Trace Compass UX Experiments - Feature #971

Trace digging tool

11/04/2015 12:00 AM - Julien Desfossez

Status:	New	**Start date:**	11/03/2015
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

When trying to understand something that happened in a trace, there is a workflow that for now is only possible using a text editor that can search.

The goal is to hide all the events that are unrelated to what we are looking for, and let the user figure out what happened. It should not be automated because it can be very complex and automations would probably make the process more painful, but I think TraceCompass could have an enhanced textdump view dedicated for this kind of investigation, and combined with the state system it could provide very powerful tools to help quickly dig into the trace.

Here are a couple of features that would be useful:
- hide (collapse maybe) any number of lines selected by the user
- provide a way to hide or highlight (with a simple switch between the 2 modes) all the events in a selection based on a filter (cpu_id, tid, event name with wildcard (sched_*), etc)
- highlight all the events that are on the same CPU as the current event
- multiple highlight patterns at the same time (ex: I want to watch all the sched_switch on CPU 3 and also see when PID X is woken up)
- resolve the FD number to a filename when we can
- show what is the current process scheduled on each CPU and its state
- jump to the previous/next sched_switch on the current CPU
- jump to the sched_switch related to a sched_wakeup/waking
- jump to the last wakeup/waking before the sched_switch
- if multiple wakeup/waking occurred before the sched_switch, allow to jump back and forth
- on the sched_waking event allow to jump to the entry of the softirq (if any) in which it occurred
- on the softirq_entry, allow to jump the related softirq_raise
- on the softirq_raise, allow to jump to the irq_handler_entry event
- link with the other views/analyses, ex: select a group of events and attach as a note the top 10 CPU users in this period, or I/O, etc
- let the user arbitrarily color a line or a group of lines
- attach notes to events or group of events that we can see from the listing
- keep bookmarks of events
- select 2 events in the listing and display the time delta
- always allow to hide/show the hidden events in case something hidden becomes useful after all
- the collapsed events could be one line in which we could display the number of events hidden, the time spent, etc
- export the final log in various formats (rich as well as text wrapped at 80 chars), it should be possible to send the report to someone (software engineers, suppor team, ML, etc) and show them what happened with eventually notes and colors to add context

Here is an example if such an investigation done with vim:
before: http://pastebin.ubuntu.com/13099101/
after: http://pastebin.ubuntu.com/13097737/

If you need example traces, just let me know.

For me, this kind of workflow is the most important part of trace processing, and I do it manually very frequently, so maybe I am dreaming with this list of features, but the day I have tool that speeds up this process, I'll switch !
And please, don't let me start coding a vim plugin to do that ;-)