Trace Compass Update

December 2019

Matthew Khouzam BNEW SAN Eng Tools 2

What we will talk about

- About
- New Trace Compass Features
- Towards Theia Distributed IDE
- Scripting
- "Demo" walkthrough
- Internal Success story
- Looking forward

Who am I?

- Ericsson Open Source Developer
- Committer to Trace Compass
- Co-Lead of the Incubator
- Loud
- Enthusiast of most things technical (nerd)

What is Trace Compass

Eclipse Trace Compass is an open source application to solve performance and reliability issues by reading and analyzing traces and logs of a system. Its goal is to provide views, graphs, metrics, and more to help extract useful information from traces, in a way that is more user-friendly and informative than huge text dumps.

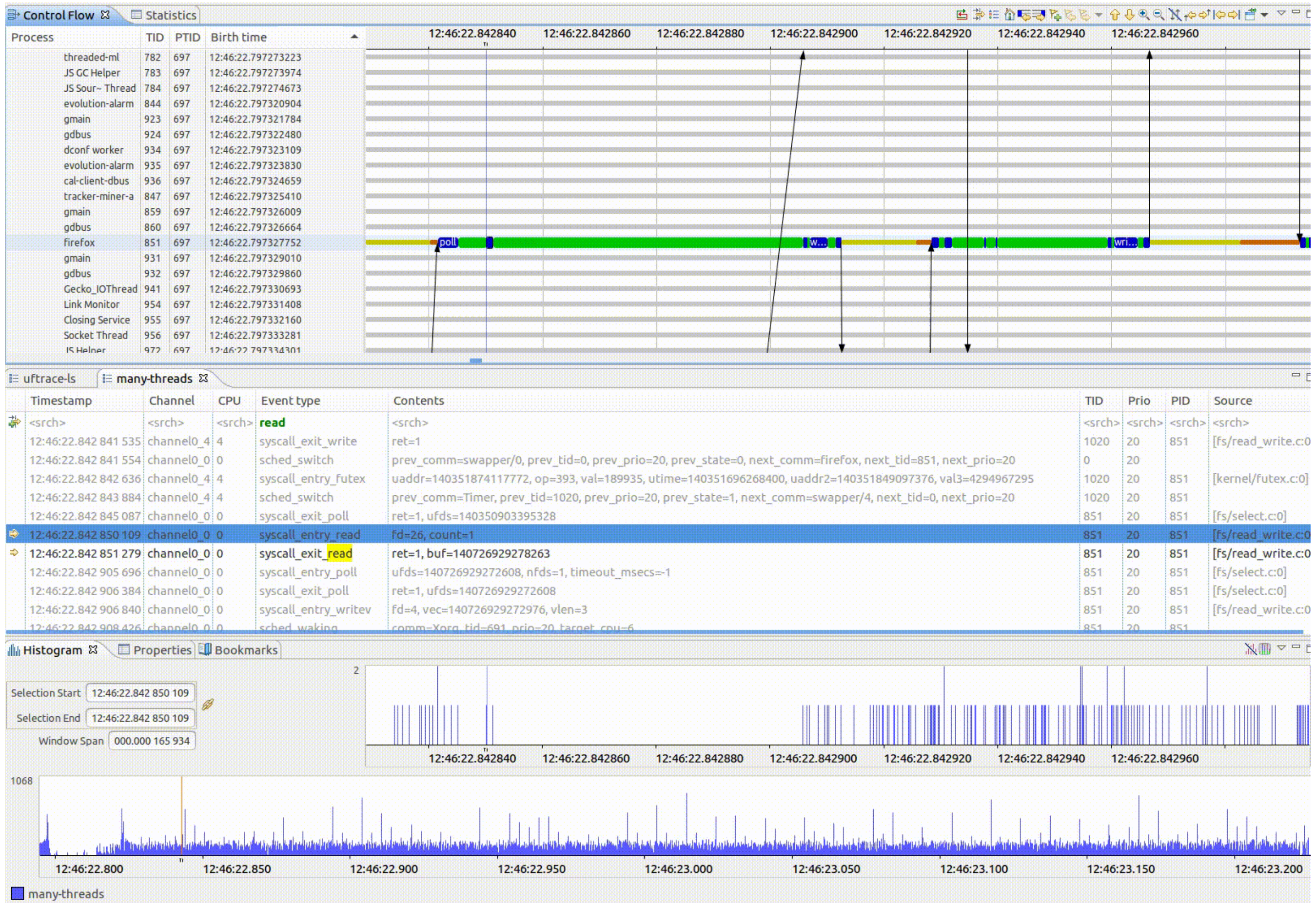
New in Trace Compass (1/2)

- Tutorials (https://github.com/tuxology/tracevizlab)
- Memory Leak Detector
- WASD navigation (XY)
- Getting ready for CTF 2.0 (not supported yet)
- Default way to get source code lookup
- Source code lookup on states/segments
- Support State/segment filtering on start/end/duration
- State system compression
- UX improvements
- Java 11 support

New in Trace Compass (2/2)

- Data Driven styles
- Aggregate tooltips
- Lines in Time Graphs
- Aspects categorized, (category/numerical)
- Launch analysis by double-clicking it.
- Add way to customize columns of extended tables
- Performance improvements on filtering

