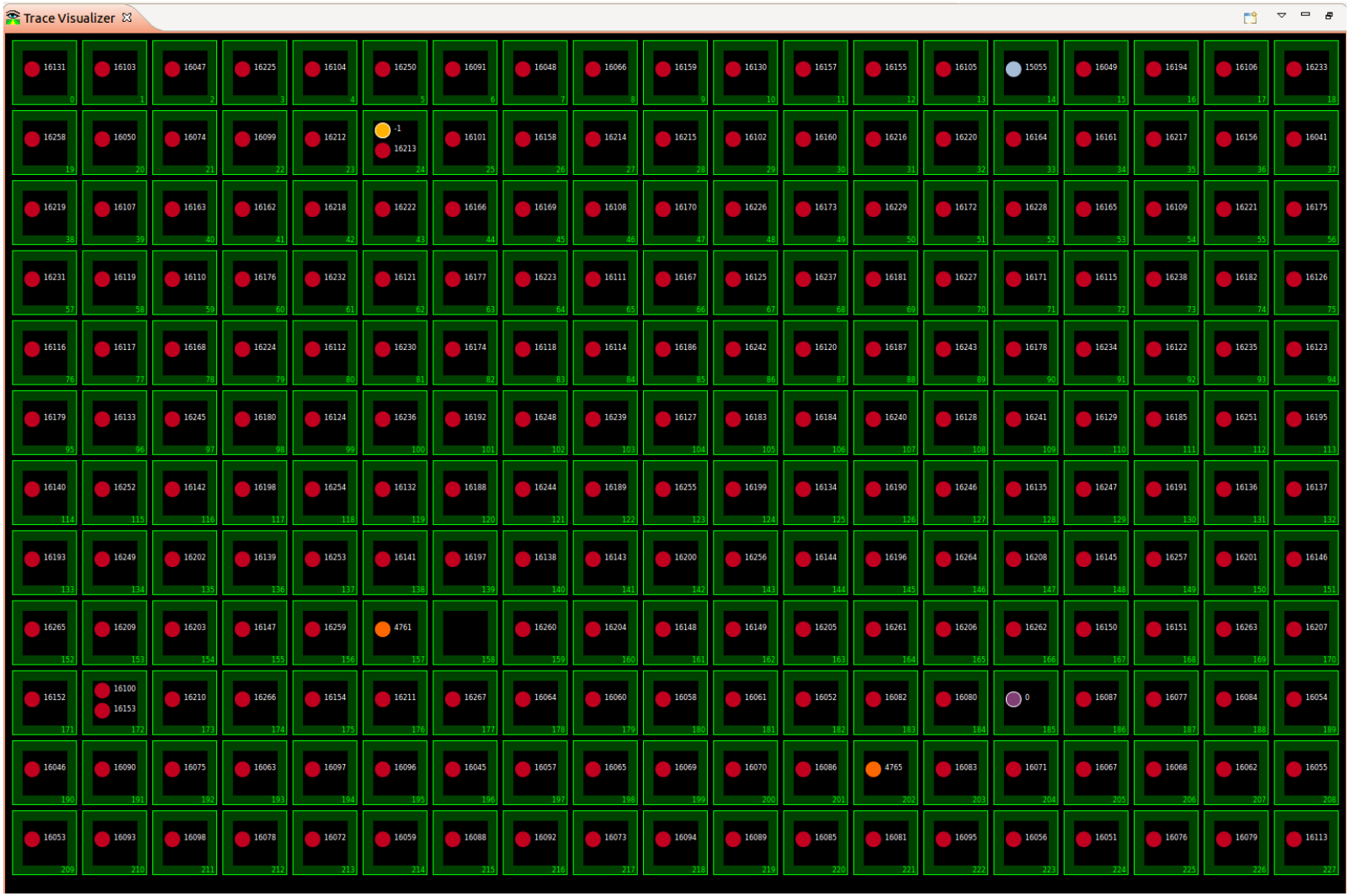
VISUALIZER WITH XEON PHI

› Coloured by kernel state (RUNNING & SYSCALL)



