



Powerful Software Analysis

Software tracing is an efficient way to record information about a program's execution. Programmers, experienced system administrators, and technical support personnel use it to diagnose problems, tune for performance, and improve comprehension of system behavior. Trace Compass facilitates the visualization and analysis of traces and logs from multiple sources; it enables diagnostic and monitoring operations from a simple device to an entire cloud.

System-Wide Tracing is here

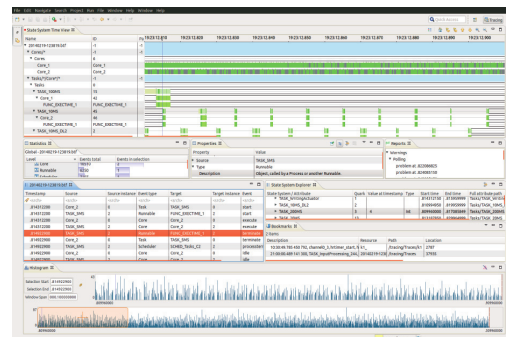
Trace Compass can take multiple traces and logs from different sources and formats, and merge them into a single event stream that allows system-wide tracing. It is possible to correlate application, virtual machine, operating systems, bare metal, and hardware traces and to display the results together, delivering unprecedented insight into your entire system.

Overview

- Can be integrated into Eclipse IDE or used as a standalone application. Eclipse plug-ins facilitates the addition of new analysis and views.
- Provides an extensible framework written in Java that exposes a generic interface for integration of logs or trace data input.
- Custom text or XML parsers can be added directly from the graphical interface by the user.
- Extendable to support various proprietary log or trace files.
- Designed to be scalable: can handle traces that exceed available memory.
- Remote tracer configuration and control with built-in support for LTTng traces.
- Traces taken from different clocks can be correlated through event relationships.
- Configurable, data-driven analyses and views.

Benefits

- Faster resolution of complex problems
- Clear understanding of system behaviour
- System performance optimization
- Adds value in many stages of a product's life cycle: development, test, production
- Facilitates the addition of new parsers, analyses, and views to solve specific issues
- Scales to GBs of compact binary trace
- Correlate traces from different nodes in a cloud with trace synchronization
- Multi-platform: Linux, Windows, MacOSX.



Built-In Analyses and Views

- Memory and Processor Usage
- Control Flow
- Hardware and Software Resource Allocation
- Histogram
- Call Stack
- Network Stream
- Detailed Events List
- Trace Statistics
- Flame Graph
- Critical Path



Real-time Analysis

In real-time systems, late data is bad data. Trace Compass allows developers to identify misdeadlines, providing them latency analyses of custom execution phases. Trace Compass provides informative metrics such as period distribution, density and statistics, that help spot outliers.



Critical Flow Analysis

Blocked threads could lead to misbehavior or failures. Trace Compass provides a critical flow view that shows dependency chains for a given process. This allows to identify higher priority process blocked by a lower one.



Trace Annotation

Trace Compass provides bookmarks that allows users to annotate their traces and easily navigate to region of interests.

Built-in Trace Format support

- Common Trace Format (CTF)
- Linux LTTng kernel traces (CTF)
- Linux LTTng-UST userspace traces (CTF) which can come from proprietary software or open source eg. Mir, MariaDB, QEMU traces
- Linux Perf traces (CTF)
- Bare metal traces (CTF)
- Hardware traces (e.g. IEEE Nexus 5001 CTF conversion).
- GDB traces for debugging
- Best Trace Format for OSEK
- Libpcap (Packet CAPture) format, for network traces
- Custom text or XML parsers that can be added directly from the graphical interface by the user
- Can be extended to support various proprietary log or trace files.

Resources

- Trace Compass: <http://www.tracecompass.org>
- Trace Compass project: <http://projects.eclipse.org/projects/tools.tracecompass>
- Linux Tracing Toolkit: <http://lttng.org>
- Diagnostics and Monitoring Workgroup: <http://www.diamon.org>
- Babletrace: <http://diamon.org/babletrace>
- Common Trace Format: <http://diamon.org/ctf>