Source code lookup

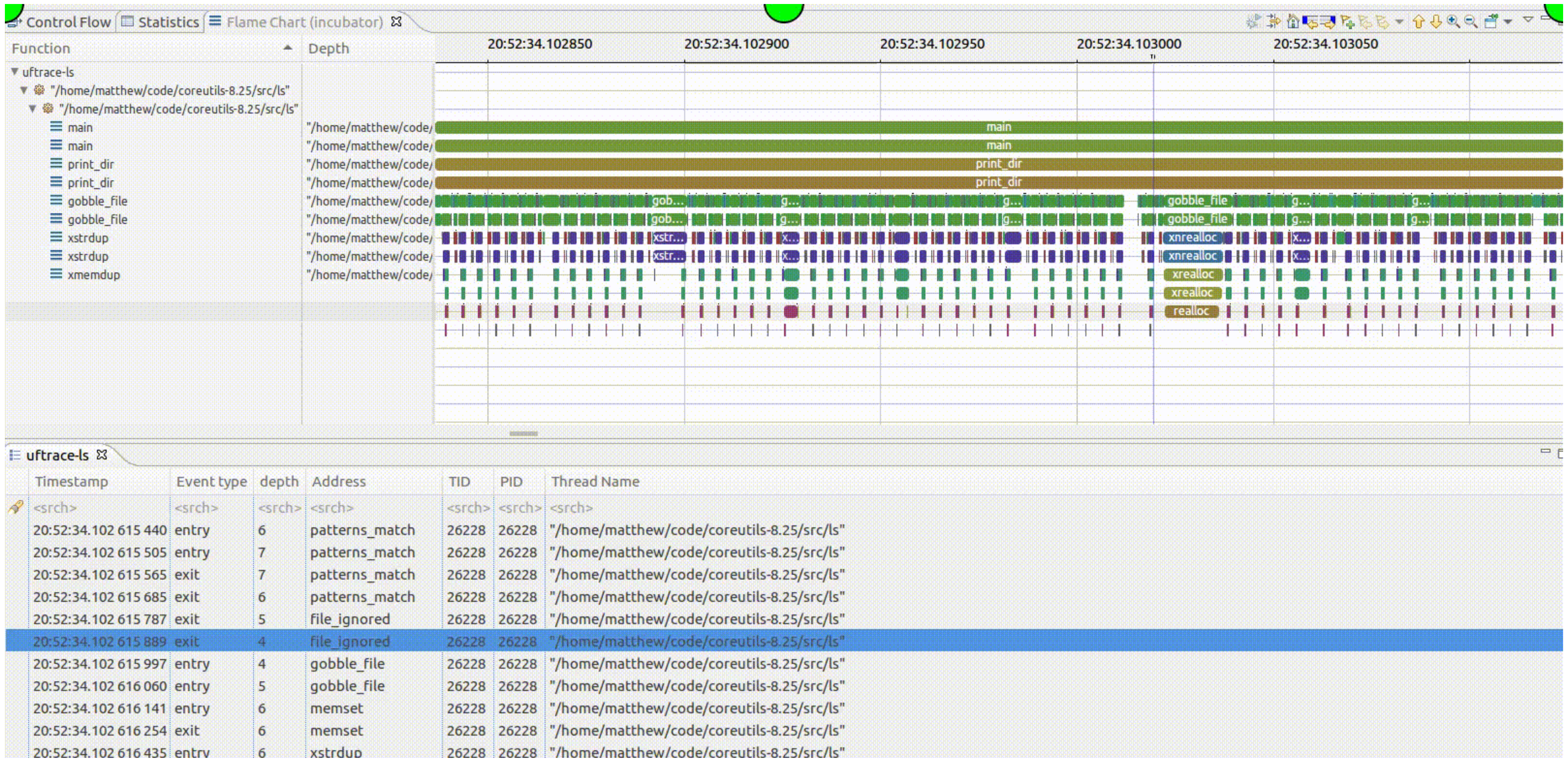


Process	TID	PTID	Birth time
threaded-ml	782	697	12:46:22.797273223
JS GC Helper	783	697	12:46:22.797273974
JS Sour~ Thread	784	697	12:46:22.797274673
evolution-alarm	844	697	12:46:22.797320904
gmain	923	697	12:46:22.797321784
gdbus	924	697	12:46:22.797322480
dconf worker	934	697	12:46:22.797323109
evolution-alarm	935	697	12:46:22.797323830
cal-client-dbus	936	697	12:46:22.797324659
tracker-miner-a	847	697	12:46:22.797325410
gmain	859	697	12:46:22.797326009
gdbus	860	697	12:46:22.797326664
firefox	851	697	12:46:22.797327752
gmain	931	697	12:46:22.797329010
gdbus	932	697	12:46:22.797329860
Gecko_IOThread	941	697	12:46:22.797330693
Link Monitor	954	697	12:46:22.797331408
Closing Service	955	697	12:46:22.797332160
Socket Thread	956	697	12:46:22.797333281
JS Helper	972	697	12:46:22.797334301

Timestamp	Channel	CPU	Event type	Contents	TID	Prio	PID	Source
<srch>	<srch>	<srch>	read	<srch>	<srch>	<srch>	<srch>	<srch>
12:46:22.842 841 535	channel0_4	4	syscall_exit_write	ret=1	1020	20	851	[fs/read_write.c:0
12:46:22.842 841 554	channel0_0	0	sched_switch	prev_comm=swapper/0, prev_tid=0, prev_prio=20, prev_state=0, next_comm=firefox, next_tid=851, next_prio=20	0	20		
12:46:22.842 842 636	channel0_4	4	syscall_entry_futex	uaddr=140351874117772, op=393, val=189935, utime=140351696268400, uaddr2=140351849097376, val3=4294967295	1020	20	851	[kernel/futex.c:0]
12:46:22.842 843 884	channel0_4	4	sched_switch	prev_comm=Timer, prev_tid=1020, prev_prio=20, prev_state=1, next_comm=swapper/4, next_tid=0, next_prio=20	1020	20	851	
12:46:22.842 845 087	channel0_0	0	syscall_exit_poll	ret=1, ufds=140350903395328	851	20	851	[fs/select.c:0]
12:46:22.842 850 109	channel0_0	0	syscall_entry_read	fd=26, count=1	851	20	851	[fs/read_write.c:0
12:46:22.842 851 279	channel0_0	0	syscall_exit_read	ret=1, buf=140726929278263	851	20	851	[fs/read_write.c:0
12:46:22.842 905 696	channel0_0	0	syscall_entry_poll	ufds=140726929272608, nfds=1, timeout_msecs=-1	851	20	851	[fs/select.c:0]
12:46:22.842 906 384	channel0_0	0	syscall_exit_poll	ret=1, ufds=140726929272608	851	20	851	[fs/select.c:0]
12:46:22.842 906 840	channel0_0	0	syscall_entry_writev	fd=4, vec=140726929272976, vlen=3	851	20	851	[fs/read_write.c:0
12:46:22.842 908 426	channel0_0	0	sched_waking	comm=Xorg, tid=691, prio=20, target_cpu=6	851	20	851	

Selection Start	12:46:22.842 850 109
Selection End	12:46:22.842 850 109
Window Span	000.000 165 934

Improved filtering



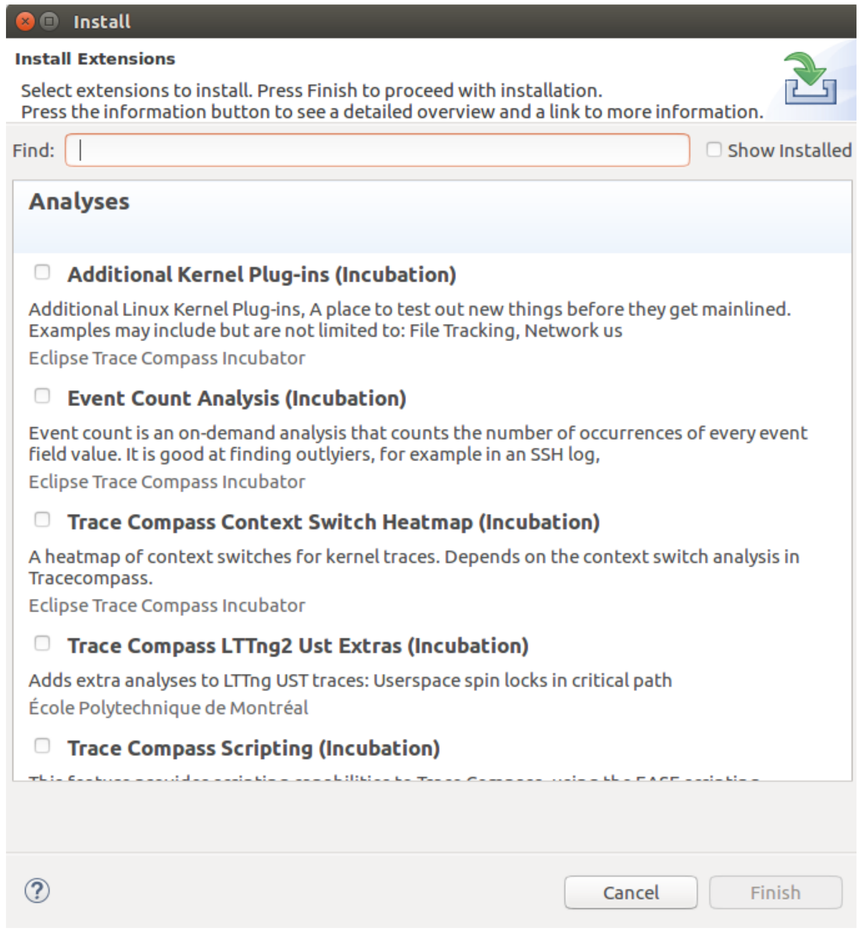
Function	Depth
uftrace-ls	
"/home/matthew/code/coreutils-8.25/src/ls"	
"/home/matthew/code/coreutils-8.25/src/ls"	
main	"/home/matthew/code/
main	"/home/matthew/code/
print_dir	"/home/matthew/code/
print_dir	"/home/matthew/code/
gobble_file	"/home/matthew/code/
gobble_file	"/home/matthew/code/
xstrdup	"/home/matthew/code/
xstrdup	"/home/matthew/code/
xmemdup	"/home/matthew/code/