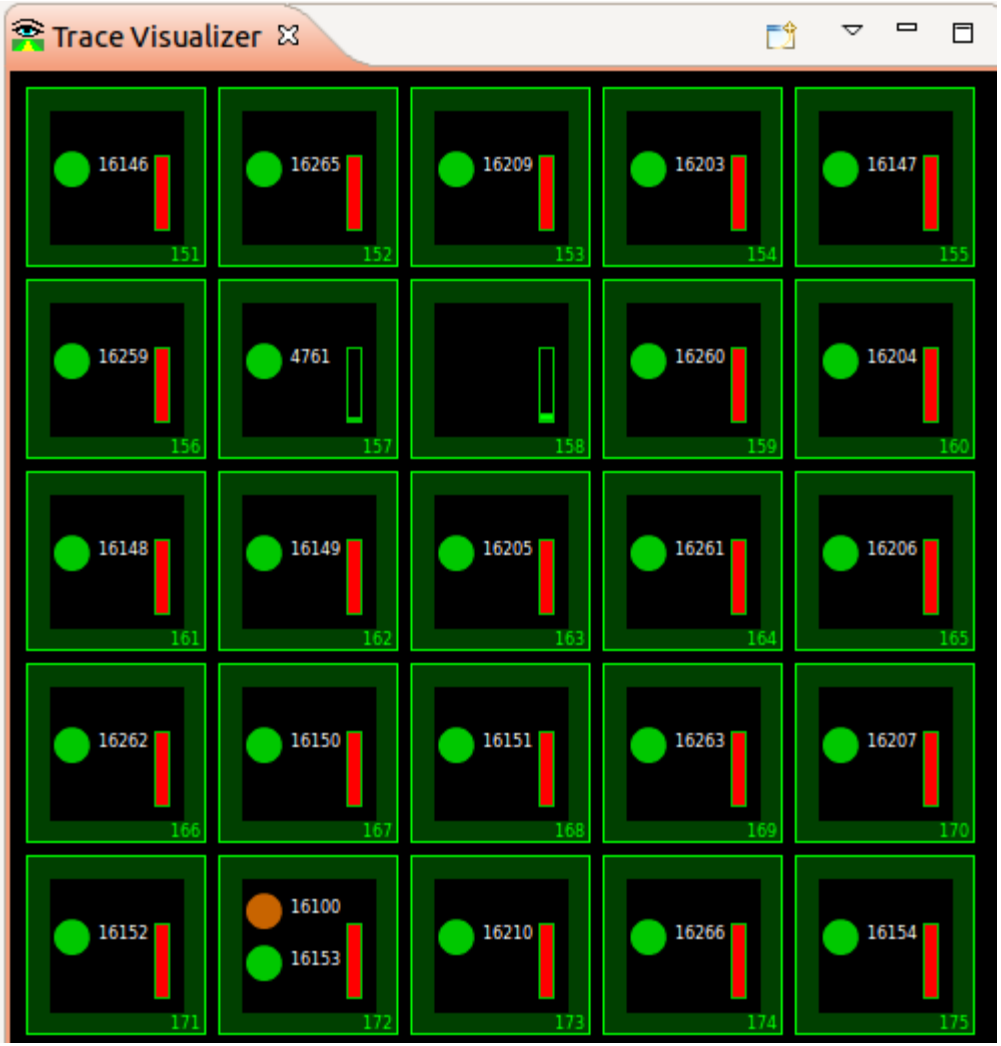
VISUALIZER WITH XEON PHI

› Coloured by process



VISUALIZER WITH XEON PHI

› Filtering of cores to display



AGENDA

- A bit of background: Debug and Tracing

- CDT Debug and Trace Compass integration

  1. Enhanced Post-mortem troubleshooting
  2. Debugging with Trace snapshots
  3. Tracing with the (Multicore) Visualizer
  4. **GDB Traces with Trace Compass**

