The Tracing and Monitoring Framework in 2013
What will we see today

› Review
  - Why a trace viewer?
  - What is Eclipse/TMF

› Feature Recap
  - Developer features
  - User features

› New Developments
  - New features

› Trace Extensions

› Upcoming features
Why a Tracer or Trace Viewer

› **Production Machine Troubleshooting**: Tracing does not affect the program flow and can be used in the field.

› **Performance**: know where your performance is lost, don't guess it.

› **System understanding**: see the flow of a program to understand its functioning.
What is Eclipse? TMF?

› Eclipse is an IDE
› TMF is the Tracing and Monitoring Framework
› Eclipse-LTTng is the trace viewer for LTTng 2.x traces
› Part of the Eclipse Linux Tools Project
› Mentor, Freescale, MontaVista, Intel, Google and more are using TMF
Features for developers

› A trace and event data model
› Extension point to add new trace types
› Reusable views and widgets
› Integration into common navigator framework of Eclipse (e.g. project explorer)
› An event filter model
› Time window and event synchronization
› Generic state system
› Ability to hook own analysis tools

› Common Trace Format (CTF) parser v1.8.x
› Custom text & XML parser wizards (no code required!)
› Documentation
› Tested code!
TMF/LTTng features

- Detailed events
- Filters
- Search
- Highlighting
- Bookmarking
- Histogram (event density)
- Time Chart View
- Statistics

- Sequence Diagram
- Trace aggregation by experiments (sets of traces)
- Integration into Project Explorer
- Environment Variables (CTF)
- Tracer Control (LTTng)
- Control Flow (LTTng Kernel)
- Resource (LTTng Kernel)
Recent features

› Navigate to source model and call-site from event
› Support for multiple state systems per trace
› State system now drives statistics
› Trace indexing progress / speed shown
› Selected event details in Properties view
› Process filtering in the control flow view (kernel)
› Support for LTTng Tools 2.1 (Tracer Control), 2.2 is waiting for 2.2
› Verbose trace error messages
› Call stack display vs time
Event source navigation

› Select CTF event in trace editor, and if available, the context menu will allow the user to:
  – Open the source code in C editor at line where trace event was created
  – Open in EMF editor the model element where trace event was created
Statistics

› Shows number of events by type per trace
› Shows number of events by type in selected time range
› Customizable statistics per trace type
› Fast computation thanks to state system

![Statistics Table]

- **Level**
- **Events total**
- **Events in selected time range**

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Events total</th>
<th>Events in selected time range</th>
</tr>
</thead>
<tbody>
<tr>
<td>block_bio_backmerge</td>
<td>236</td>
<td>0</td>
</tr>
<tr>
<td>block_bio_queue</td>
<td>650</td>
<td>44</td>
</tr>
<tr>
<td>block_bio_remap</td>
<td>650</td>
<td>44</td>
</tr>
<tr>
<td>block_getrq</td>
<td>414</td>
<td>44</td>
</tr>
<tr>
<td>block_plug</td>
<td>326</td>
<td>44</td>
</tr>
<tr>
<td>block_rq_complete</td>
<td>414</td>
<td>44</td>
</tr>
<tr>
<td>block_rq_insert</td>
<td>414</td>
<td>44</td>
</tr>
<tr>
<td>block_rq_issue</td>
<td>414</td>
<td>44</td>
</tr>
<tr>
<td>block_unplug</td>
<td>326</td>
<td>44</td>
</tr>
<tr>
<td>exit_syscall</td>
<td>231170</td>
<td>1447</td>
</tr>
<tr>
<td>irq_handler_entry</td>
<td>2372</td>
<td>55</td>
</tr>
<tr>
<td>irq_handler_exit</td>
<td>2372</td>
<td>55</td>
</tr>
</tbody>
</table>
Tracer Control

› Control the LTTng tracer on local or remote systems
› Works using RSE, SSH/SFTP (industry standard)
› Configure tracer, control trace session and import trace into projects
› Supports Kernel and UST
› Users do not need to go to the command line
› Configure event filtering at tracer level
› Network streaming
CTF 1.8.2 support

› Support for call-sites added
  - When the trace point is written, the code location is stored. It can be re-opened, allowing navigation between an event and the corresponding source code.

› Support for models added
  - A model site can be attached to a trace event, this feature allows back navigation from the trace to the model.
Detailed error messages

› Relays more information to users
› Helpful for developing your traces or seeing why your friend's trace didn't work.
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
- Supports multiple backends:
  - Full
  - In memory
  - Partial
- Each trace type can define its own state system (example)
  - TmfTrace (base class) defines a state system for statistics
  - LttngKernelTrace (specific) defines a state system for kernel traces
Trace Extensions

GDB Tracepoint Analysis
- Integrated with CDT Debug which supports creating of GDB Tracepoints and collection of tracepoint information
- Visualization of GDB Trace Log in TMF
- Synchronization of TMF with CDT Debug
- Part of open source
Upcoming features

› Trace synchronization of traces from multiple hosts
› Data driven state systems (Thanks to collaborators!)
› Batch import trace wizard (So close!)
› Exporting of bookmarks
› CTF Writer
› Live traces (reading and viewing while tracing is ongoing)
› New analysis views (ie: generic charts, latency, CPU usage, network usage, data x-y plots…)

[Graph showing CPU usage over time]
Demo
REFERENCES

› Download at http://www.eclipse.org
  - Git: http://git.eclipse.org/c/linuxtools/org.eclipse.linuxtools.git
  - Development Environment Setup
    http://wiki.eclipse.org/Linux_Tools_Project/LTTng_Eclipse_Plug-in_Development_Environment_Setup
  - More instructions: http://lttng.org/eclipse

› LTTng: http://lttng.org

Contact:
{matthew.khouzam|alexandre.montplaisir-gon.alves|patrick.tasse|marc-andre.laperle|bernd.hufmann} <at> ericsson.com

Or
Join us on mailing lists

Or
Chat