uftrace-ls

Timestamp	Event type	depth	Address	TID	PID	Thread Name
<srch>	<srch>	<srch>	<srch>	<srch>	<srch>	<srch>
20:52:34.102 615 440	entry	6	patterns_match	26228	26228	"/home/matthew/code/coreutils-8.25/src/ls"
20:52:34.102 615 505	entry	7	patterns_match	26228	26228	"/home/matthew/code/coreutils-8.25/src/ls"
20:52:34.102 615 565	exit	7	patterns_match	26228	26228	"/home/matthew/code/coreutils-8.25/src/ls"
20:52:34.102 615 685	exit	6	patterns_match	26228	26228	"/home/matthew/code/coreutils-8.25/src/ls"
20:52:34.102 615 787	exit	5	file_ignored	26228	26228	"/home/matthew/code/coreutils-8.25/src/ls"
20:52:34.102 615 889	exit	4	file_ignored	26228	26228	"/home/matthew/code/coreutils-8.25/src/ls"
20:52:34.102 615 997	entry	4	gobble_file	26228	26228	"/home/matthew/code/coreutils-8.25/src/ls"
20:52:34.102 616 060	entry	5	gobble_file	26228	26228	"/home/matthew/code/coreutils-8.25/src/ls"
20:52:34.102 616 141	entry	6	memset	26228	26228	"/home/matthew/code/coreutils-8.25/src/ls"
20:52:34.102 616 254	exit	6	memset	26228	26228	"/home/matthew/code/coreutils-8.25/src/ls"
20:52:34.102 616 435	entry	6	xstrdup	26228	26228	"/home/matthew/code/coreutils-8.25/src/ls"

Incubator

- Friendly reminders:
  - Many interesting features are here
  - Integrated into Trace Compass (Tools->Add-Ons...)
  - Many other trace types available
- If something is useful, tell us!



Install Extensions

Select extensions to install. Press Finish to proceed with installation.
Press the information button to see a detailed overview and a link to more information.

Analyses

Additional Kernel Plug-ins (Incubation)
Additional Linux Kernel Plug-ins, A place to test out new things before they get mainlined. Examples may include but are not limited to: File Tracking, Network us
Eclipse Trace Compass Incubator

Event Count Analysis (Incubation)
Event count is an on-demand analysis that counts the number of occurrences of every event field value. It is good at finding outlyiers, for example in an SSH log,
Eclipse Trace Compass Incubator

Trace Compass Context Switch Heatmap (Incubation)
A heatmap of context switches for kernel traces. Depends on the context switch analysis in Tracecompass.
Eclipse Trace Compass Incubator

Trace Compass LTTng2 Ust Extras (Incubation)
Adds extra analyses to LTTng UST traces: Userspace spin locks in critical path
École Polytechnique de Montréal

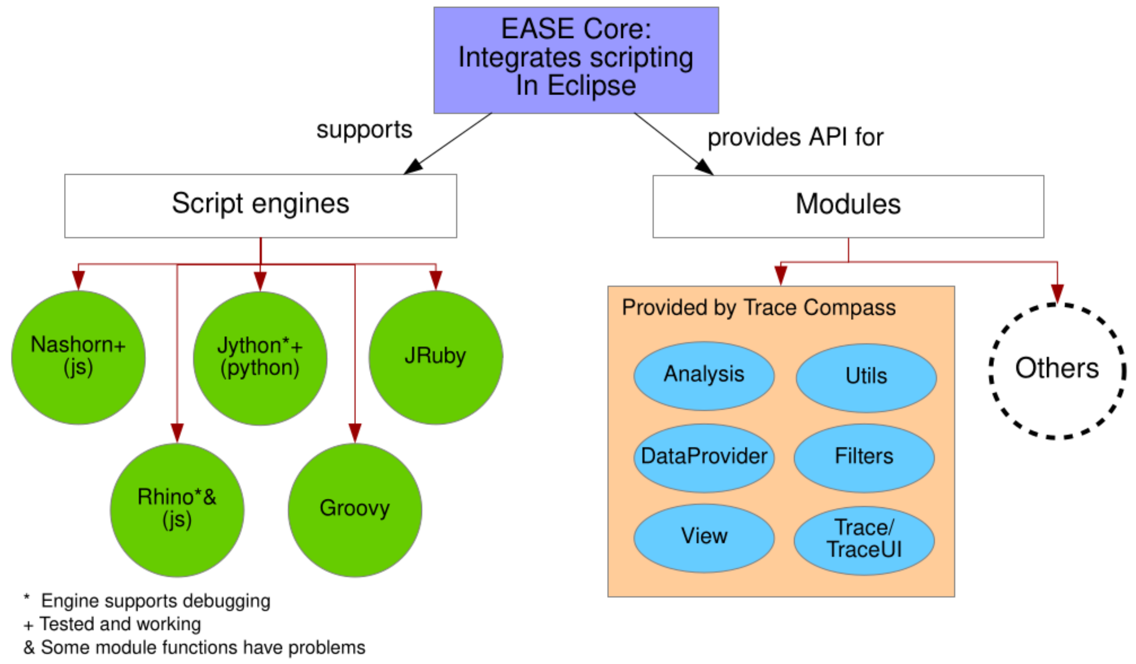
Trace Compass Scripting (Incubation)

Towards Theia

- Theia is a distributed IDE Framework
- More decoupling
- Porting views to data provider
- Providing style suggestions on the core side
- Improving Eclipse implementation at the same time
- Opportunities:
  - Command Line Mode (headless)
  - CI integration
  - Different views
  - Context sharing
  - Security/Performance/UX
- Caveat: With current resource level, it's not happening overnight

Let’s make extending Trace Compass EASEier

- Eclipse Advanced Script Environment
- There are several patterns we can infer easily
- BUT we cannot be domain experts on YOUR application.
- Give flexibility to the user
- Proposed solution: scripting.