- Conclusion

ERICSSON

GDB TRACES WITH TRACE COMPASS

GDB TRACEPOINTS

› Instrumentation, collection and visualization in CDT

```
112     if ((sd = socket(PF_INET, SOCK_DGRAM, 0)) < 0) {
113         perror("Socket");
114         abort();
115     }
116
117     addr.sin_family = AF_INET;
118     addr.sin_port = htons(10010);
119     if (inet_aton("127.0.0.1", &addr.sin_addr) == 0) {
120         perror("127.0.0.1");
121         abort();
122     }
123
124     int delay = 0;
```

Toggle Breakpoint
Enable Breakpoint
Breakpoint Properties...
Breakpoint Types
Go to Annotation Ctrl+G
Add Bookmark...
Add Task...
Show Quick Diff Shift+Ctrl+Q
Show Line Numbers
Folding
Preferences...

C/C++ Breakpoints
C/C++ Tracepoints

DEBUG GDB TRACES

Debug - DSFTestApp/src/DSFTestApp.cpp -

File Edit Source Refactor Navigate Search Project Run Window Help

Debug

DSFTestApp Trace [C/C++ Remote Application]
DSFTestApp [3347] [cores: 3]
Thread #1 3347 [core: 3] (Suspended : Tracepoint 1, Record 80)
main() at DSFTestApp.cpp:235 0x40184d
gdb.7.9 (7.9)

Variables | Registers | Modules

Name	Type	Value
argv	char **	<unavailable>
i	int	80
b	int [5]	0x7fffffffe0a0

Collected Data

DSFTestApp.cpp | kernel

```
int step = secondScope(firstScope()) + samename();

testmethod();

int thatIsABigArray[200];

for (int i = 0; i < 200; ++i) {
    thatIsABigArray[i] = i;
}

float myFloat[4];
myFloat[0] = 4.6;
myFloat[1] = 3.14;
myFloat[2] = 6;
myFloat[3] = (float)1/(float)3;

dup();

int x = 1;
```

Line where trace was collected

Breakpoints

DSFTestApp.cpp [line: 241]
DSFTestApp.cpp [line: 235]

Tracepoint that was hit

Outline | Trace Control

Examining trace End

5s trace by lmckhou, stopped 4:31 PM today on user request

Buffer

47%

Circular OFF

200 frames, 2468Kb

Current selected trace frame: 80, tracepoint 1

GDB TRACES EVENT TABLE

› Synchronized Trace Compass's Events Table

GDB Trace - GDBTraces/Traces/gdb.trace -

File Edit Navigate Search Project Run Window Help

Quick Access | Resource | C/C++ | Debug | GDB Trace | Tracing | LTTng Kernel

Project Explorer

- DemoTracef
- DSFTestApp
- GDBTraces
  - Experiments [0]
  - Traces [1]
    - gdb.trace
- PostMortemTraces
- Remote
- StressTrace

Debug

- gdb.trace [C/C++ Postmortem Debugger]
  - DSFTestApp
    - Thread #1 1 (Suspended : Tracepoint 2, Record 111)
      - dup() at DSFTestApp.cpp:204 0x401704
      - 0x0
  - gdb (7.4)

Trace Control

Examining trace data from a file

Tracing was started 9:22 AM today by lmckhou, stopped 9:22 AM today on user request

Buffer 2% Circular OFF

160 frames, 132Kb

Current selected trace frame: 111, tracepoint 2

Console

gdb.trace [C/C++ Postmortem Debugg

```
GNU gdb (Ubuntu/Linaro 7.4-20
Copyright (C) 2012 Free Softw
License GPLv3+: GNU GPL versi
This is free software: you ar
There is NO WARRANTY, to the
and "show warranty" for detai
This GDB was configured as "x
For bug reporting instruction
<http://bugs.launchpad.net/gdl
Created tracepoint 1 for targ
Created tracepoint 2 for targ
Warning: the current language
```

