Trace Compass: Finding Your Way Through All Sorts of Traces

DOMINIQUE TOUPIN, ERICSSON
About me

› Tool Manager at Ericsson, helping Ericsson sites to develop better software efficiently

› Telecommunication systems
  – Open, standards-based common platform
  – High availability, 99.999 %
  – Broad range of support for both infrastructure and value-added applications
  – Multimedia, network and application processing capabilities

I do not sell processors or tools 😊
DIFFICULT-TO-FIND BUGS

- Single core to multicore makes it harder:
  - Race conditions, Deadlocks, Non-deterministic behavior

- Many layers makes it worst:
  - Middleware, VM, OS, hypervisor, sometimes across nodes or clusters

- Debuggers are invaluable for algorithm issues but they fall short for the above category of problems, not to mention that some problems are not reproducible in the developer’s environment!
Heterogeneous Trace Correlation

› Linux on RISC high-level scheduling info: instrumented to collect kernel-level and user space level trace data
› Function call-level detail: HW trace probe collects low-level instruction and data trace on the same RISC processor
› Bare metal DSP events: SW trace instrumentation
› Network “accelerator” hardware block events: HW trace probe

Correlation of traces from different context with IPC between the cores
Tracing usefulness

- Tracing instead of logging to minimize impact on system behaviour
- Performance tuning
- Monitoring of live systems

Extreme cases and simple one: Linux Kernel, ust database
A TRACE USE CASE

› Log Levels
Assigned to static tracepoints, with a verbosity level: 0 less verbose to 14, most verbose

› Wildcards
Enable all events under certain hierarchy level: * for all events, libc_* for all events within libc, etc.

The combination of wildcards and loglevels allow users to gradually enable more specific instrumentation, and increase the verbosity level, as they narrow-down the source of their problem.

› Filtering
Filtering on specific event fields allow use-cases such as following a call-id
Started as **Tracing and Monitoring Framework (TMF)**

Initially for LTTng traces

Used to be under Eclipse Linux Tools Project

New project **Trace Compass** at Eclipse Foundation

- To **increase community and collaboration** in open-source
- Keep the momentum going!
› Framework to build trace visualization and analysis tools

› Scalability allows to handle traces exceeding memory

› Enables trace analysis and correlation from different sources

› Extensible for any trace or log format
  – Binary, text, XML etc.
Built-in trace parsers

- **Common Trace Format (CTF)**
  - Multi-Core Association Tool Infrastructure Workgroup
  - Used by LTTng Tracer for Linux Kernel and UST
  - Reviewed by Linux kernel developers

- **Packet Capture (libPcap)**
  - Used for IP network tracing

- **Best Trace Format (BTF)**

Available as **standalone application** or set of plug-ins
WHAT TRACE COMPASS PROVIDES
(FOR PLUG-IN DEVELOPERS)

› A trace and event data model
› Event request handling with coalescing
› Extension point to add new trace types
› Reusable views and widgets
  – E.g Time Graphs (Gantt-charts), X-Y-plots
› Integration into common navigator framework
  (e.g. project explorer)
› An event filter model
› Signal broadcasting between components
› Time window and event synchronization
› Generic state system
› Sequence diagram framework
COMMON FEATURES

› Management of traces, trace formats and experiments
  – Trace file handling
  – Experiment handling
  – Importing/Exporting
  – Automatic Type Detection
COMMON FEATURES

› Events Table

- Opens and closes with trace or experiment
- Columns are configurable per trace type
- All views synchronized to the currently selected trace
- Export table to text (columns tab-separated, with filter)
COMMON FEATURES

› Searching
- by text or regex on any field(s)
- highlighting of matching events
- navigation to next/previous match

› Filtering
- by text or regex on any field(s)
- incremental view of filter results
- filtered event count

› Highlighting
- user customizable color settings
- settings are applied by priority
- persistent and import/exportable
COMMON FEATURES

› Complex Filter
  – Filter model allowing multiple nested conditions of different type
  – Persistent and import/exportable
  – Can be applied on any trace

![Complex Filter Example](image-url)
COMMON FEATURES

› Bookmarks

– Store bookmarks to events of interest on a given trace
– Attach detailed text to a bookmark
– Quickly open trace and navigate to bookmarked event
COMMON FEATURES

› Package export and import

› For sharing of
  – Traces
  – Computed analyses results
  – Bookmarks

› Field engineer do some work and then send it to the design team for further analysis.
State System Support