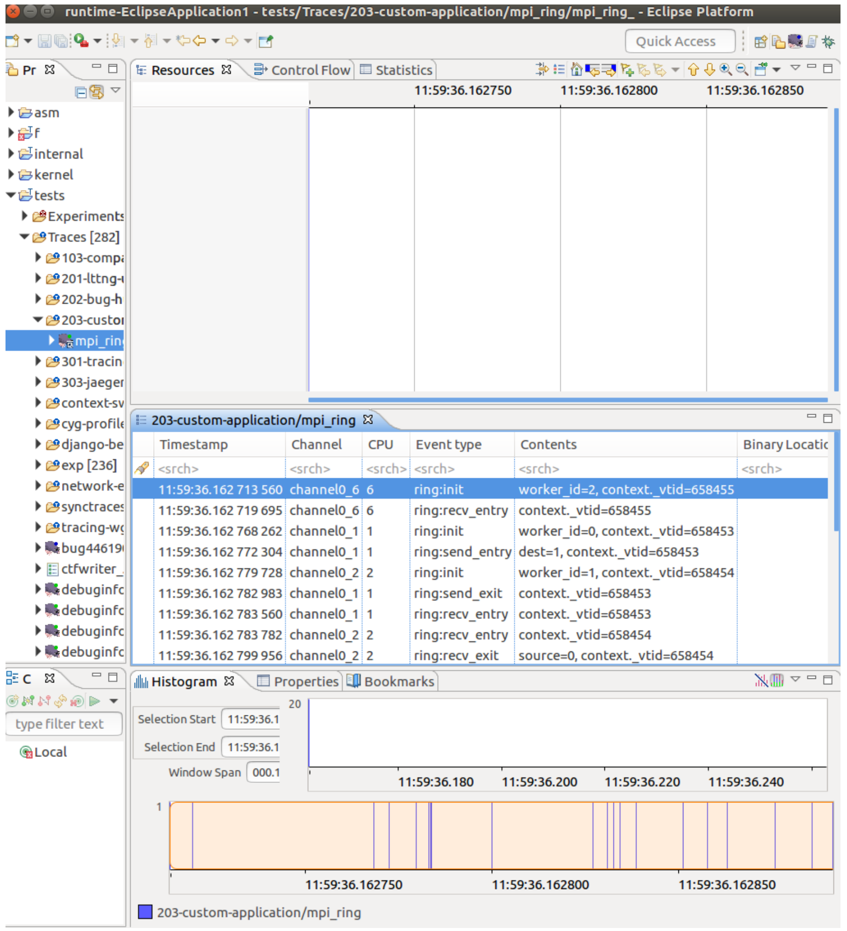


Histogram per event type

```
var map = new java.util.HashMap();
map.put(ENTRY_PATH, 'event_types/sched*');
map.put(ENTRY_DELTA, true);
// create a XY data provider
var provider = createXYProvider(analysis, map);

// Open an XY chart with this data provider
if (provider != null) {
          openXYChartView(provider);
}
```

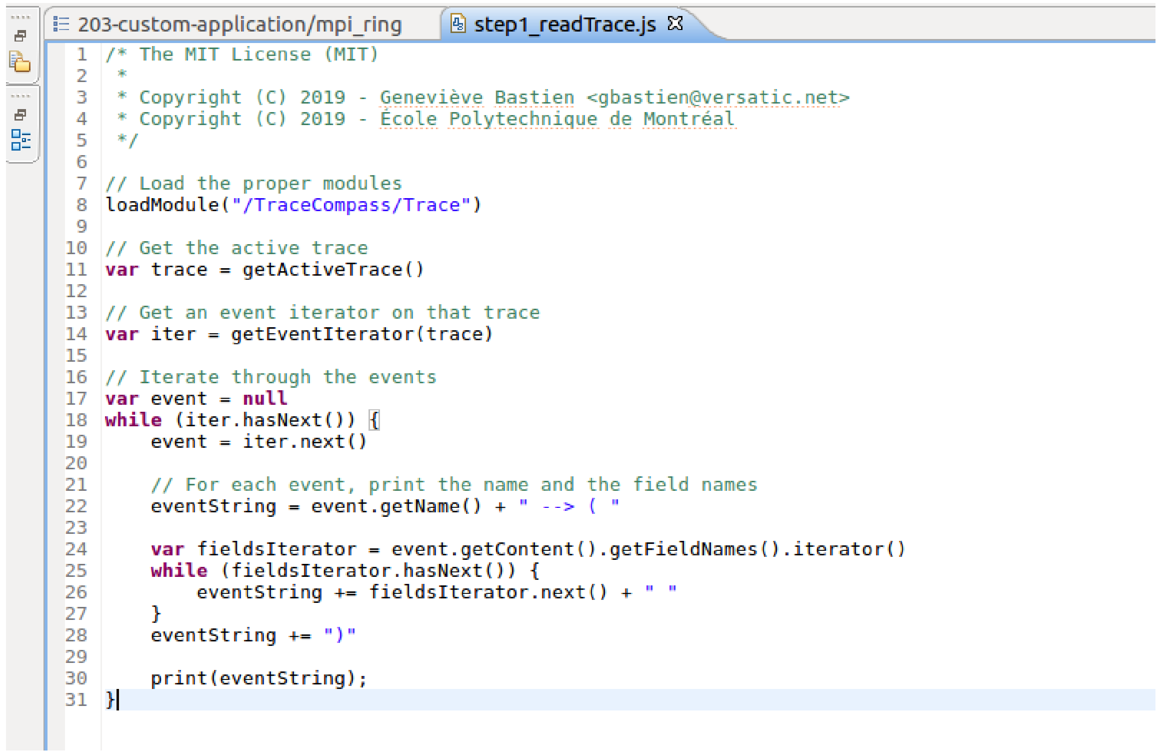
More elaborate script

- MPI Message tracking.