DSFTestApp.cpp

```
200
201     for (int i=0; i<80; i++)
202     {
203         int myint = 55;
204         myint++;
205     }
```

kernel | trace.DSFTestApp | uid/44328/64-bit | kernel(3) | kernel | gdb.trace

Trace Frame	Tracepoint	File	Contents
<srch>	<srch>	<srch>	<srch>
111	2	../src/DSFTestApp.cpp:204 :: dup()	Data collected at tracepoint 2, trace frame 111: | ra
112	1	../src/DSFTestApp.cpp:203 :: dup()	Data collected at tracepoint 1, trace frame 112: | my
113	2	../src/DSFTestApp.cpp:204 :: dup()	Data collected at tracepoint 2, trace frame 113: | ra
114	1	../src/DSFTestApp.cpp:203 :: dup()	Data collected at tracepoint 1, trace frame 114: | my

Trace Visualizer

CONCLUSION

MULTICORE DEBUG GROUP

- Joint effort to bring multicore debugging to the CDT
  - Visualizer, Pin&Clone, Multiprocess, etc
- Support for those that want to add new features
- Monthly conference calls (open to all interested and free ☺)
  - http://wiki.eclipse.org/CDT/MultiCoreDebugWorkingGroup

MORE ON TRACING

- Learn more about tracing and Trace Compass:
- Thursday 12 noon in Harbour AB with Marc-Andre:
  - “Analyzing Eclipse Applications with Trace Compass”

Thursday, March 12, 2015

	Grand Peninsula D	Grand Peninsula EFG	Harbour AB
12:00 - 12:35			PolarSys Day Marc-Andre Laperle [Ericsson]

SOME REFERENCES

- Integration on GitHub, https://github.com/MarkZ3/Trace-Compass/tree/dsf-mv-integration

- CDT Project, http://www.eclipse.org/cdt
- Trace Compass, https://projects.eclipse.org/projects/tools.tracecompass

- CDT FAQ, http://wiki.eclipse.org/CDT/User/FAQ
- CDT Debug workgroup http://wiki.eclipse.org/CDT/MultiCoreDebugWorkingGroup
- CDT Wiki, http://wiki.eclipse.org/CDT