- State system abstracts events, analyses traces and creates models to be displayed
- Persistent on disk, does not need to be rebuilt between runs
- Allows fast (O(log n)) queries of state attributes by time or type
- Support for several state systems in parallel

**Events:**
- sched_switch(process)
- irq_entry
- irq_exit
- sched_switch(swapper)

**States:**
- WAIT
- USERMODE
- INTERRUPTED
- USERMODE
- WAIT
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
    › Gantt-chart or X-Y-Graphs
  – Can be created without changing source code or recompiling

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

› Custom Text Parsers
  - line based parser with regex
  - allows user to define own parser with extracted data and output fields
  - parser definition created and edited with a wizard
  - parser definitions can be shared by importing / exporting to file

› Custom XML Parsers
  - XML based extracting data from XML elements and their attributes
COMMON FEATURES

› Histogram
  - Shows event density vs. time
  - Can optionally color events per trace, show lost events
  - Can be used to navigate the trace
  - Select time range and window span with mouse or text
COMMON FEATURES

› Sequence Diagrams
  - Works with any trace type
  - Extensions can specify how to translate events to sequence diagram transaction
  - Reference view provided that will work with supplied model
LT TNG INTEGRATION

› Reference implementations for
  – a plug-in extension to Trace Compass
  – various trace analyses
  – several visualization views

› Analysis of LT Tng Kernel and UST Traces
  – Linux Tracing Toolkit next generation
  – Traces in Common Trace Format (CTF)

› Support of Live Trace Reading

› LT Tng Remote Tracer Control
CONTROL FLOW VIEW

› Displays processes state changes (color-coded) over time
› State 'tooltips'
› Can display arrows to follow CPU execution
› Zooming and filtering
RESOURCES VIEW

› Displays system resource states (color-coded) over time
› State 'tooltips'
› Zooming and filtering
› Quick navigation between resources, states
CPU USAGE VIEW

- Display of total CPU usage and selected process
- Derived from Linux Kernel sched_switch events
CALL STACK VIEW

› Shows the stack trace at any point during execution
› Can be generated automatically from LTTng UST traces
› Name mapping can be fetched from binary debug symbols
› View can be re-used for other data sources
SYSTEM-WIDE TRACING

› Correlation of events within a cluster across
  – Nodes,
  – CPUs, ASICs, GPUs,
  – Hypervisor, Kernel, User-space, Virtual Machines, Libraries, 3PP, Applications

› Time synchronization of traces from multiple sources

› Based on synchronization information sources
  – Message exchanges, Time-stamps

› Complex timestamp transformation algorithm part of Trace Compass

› Complete reference implementation for LTTng Kernel traces
### PCAP NETWORK TRACE ANALYSIS

#### Custom Table

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004-05-13 06:17:08:222 534 000</td>
<td>feff:20:00:01:00:00/65.208.228.223/80</td>
<td>145.254.160.237/32</td>
<td>TCP</td>
<td>mostlyTCP.pcap</td>
</tr>
<tr>
<td>2004-05-13 06:17:08:222 534 000</td>
<td>feff:20:00:01:00:00/65.208.228.223/80</td>
<td>145.254.160.237/32</td>
<td>TCP</td>
<td>mostlyTCP.pcap</td>
</tr>
<tr>
<td>2004-05-13 06:17:08:222 534 000</td>
<td>feff:20:00:01:00:00/65.208.228.223/80</td>
<td>145.254.160.237/32</td>
<td>TCP</td>
<td>mostlyTCP.pcap</td>
</tr>
<tr>
<td>2004-05-13 06:17:08:222 534 000</td>
<td>feff:20:00:01:00:00/65.208.228.223/80</td>
<td>145.254.160.237/32</td>
<td>TCP</td>
<td>mostlyTCP.pcap</td>
</tr>
</tbody>
</table>

#### Packet Details:
Ethernet, IPv4, TCP, UDP

#### Stream List view

<table>
<thead>
<tr>
<th>ID</th>
<th>Endpoint A</th>
<th>Endpoint B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00:00:01:00:00/145.254.160.237/3372</td>
<td>Feff:20:00:01:00/65.208.228.223/80</td>
</tr>
<tr>
<td>1</td>
<td>00:00:01:00:00/145.254.160.237/3371</td>
<td>Feff:20:00:01:00/216.239.59.99/80</td>
</tr>
</tbody>
</table>
Event Traces & Analysis