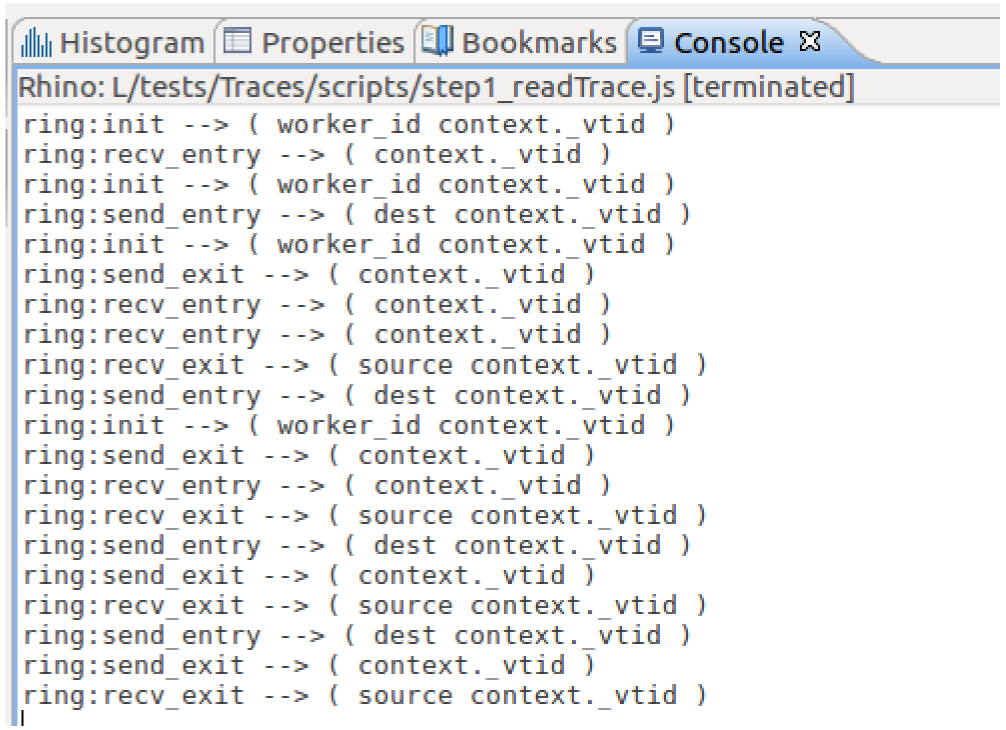

runtime-EclipseApplication1 - tests/Traces/203-custom-application/mpi_ring/mpi_ring_ - Eclipse Platform
Resources
Control Flow
Statistics
Histogram
Properties
Bookmarks
203-custom-application/mpi_ring

Timestamp	Channel	CPU	Event type	Contents	Binary Locatic
<srch>	<srch>	<srch>	<srch>	<srch>	<srch>
11:59:36.162 713 560	channel0_6	6	ring:init	worker_id=2, context._vtid=658455	
11:59:36.162 719 695	channel0_6	6	ring:recv_entry	context._vtid=658455	
11:59:36.162 768 262	channel0_1	1	ring:init	worker_id=0, context._vtid=658453	
11:59:36.162 772 304	channel0_1	1	ring:send_entry	dest=1, context._vtid=658453	
11:59:36.162 779 728	channel0_2	2	ring:init	worker_id=1, context._vtid=658454	
11:59:36.162 782 983	channel0_1	1	ring:send_exit	context._vtid=658453	
11:59:36.162 783 560	channel0_1	1	ring:recv_entry	context._vtid=658453	
11:59:36.162 783 782	channel0_2	2	ring:recv_entry	context._vtid=658454	
11:59:36.162 799 956	channel0_2	2	ring:recv_exit	source=0, context._vtid=658454	

Simple script



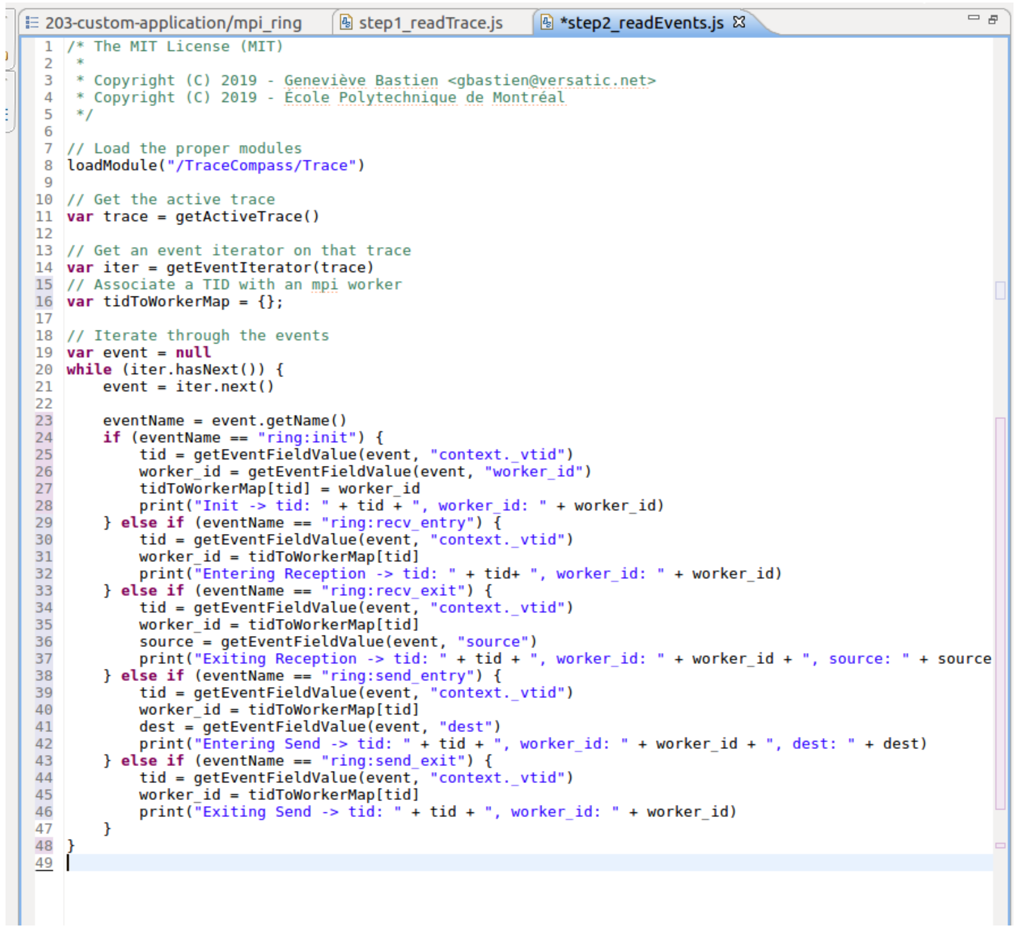
```javascript
/* The MIT License (MIT)
 *
 * Copyright (C) 2019 - Geneviève Bastien <gbastien@versatic.net>
 * Copyright (C) 2019 - École Polytechnique de Montréal
 */

// Load the proper modules
loadModule("/TraceCompass/Trace")

// Get the active trace
var trace = getActiveTrace()

// Get an event iterator on that trace
var iter = getEventIterator(trace)

// Iterate through the events
var event = null
while (iter.hasNext()) {
    event = iter.next()

    // For each event, print the name and the field names
    eventString = event.getName() + " --> ( "

    var fieldsIterator = event.getContent().getFieldNames().iterator()
    while (fieldsIterator.hasNext()) {
        eventString += fieldsIterator.next() + " "
    }
    eventString += ")"

    print(eventString);
}
```