ERICSSON

FEEDBACK

Evaluate the sessions

Sign in: www.eclipsecon.org

FINAL Q&A

ERICSSON

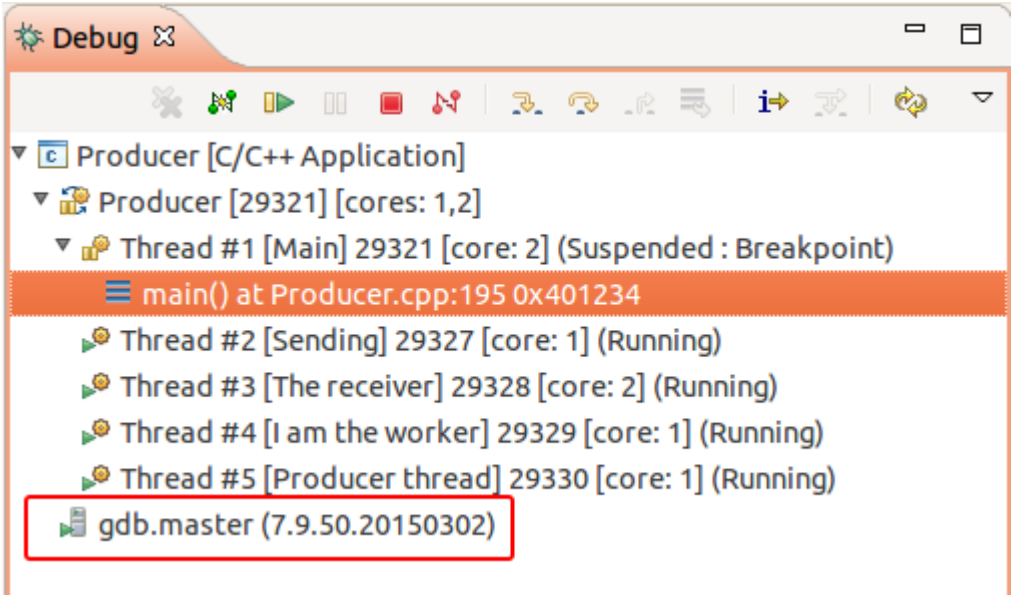
BONUS SLIDES

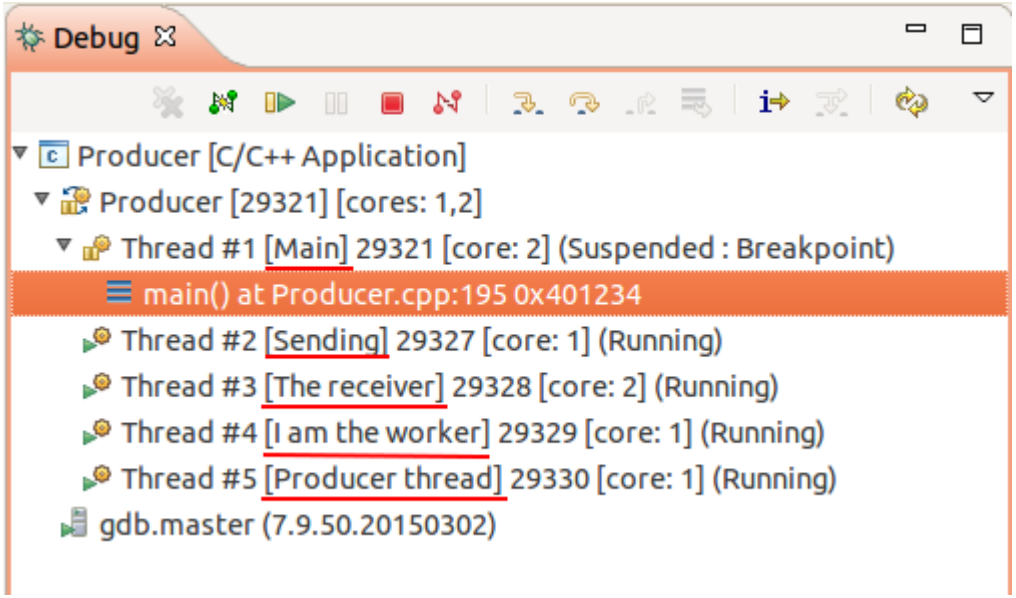
ERICSSON

OTHER CDT DEBUG NEWS

DEBUG VIEW LABELS

› GDB binary name/version





› Thread Names

PER-ELEMENT FORMAT

› Ability to set format per element

› Variables, Expressions, Registers views

Breakpoints | Expressions | Variables | Registers

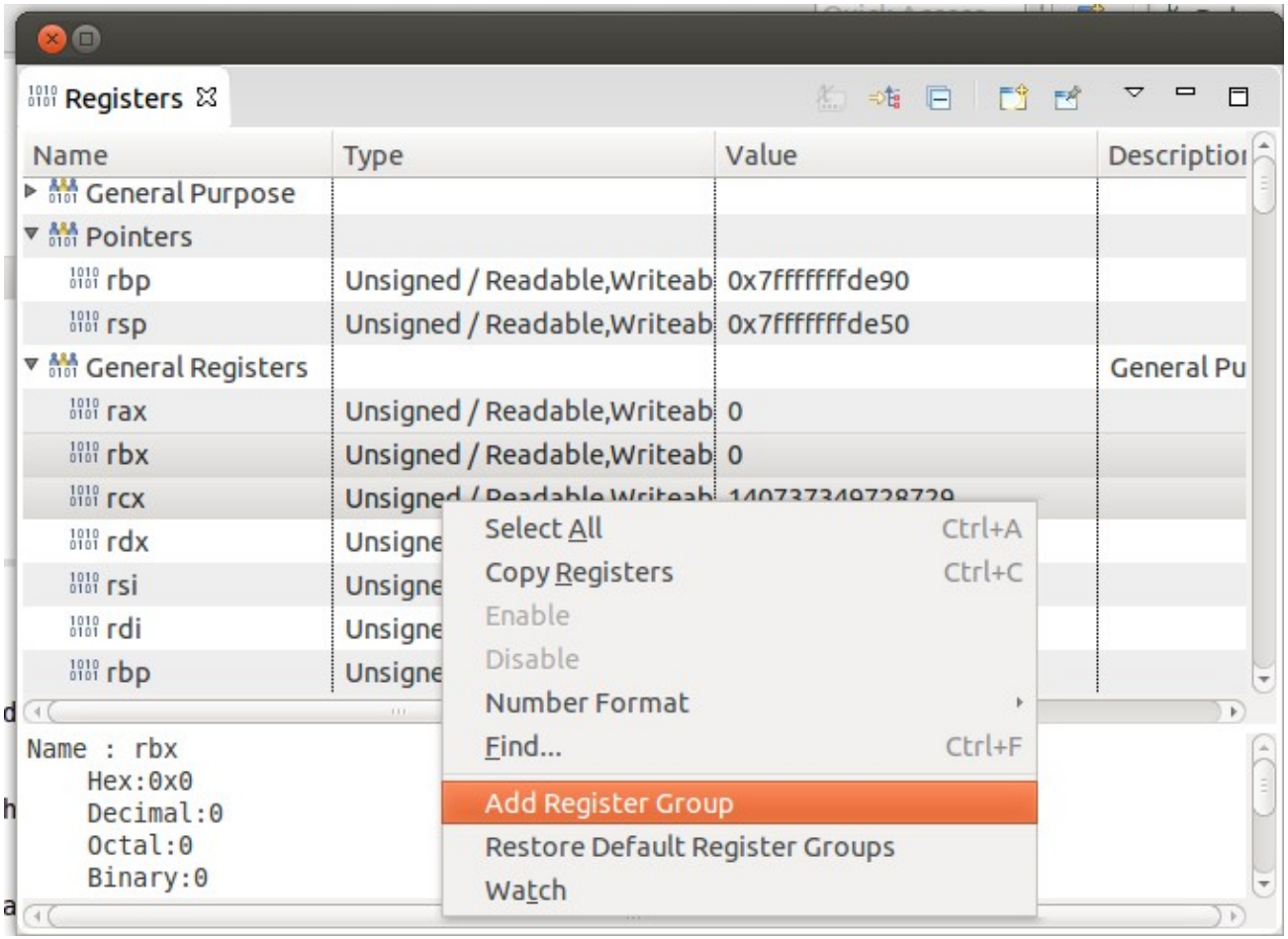
Expression	Type	Value	Address
*	Group-pattern	54 unique matches	
a	int [2]	0x7fffffffdbf0	0x7fffffffdbf0
aba	int	100100 (Binary)	0x7fffffffdcd4
argc	int	1	
argv	char **	0x7fffffffddf8	
array_large	int [111]	0x7fffffffd6e0	
array_small	int [4]	0x7fffffffdbd0	
b	int [5]	0x7fffffffdba0	
bbb	int *[5]	0x7fffffffdae0	
c	char	0 '\0'	
de	int	0 (Binary)	

```
Name : aba
    Details:36
    Default:36
    Decimal:36
    Hex:0x24
    Binary:100100
    Octal:044
```

- Select All Ctrl+A
- Copy Expressions Ctrl+C
- Remove
- Remove All
- Number Format
  - Details
  - Default
  - Decimal
  - Hex
  - • Binary
  - Octal
  - Restore To Preference
- Find... Ctrl+F
- Add Watch Expression...
- Disable
- Enable
- Edit Watch Expression...
- Add Watchpoint (C/C++)...
- Cast To Type...
- Display As Array...
- Watch

REGISTER GROUPS

› Ability to create groups of registers



PIN&CLONE FOR VISUALIZER

- Ability to pin a Multicore Visualizer to a session
- Allows to monitor multiple systems concurrently

MINI CORE DUMPS

- Effort of the Linux Diamon workgroup (diamon.org)

- Mini core dumps:
  - Configurable excerpt of full core dump
  - Space savings (good for embedded)
  - Storage of multiple mini core dumps

- Coming to a Linux distribution in the near future!

