Linux C/C++ Extended IDE

January 27, 2009

Dominique Toupin, François Chouinard

Introduction

- A Linux developer needs to use many IDE features to be productive: editing/compiling, debugging, profiling, memory analysis, code coverage, tracing/monitoring analysis, unit test, static analysis, emulator integration, etc.
- For C/C++ only editing/compiling/debugging is addressed in the Galileo CDT project. The other features are missing.
- In an effort to maximize the development effort of Eclipse committers a meeting was held with project leads from various Eclipse project to define a central place in Eclipse for Linux developer tools: Andrew Overholt (Linux Tools), Christian Kurzke (TmL), Doug Schaefer (CDT), Doug Gaff (DSDP), Eugene Chan (TPTP), Joe Green (MontaVista).
- The following slides described where/how to have a central place in Eclipse for Linux C/C++ Extended IDE.

Summary

- The Eclipse Linux Tools project will now be the central place to find/develop the Eclipse Linux extended IDE features for both host and remote target development.
- Other Eclipse Linux project will be able to re-use/develop general Linux tools from/in the Linux Tools project. Specialized feature will be part of specialized project e.g. mobile specific will be part of TmL.
- The Linux Tools Project will re-use generic features of other projects like CDT for compile/debug, TPTP for tracing/monitoring framework, DSDP/TM for remote connection, BIRT for reporting, Wascana for download/install of the whole tool chain i.e. both Eclipse and GNU tools.
- For TPTP, modifications will be made to enable LTTng, SystemTAP and other non-java feature integration.

Background

- Some of the missing features are currently being added into different projects

- Linux Distro Project (http://www.eclipse.org/linuxtools/)
  - OProfile http://www.eclipse.org/linuxtools/projectPages/oprofile
  - Valgrind http://www.eclipse.org/linuxtools/projectPages/valgrind
  - SystemTap
  - *Limited to Linux/C/C++*

- Wascana (http://wascana.sourceforge.net/)
  - Distributing Eclipse plug-ins with all the non-Eclipse tools and libraries using a p2 repository
  - Includes the GNU toolchain (e.g. GDB, GCC, etc)
  - *Limited to Windows*

Background

- C/C++ Development Tools (http://www.eclipse.org/cdt/)
  - Edit/Compile
  - Debug
  - Profiling, memory analysis was discussed on the mailing list
  - *Used for different target operating system*
  - *Has been extended to support additional languages, GCC/GDB also supports additional languages*
- Device Debugging (http://www.eclipse.org/dsdp/dd/)
  - Debugger Services Framework (DSF) with GDB reference implementation
  - GDB tracepoint being added
  - DSF was migrated into CDT
  - *Used for different target operating systems*
- Eclipse Mobile Industry Working Group (http://www.eclipse.org/org/industry-workgroups/mobilewg.php)
  - Need for a common IDE where most of the features are not mobile specific
  - Packaging/distribution of Eclipse and non-Eclipse tools to speed up end user adoption

Background

- Tools for Mobile Linux (http://www.eclipse.org/dsdp/tml/)
  - /proc file system on Linux
  - On the roadmap is http://www.eclipse.org/dsdp/tml/presentations/TmL_LinuxWorld_2008.pdf
    - Performance profiling
    - Memory analysis
    - Event logging
    - Live and post-mortem analysis
  - Some features are applicable to Linux in general

- DSDP / Target Management
  - Connection to a remote Linux system:
    - Files
    - Processes, e.g. enable/disable tracing on selected process
    - Shell
    - Etc.

- Parallel Tools Platform (http://www.eclipse.org/ptp/)
  - Memory analysis and profiling http://wiki.eclipse.org/PTP/designs/perf
  - Performance tool framework http://wiki.eclipse.org/Performance_Tools_Framework_Design
  - With multi-cores parallel performance analysis will be applicable to most new computer

Background

- Test & Performance Tools Platform (www.eclipse.org/tptp)
  - TPTP provides nice functionality for tracing/monitoring: Generic Log Adapter, Common Base Event, Symptoms Catalog, Managed Agent Explorer, etc.
  - However, TPTP is java-centric and has to be modified to work well with non-java code.
- At a tracing summit held in 2008 at Ericsson numerous companies have expressed the need to have a framework for tracing/monitoring
  - http://ltt.polymtl.ca/tracingwiki/index.php/TracingSummit2008
  - http://ltt.polymtl.ca/tracingwiki/index.php/List_of_Participants
- Numerous companies have commercial Eclipse plug-ins for those features and each has to maintain them although they are no longer differentiating features

Missing Features split in different areas

- Source Code Features
  - Profiler
  - Memory Analyzer
  - Code Coverage
  - Unit Test
  - Static Analysis
  - Tracepoint
  - More focussed on the internal of a component i.e. link to source code

- Distribution of the Linux reference implementation
  - Download/install one package to have the Linux reference implementation working

- Tracing/Monitoring
  - Tester / System Admin oriented tools
  - General issue of system ”observability”
  - Events collection, ordering, filtering, correlation
  - Log high-level information
  - Resources monitoring
  - Performance monitoring
  - Verification
  - <u>**A framework will most likely be used to implement different logs/traces features**</u>

- Emulator integration?
  - QEMU reference implementation?

Source Code Features

Source Code Features Profiling

- Integrate OProfile http://oprofile.sourceforge.net/about/
- http://www.eclipse.org/linuxtools/projectPages/oprofile

- Integrate Gprof, http://sourceware.org/binutils/docs-2.19/gprof/index.html, will have very similar features to OProfile. Gprof will not be integrated initially.