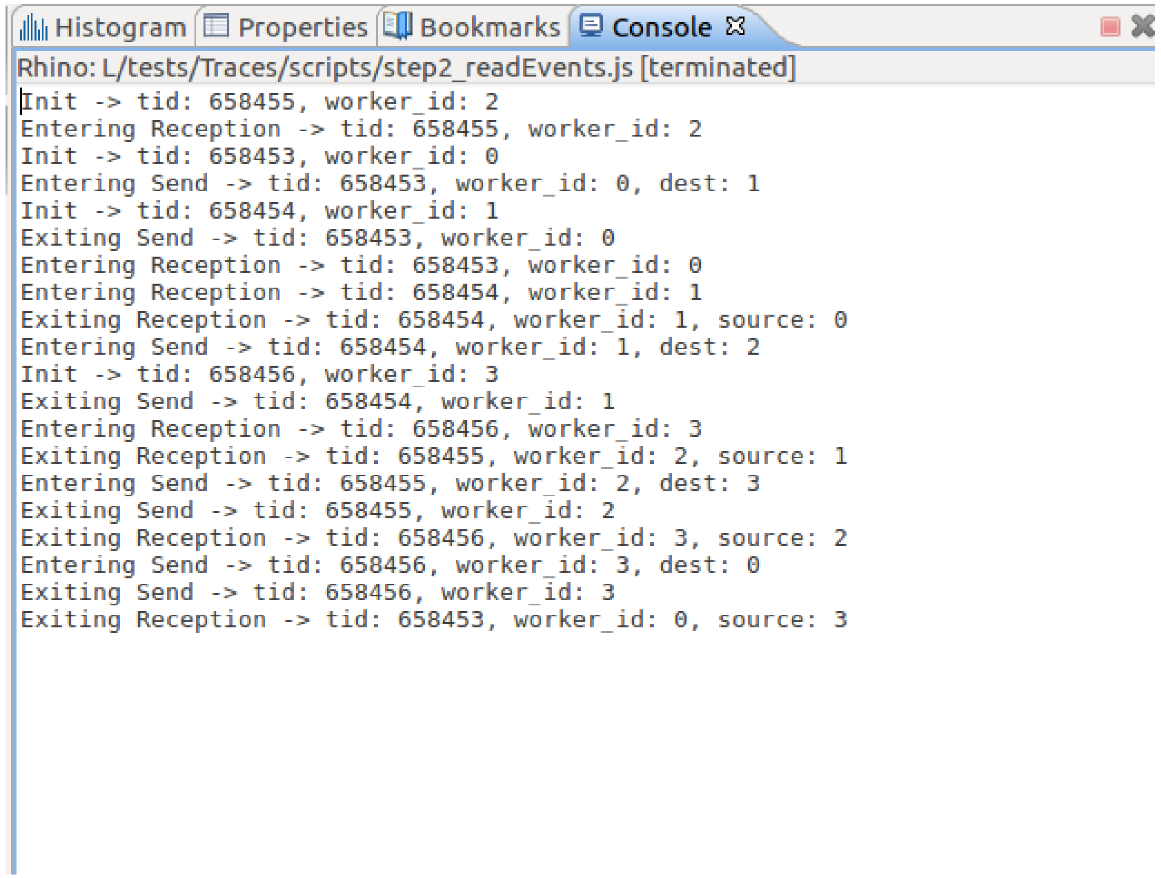
Rhino: L/tests/Traces/scripts/step1_readTrace.js [terminated]

```
ring:init --> ( worker_id context._vtid )
ring:recv_entry --> ( context._vtid )
ring:init --> ( worker_id context._vtid )
ring:send_entry --> ( dest context._vtid )
ring:init --> ( worker_id context._vtid )
ring:send_exit --> ( context._vtid )
ring:recv_entry --> ( context._vtid )
ring:recv_entry --> ( context._vtid )
ring:recv_exit --> ( source context._vtid )
ring:send_entry --> ( dest context._vtid )
ring:init --> ( worker_id context._vtid )
ring:send_exit --> ( context._vtid )
ring:recv_entry --> ( context._vtid )
ring:recv_exit --> ( source context._vtid )
ring:send_entry --> ( dest context._vtid )
ring:send_exit --> ( context._vtid )
ring:recv_exit --> ( source context._vtid )
ring:send_entry --> ( dest context._vtid )
ring:send_exit --> ( context._vtid )
ring:recv_exit --> ( source context._vtid )
```

Simple script – Read Every event



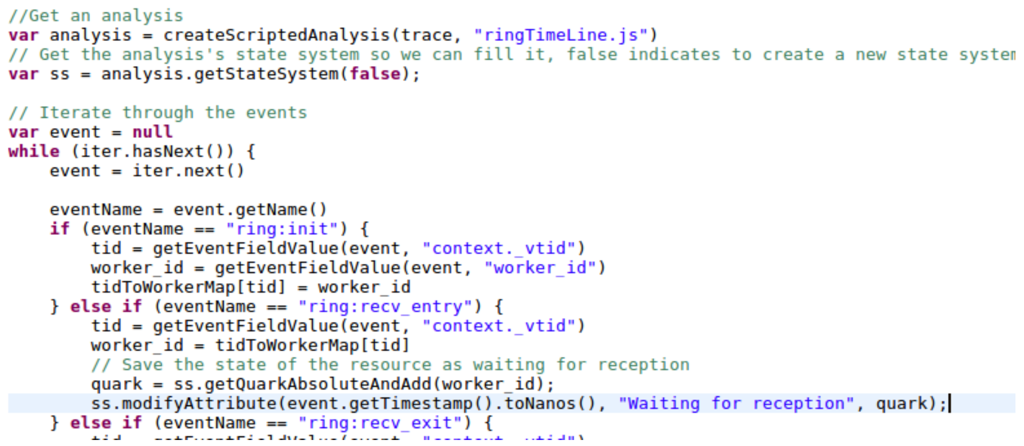
```javascript
/* The MIT License (MIT)
 *
 * Copyright (C) 2019 - Geneviève Bastien <gbastien@versatic.net>
 * Copyright (C) 2019 - École Polytechnique de Montréal
 */

// Load the proper modules
loadModule("/TraceCompass/Trace")

// Get the active trace
var trace = getActiveTrace()

// Get an event iterator on that trace
var iter = getEventIterator(trace)
// Associate a TID with an mpi worker
var tidToWorkerMap = {};

// Iterate through the events
var event = null
while (iter.hasNext()) {
    event = iter.next()

    eventName = event.getName()
    if (eventName == "ring:init") {
        tid = getEventFieldValue(event, "context._vtid")
        worker_id = getEventFieldValue(event, "worker_id")
        tidToWorkerMap[tid] = worker_id
        print("Init -> tid: " + tid + ", worker_id: " + worker_id)
    } else if (eventName == "ring:recv_entry") {
        tid = getEventFieldValue(event, "context._vtid")
        worker_id = tidToWorkerMap[tid]
        print("Entering Reception -> tid: " + tid+ ", worker_id: " + worker_id)
    } else if (eventName == "ring:recv_exit") {
        tid = getEventFieldValue(event, "context._vtid")
        worker_id = tidToWorkerMap[tid]
        source = getEventFieldValue(event, "source")
        print("Exiting Reception -> tid: " + tid + ", worker_id: " + worker_id + ", source: " + source
    } else if (eventName == "ring:send_entry") {
        tid = getEventFieldValue(event, "context._vtid")
        worker_id = tidToWorkerMap[tid]
        dest = getEventFieldValue(event, "dest")
        print("Entering Send -> tid: " + tid + ", worker_id: " + worker_id + ", dest: " + dest)
    } else if (eventName == "ring:send_exit") {
        tid = getEventFieldValue(event, "context._vtid")
        worker_id = tidToWorkerMap[tid]
        print("Exiting Send -> tid: " + tid + ", worker_id: " + worker_id)
    }
}
```