FUTURE PLANS

GLOBAL BREAKPOINTS

› Contribution to Linux Kernel ongoing

Applies to every process

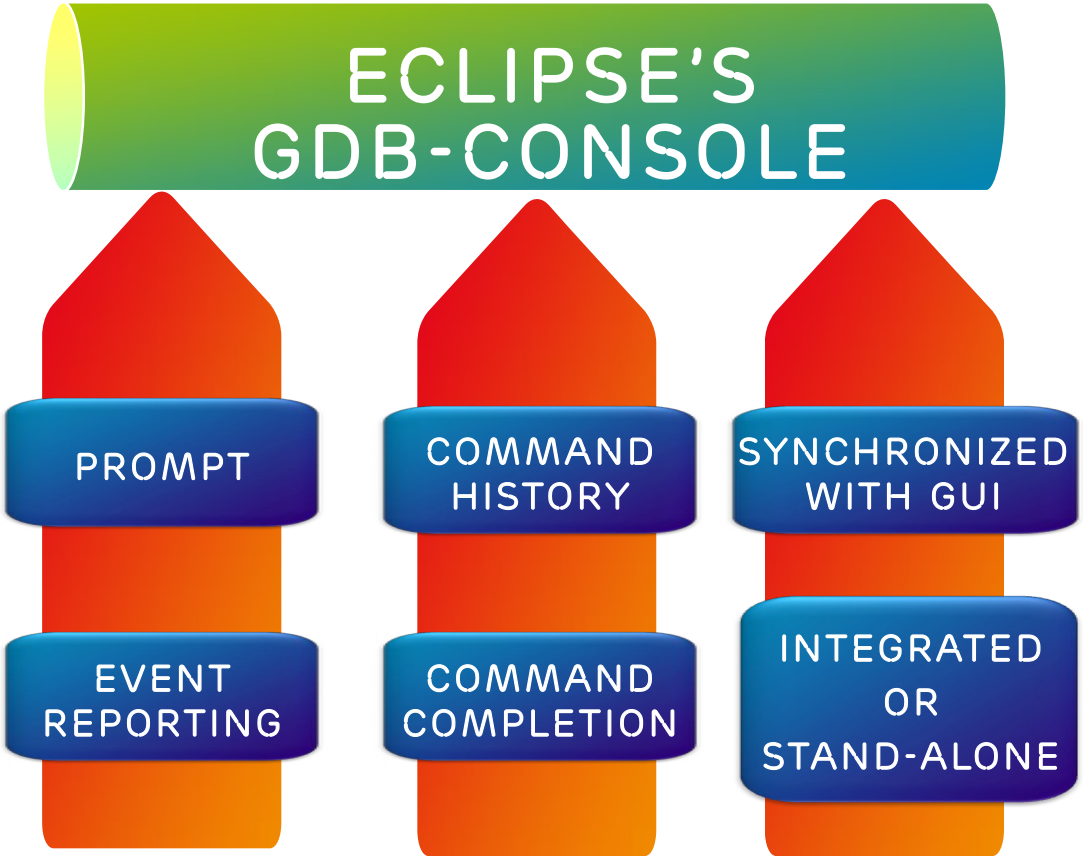
Auto attach when hit

Un-started or short lived process

```
(gdb) gbreak return.cc:14
Global Breakpoint 6 at 0x80484aa: file return.cc, line 14.
```

INTEGRATED GDB CONSOLE

› Coming in 2015!



PTC SETS

Process Thread Core (PTC) sets control groups of debug elements:

- Step threads numbered between 34 and 59

```
(gdb) step .34-59
```

- Step all threads running on core 2

```
(gdb) step @2
```

- Stop everything running on cores 5 to 7, preventing new threads from being started

```
(gdb) interrupt *.future@5-7
```