Tracing and Monitoring Framework

Linux Symposium / Tracing Summit
Montreal, July 14, 2009

François Chouinard
Ericsson Canada
Summary

- Background
- Project Scope
- Framework Overview
- Framework Structure
- Exemplary Tool Integration (LTTng)
- Screenshots
- Demo
Background

- Open-Source IDE initiative
  - Full-fledged, C/C++ development environment
  - State-of-the-art tool suite
  - Open-source
  - Eclipse integration

- Tracing and Monitoring component
  - Facilitate the integration of tracing tools
  - Provide out-of-the-box “common” functionalities
  - Hosted by Eclipse Technology / Linux Tools
Project Scope

- Extendable support for:
  - Tools discovery
  - Tools control
  - Trace/data retrieval and storage
  - Trace/data visualization
  - Analysis/correlation/comparison/… modules integration

- Additional features:
  - Local and remote tools
  - Live and concurrent trace streams
  - Asynchronous events
  - Traces/logs that exceed available memory
  - External, host-based, libraries and analysis tools
  - Custom trace/log parsers
Tool Discovery

- Purpose
  - Identify the available trace providers and their capabilities
  - This information is used to generically control the tools

- Features
  - Discovery of available log providers
  - Discovery of log provider capabilities
  - Integration scheme for existing monitoring tools
  - Support for local and remote tools
Tool Control

Purpose
- Control the tool operation
- Manage the resources allocated to tracing

Features
- Basic tool control (conf/start/stop/pause/resume/…)
- Generic trace triggering, filtering
- Tracing rate regulation (throttling)
  - To avoid congestion on the target, host, transport link, …
- Budget policy (per trace, trace type, …)
  - To constrain target resource usage (CPU, memory, bandwidth)
- Control settings persistence
Data Retrieval and Storage

- **Purpose**
  - Collect and store tracing/monitoring data
  - Generic trace/log data interface (for the analysis tools)

- **Features**
  - Collect monitoring data from the tool
    - File transfer
    - Continuous stream
    - Multiple, heterogeneous streams
  - Provide a generic log file interface
    - Support for log-specific parsers
    - Support for sequential, random access, checkpoints, DB, …
    - Support for large files (bigger than available memory)
Data Visualization

- **Purpose**
  - Provide a set of standard data visualization tools
  - Toolbox of widgets (trace agnostic)

- **Features**
  - Provide generic monitoring views
    - Event logs (raw, tabular)
    - Time Line, Sequence Diagram, Logic Analyser, Gantt Chart
    - CPU/Memory/Heap/Network usage
    - Search filters, pattern matching, saved search queries, …
  - Provide generic graphical widgets
    - Charts, Histograms, …
  - Extendable for application-specific contents
Analysis Tools Integration

• Purpose
  ➢ Provide basic analysis functions
  ➢ Support host-based, external analysis tools and libraries

• Features
  ➢ Log comparison (regression testing, health monitoring, performance analysis,…)
  ➢ Causal dependency analysis
    ✤ Event Dependency Tree
    ✤ Critical Path
    ✤ Correlation of event data
    ✤ Reconstruction of event sequences from related traces
    ✤ Execution replay
  ➢ External tools integration
    ✤ Scheme to access the tracing data generically
    ✤ Scheme to send the analysis results to UI views/widgets
Framework Structure

- OS (Linux, Windows, OS/X, ...)
- Eclipse platform
- TMF plugin
- TM FW UI plugin
- Tool plugin (e.g. LTTng)
- Tool UI plugin
Framework Structure

- Key Concept: The Event
  - Basic Event
    - Timestamp
    - Source
    - Type (format)
    - Content (fields)
  - Extended for application-specific events
  - Possibility to handle derived/synthetic events

- Benefit:
  - Allows the handling of events using the framework generic components and APIs
Framework Structure

- Clock Adj.
- Checkpoints
- Event Cache
- Abstract Stream
- Abstract Parser
- Trace
- Event
- Trace Set
- Concrete Event
- Concrete Stream
- Concrete Parser
Framework Structure

Concrete View

View

Analysis Data

Concrete Analysis

Analysis Tool

Event

Concrete Event

Trace Set
Framework Structure
**LTTng Integration**

- **LTTng Perspective**
  - Project View
  - Control View
  - Time Frame View
  - Statistics View
  - Events Table View
  - Control Flow View
  - Resources View
  - Histogram View

- **LTTng Control (remote and local)**
  - Probe configuration
  - Start, stop, pause, resume
  - Trace retrieval

- **Framework Models**
  - Event Model
  - Event Log (Trace) Model
  - Request Model
  - View Model
  - Control Model

- **Framework Components**
  - Generic Events Table View
  - Widgets Toolbox
  - Support for very large trace files
  - Support for non-java parsers
  - Support for analysis components
External Library Integration

- Abstract Stream
- Abstract Parser
- Stream
- Parser
- JNI
- LTTV Library (C)
Screenshots: LTTV
Screenshots: LTTng Perspective
Demo!
References

- Linux Tools Project
  http://www.eclipse.org/linuxtools

- LTTng Integration
  http://www.eclipse.org/linuxtools/projectPages/lttng

- Linux Tracing Toolkit (LTTng)
  http://lttng.org
Contacts

- François Chouinard (TM Framework, LTTng)
  francois.chouinard@ericsson.com

- Dominique Toupin (Open-Source IDE)
  dominique.toupin@ericsson.com
Questions?

ERICSSON

TAking YOU Forward