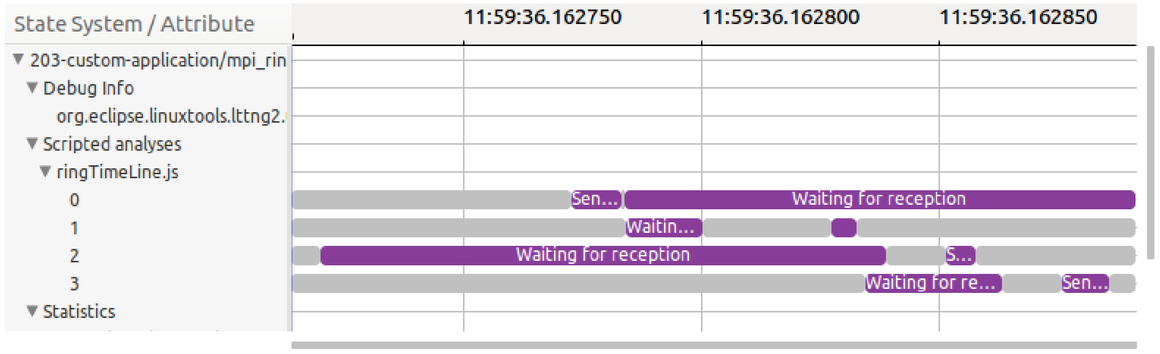
Rhino: L/tests/Traces/scripts/step2_readEvents.js [terminated]

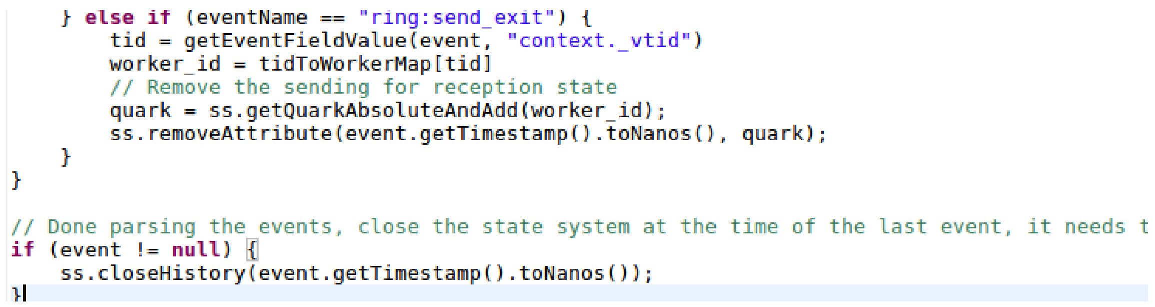
```
Init -> tid: 658455, worker_id: 2
Entering Reception -> tid: 658455, worker_id: 2
Init -> tid: 658453, worker_id: 0
Entering Send -> tid: 658453, worker_id: 0, dest: 1
Init -> tid: 658454, worker_id: 1
Exiting Send -> tid: 658453, worker_id: 0
Entering Reception -> tid: 658453, worker_id: 0
Entering Reception -> tid: 658454, worker_id: 1
Exiting Reception -> tid: 658454, worker_id: 1, source: 0
Entering Send -> tid: 658454, worker_id: 1, dest: 2
Init -> tid: 658456, worker_id: 3
Exiting Send -> tid: 658454, worker_id: 1
Entering Reception -> tid: 658456, worker_id: 3
Exiting Reception -> tid: 658455, worker_id: 2, source: 1
Entering Send -> tid: 658455, worker_id: 2, dest: 3
Exiting Send -> tid: 658455, worker_id: 2
Exiting Reception -> tid: 658456, worker_id: 3, source: 2
Entering Send -> tid: 658456, worker_id: 3, dest: 0
Exiting Send -> tid: 658456, worker_id: 3
Exiting Reception -> tid: 658453, worker_id: 0, source: 3
```

Script – State System



```
//Get an analysis
var analysis = createScriptedAnalysis(trace, "ringTimeLine.js")
// Get the analysis's state system so we can fill it, false indicates to create a new state systen
var ss = analysis.getStateSystem(false);

// Iterate through the events
var event = null
while (iter.hasNext()) {
    event = iter.next()

    eventName = event.getName()
    if (eventName == "ring:init") {
        tid = getEventFieldValue(event, "context._vtid")
        worker_id = getEventFieldValue(event, "worker_id")
        tidToWorkerMap[tid] = worker_id
    } else if (eventName == "ring:recv_entry") {
        tid = getEventFieldValue(event, "context._vtid")
        worker_id = tidToWorkerMap[tid]
        // Save the state of the resource as waiting for reception
        quark = ss.getQuarkAbsoluteAndAdd(worker_id);
        ss.modifyAttribute(event.getTimestamp().toNanos(), "Waiting for reception", quark);
    } else if (eventName == "ring:recv_exit") {
```



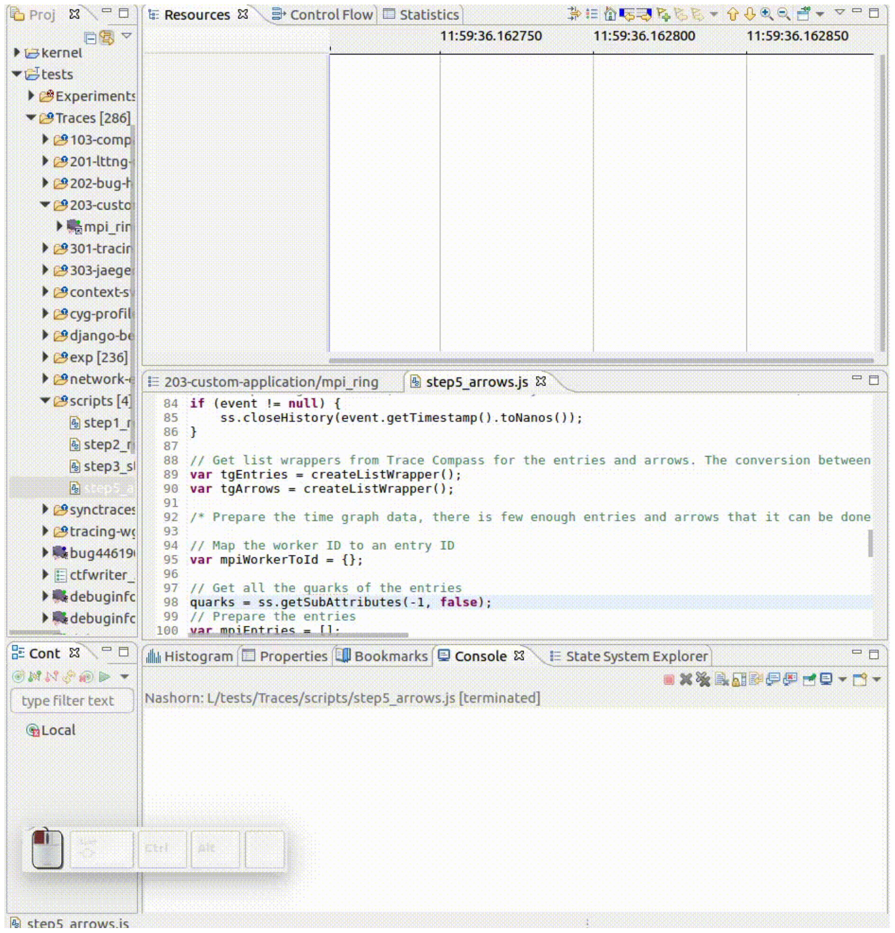


```
    } else if (eventName == "ring:send_exit") {
        tid = getEventFieldValue(event, "context._vtid")
        worker_id = tidToWorkerMap[tid]
        // Remove the sending for reception state
        quark = ss.getQuarkAbsoluteAndAdd(worker_id);
        ss.removeAttribute(event.getTimestamp().toNanos(), quark);
    }
}

// Done parsing the events, close the state system at the time of the last event, it needs t
if (event != null) {
    ss.closeHistory(event.getTimestamp().toNanos());
}
```