Source Code Features
Memory Analyzer

- Integrate Valgrind into Eclipse

  http://valgrind.org/info/about.html, initially memcheck and massif.

- http://www.eclipse.org/linuxtools/projectPages/valgrind

- Integrate mpatrol into Eclipse
  http://sourceforge.net/projects/mpatrol/

Source Code Features
Tracepoint

- Integrate GDB tracepoint into Eclipse
  http://sourceware.org/gdb/current/onlinedocs/gdb_12.html#SEC92
- Part of the implementation (e.g. MI calls) will end up in the CDT project
- The visualisation part will end up in the tracing/monitoring framework
- In addition to existing tracepoint functionality, the following features are currently being added and will also be part of the Eclipse integration
  - tracepoint dynamic conditions
  - trace state variables
  - target specific timestamp state
  - default tracepoint expressions
  - both fast and regular tracepoint definition
  - disconnected trace gathering
  - tracepoint expressions to allow comparison
  - shortcircuited boolean operators and assignment
  - on disk format for tracepoint data
  - write and read of tracepoint data
  - Tracepoint management.

Source Code Features Unit Test / Code Coverage

- Unit Test
  - Integrate CPPUnit into Eclipse
  - Eclipse integration already exists in source forge
    http://apps.sourceforge.net/mediawiki/cppunit/index.php?title=CppU

- Code Coverage
  - Integrate into Eclipse the functionality of Gcov
    http://gcc.gnu.org/onlinedocs/gcc-4.3.2/gcc/Gcov.html#Gcov

Source Code Features
Static Analysis

- Initially this part will most likely not be addressed because free open source tools don't yet provide mature static analysis tools
- GCC should soon have a proper plug-in architecture (http://gcc.gnu.org/wiki/GCC_PluginAPI) and static analysis tools
- Already some GCC based static analysis tools are showing promising results:
  - https://developer.mozilla.org/en/Treehydra
  - https://developer.mozilla.org/en/Dehydra
  - http://www.ggcc.info/, http://gcc.gnu.org/wiki/MiddleEndLispTranslator

Linux Reference Implementation Distribution

Linux Reference Implementation Distribution

- One click download/install of all necessary Eclipse plug-ins and GNU tools

- End users only need to download/install and they are up and running i.e. all the tools/plug-ins versions are compatible/tested

- Will be achieved by adding the Linux part to Wascana http://wascana.sourceforge.net/

Tracing/Monitoring Framework

Tracing/Monitoring Framework Approach

- Because of similarities in handling traces and logs, a tracing/monitoring framework would be advantageous.
- TPTP will be adapted to enable LTTng, systemTAP features

Note that in this section, the terms “trace” and “log” are used interchangeably.

Tracing/Monitoring Framework Scope

- Eclipse-based Tracing and Monitoring Framework providing extensible support for:
  - Tracing and Monitoring tool discovery
  - Tracing and Monitoring tool control
  - Data retrieval and storage
  - Data visualization
  - Analysis and correlation tool integration

- Framework Features
  - Remote and local tools support
  - Live, concurrent trace streams
  - Asynchronous events
  - Traces/logs that exceed available memory
  - External, host-based, libraries and analysis tools
  - Custom log parsers

Tracing/Monitoring Framework 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

Tracing/Monitoring Framework Tool Control

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

- Features
  - Basic tool control (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

Tracing/Monitoring Framework Data Retrieval and Storage

- Purpose
  - Collect and store tracing/monitoring data
  - Generic 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, …
    - Support for large files (bigger than available memory)

Tracing/Monitoring Framework Data Visualization

- Purpose
  - Provide a set of standard data visualization tools
  - Toolbox of widgets (language 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, sorting, …
  - Provide generic graphical widgets
    - Charts, Histograms, …
  - Extensible to allow for application-specific contents

Tracing/Monitoring Framework Analysis Tools Integration

- Purpose
  - Provide basic analysis functions
  - Support host-based, external analysis tools and libraries
- Features
  - Log comparison (regression testing, health monitoring, perf. 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
    - Set of APIs to access the monitoring data generically
    - Set of APIs to send the analysis results to UI views/widgets

Tracing/Monitoring Framework LTTng

- Integrate LTTng into the Eclipse tracing/monitoring framework: http://www.lttng.org/
- In addition to existing functionality, user space tracing is currently being added and will also be part of the Eclipse integration
- Features of the monitoring framework to be demonstrated with LTTng:
  - Tool control
  - Handling of very large trace files (Linux kernel traces)
  - Integration of a non-java custom parsing library
  - Generic presentation views
    - Raw trace data
    - Parsed trace data (tabular format)
  - Integration of a specialized analysis tool
    - Dependency analysis

Tracing/Monitoring Framework Other Integration Candidates

- SystemTap, http://sourceware.org/systemtap/

  A few Eclipse integrations already exist, Red Hat is planning to contribute an implementation in the Linux Tools project http://www.eclipse.org/projects/project-plan.php?projectid=technology.linux-distros

- Wireshark http://www.wireshark.org/

- Etc.

Emulator Integration?

Emulator Integration?

QEMU reference implementation?

Additional Information

Contact

- Dominique Toupin, dominique.toupin@ericsson.com
- François Chouinard, francois.chouinard@ericsson.com