Task States and Transitions

- Extended Definition of Task State Machine
  - Compatible with OSEK Task State Machine
  - Includes states for Multi-Core tracing and analysis
  - Additional Polling State (active waiting)
  - Released as BTF 2.1
BTF INTEGRATION

BTF Parser

Gantt-Chart based on Generic Static System

Custom Table

Trace and Event Details
GDB TRACEPOINT ANALYSIS

› Plug-in extension to Trace Compass

› Dynamic Tracepoints with GDB

› Integrated with Eclipse CDT Debug
  – For creating of GDB tracepoints
  – For collection of tracepoint information
  – Synchronization of Trace Compass with CDT Debug

› Visualization of GDB Trace Log using Trace Compass
GDB TRACEPOINT ANALYSIS
UPCOMING FEATURES

▶ More LTTng live trace reading support (monitoring)
▶ Common time axis across all time-based views
▶ User-customizable columns in table
▶ Enhanced filtering on state system values
▶ Sequence diagrams from network traces (libPcap)
▶ Correlation of core dumps with application traces
▶ Critical path analysis
▶ Analysis of traces from virtualized environments
People

Project Leads:
Bernd Hufmann
Alexandre Montplaisir

Comitters:
Geneviève Bastien
Bernd Hufmann
Matthew Khouzam
Marc-André Laperle
Alexandre Montplaisir
Patrick Tassé

Mentors:
Marc Khouzam
Jonas Helming

Interested Parties:
- Dominique Toupin, Ericsson
- Mathieu Desnoyers, EfficiOS
- Michel Dagenais, École Polytechnique de Montréal
- Xavier Raynaud, Kalray
- François Tétreault, Ciena
- Aaron Spear, VMware
- Anna Dushistova
- Generoso Pagano, Inria
- Harald Mackamul, Robert Bosch GmbH
- Daniel Kunz, Robert Bosch GmbH
- Andreas Olofsson, Adapteva
- Sébastien Gérard, CEA
- Chrstelle Burguera, STMicroelectronics
- Jerome Correnoz, STMicroelectronics
- Philippe Maisonneuve, Wind River
- Jonas Svennebring, Avago Technologies
- Ed Martinez, Freescale
- Lukas Mewes, Behr-Hella Thermocontrol
- Al Grant, ARM
FEATURE WISHLIST

› Tracing in heterogeneous cloud environments at many different levels
› Perf support
› Support for system-wide performance data
› Epiphany specific support
  – Native support for coprocessors
  – Monitor chip network traffic using Epiphany network monitor
› Practical support for thousands of traces
› Papyrus integration
  – Revert a trace into a UML sequence diagram using Papyrus
› Live Mode
› Stack trace on selected events
REFERENCES

› Project pages
  - Trace Compass Proposal
  - Eclipse Trace Compass Project
  - Is Trace Compass Fast Yet
  - Eclipse Linux Tools Project - LTTng Integration

› User and Developer Guides
  - Trace Compass User Guide
  - PCAP - Network Analysis User Guide
  - GDB Tracepoint Analysis User Guide
  - Trace Compass Developer Guide
REFERENCES

› Working Groups

- Multicore Association Tool Infrastructure Working Groups
- PolarSys
- Diagnositic and Monitoring Working Group

› Others

- CTF Requirements, specification, reference implementation
- Best Trace Format (BTF)
- libpcap (PAcket CAPture)
CONTACTS

› Product Manager
  – Dominique.Toupin@ericsson.com

› Trace Compass Committers
  – Bernd.Hufmann@ericsson.com
  – Alexandre.Montplaisir-gon.alves@ericsson.com
  – Marc-Andre.Laperle@ericsson.com
  – Matthew.Khouzam@ericsson.com
  – Patrick.Tasse@ericsson.com
  – Genevieve.Bastien@polymtl.ca

› Trace Compass Mailing List
QUESTIONS
IS TRACE COMPASS FAST YET?

Performance Tracking Infrastructure

Summary
Application component:
- CTF Read & Seek Benchmark (500 seeks)
- CTF Read Benchmark
- Event matching
- Experiment benchmark
- LTtng kernel analysis
- Pack Read & Seek Benchmark (1000000 seeks)
- Pack Read Benchmark
- Trace synchronization

Filter:
- O/S
- JVM
- All
- Linux / OpenJDK 7

Performance changes compared to baseline
If no indication: LOWER is better

Date
05/05/14 - 09/25/14