Script- Everything Together

— Script is 100 lines
— Link here

— GitHub examples:
  — https://github.com/tahini/tracecompass-ease-scripting



```
if (event != null) {
    ss.closeHistory(event.getTimestamp().toNanos());
}

// Get list wrappers from Trace Compass for the entries and arrows. The conversion between
var tgEntries = createListWrapper();
var tgArrows = createListWrapper();

/* Prepare the time graph data, there is few enough entries and arrows that it can be done

// Map the worker ID to an entry ID
var mpiWorkerToId = {};

// Get all the quarks of the entries
quarks = ss.getSubAttributes(-1, false);
// Prepare the entries
var mpiEntries = [];
```

```
Nashorn: L/tests/Traces/scripts/step5_arrows.js [terminated]
```

Trace Compass inside Ericsson

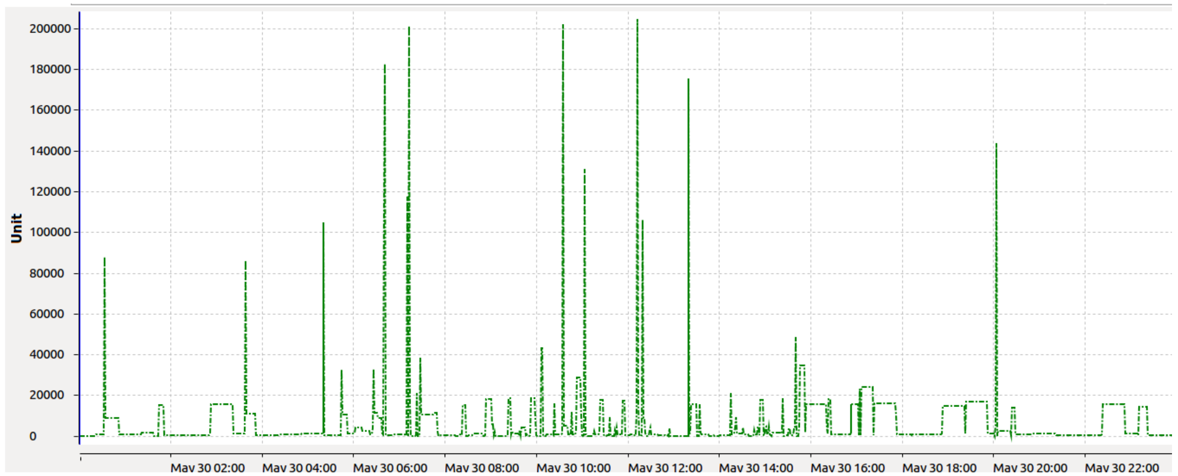
1. Ericsson has about 25000 R&D employees
2. Deal with performance issues on internal massively deployed services
3. We had an unexpected spike due to configuration issues (solved)
4. There were resource drains that were hidden in the noise (solved)

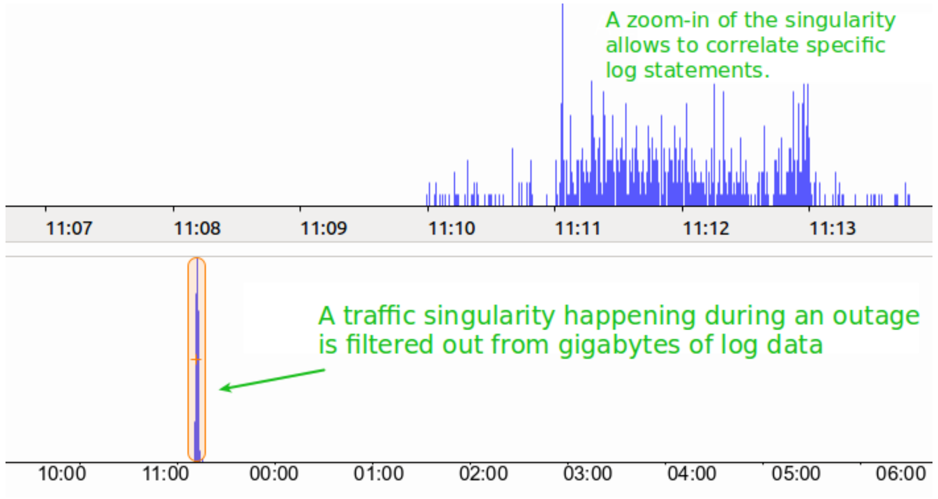
What did Trace Compass gain

- New (custom XML) trace formats supported
  - Catalina Logs An implementation of Apache Tomcat.
  - Hadoop Logs Logs from Apache Hadoop, a distributed file system.
  - HAProxy Logs High availability proxy and load balancer.
  - OpenSSH Logs Open Secure Shell log
  - OpenSSHd Logs Open Secure Shell server log
    - SSH Failed connections XML analysis A simple analysis to see the cause of connection failures vs the user.
- New use cases
  - Performance analysis with second granularity instead of nanosecond
- Improved documentation

What did they gain

- Problems fixed!
- Offline analysis of the traffic for every node by using custom parsers
- Ability to plot events over time, for selected log fields (ex: bandwidth usage over time, for a given IP address, see leftmost figure)
- Ability to corrolate information coming from different log file types (ex : correlate haproxy frontend logs with backend logs)





Looking forward

- Continue to progress towards the development of a scalable Client Server Architecture using Theia
- Continue to support internal Ericsson needs
- Community support, promote Trace Compass to external companies
  - Trace Server Protocol
  - EASE
- UX
- Performance

Takeaway

"Something that may be very easy for one person... might be very difficult for another." - Fred Rogers

- Thank you for contributing towards Trace Compass and the Tracing community
- I will probably be writing scripts in the hack-a-thon :)

Q & A

Contact

- Download
  - http://www.tracecompass.org
- Online
  - Jean Roussel Personna (TPM) jean-roussel.personna@ericsson.com
  - Bernd Hufmann (Lead) bernd.hufmann@ericsson.com
  - Matthew Khouzam (Presenter) matthew.khouzam@ericsson.com
  - Patrick Tasse patrick.tasse@ericsson.com
  - Simon Delisle simon.delisle@ericsson.com
  - IRC (oftc.net #tracecompass)
- Offline
  - Come by, we have cookies