Using RapiTime and RapiTask with RTEMS®
to verify real-time performance in a multicore environment

To show that your application meets hard real-time requirements, you need to know the worst-case response time of time-critical threads within the application. To establish this, you need to satisfy two conditions: use an RTOS with deterministic scheduling behaviour and know the worst case execution time (WCET) for each thread.

The first condition is met by RTEMS with real-time processes (RTPs). The second can be met by RapiTime WCET analysis of individual threads. RapiTask can also be used to identify and analyze the different tasks in a multicore environment. Part of RVS (Rapita Verification Suite), RapiTime goes beyond the capabilities of conventional performance profilers in several ways. Specifically, RapiTime:

- Collects distributions of execution time measurements taken directly from a multicore target.
- Allows the level of detail for measurements to be configured on the level of lines or blocks of code, functions or entire subsystems.
- Combines measurements with a structural model of the application’s source code to determine predicted WCET.
- Displays the predicted worst-case path through the code.
- Shows which lines of code have the biggest impact on WCET.

RapiTask gives the user the possibility of:

- Representing a high-level overview of multi-core software system scheduling.
- Locating rare timing events in multi-core systems, such as race conditions and interference.
- Core identification of executed tasks in a multi-core system.
- Workload identification of the system.

How RapiTime works

Figure 3 shows how RVS fits into the build process. The instrumenter embeds instrumentation points (Ipoints) into code between pre-processing and compiling the code.

Using data from the instrumenter, the structural analysis generates a complete structural model of the application. During testing, trace data in the form of instrumentation point identifiers and timestamps are collected. Once the trace data has been collected, it is combined with the structural model in the RapiTime Database, which can be examined in the Report Viewer.

RVS’s trace collection is via an open interface. This allows the trace to be captured in a variety of ways where it is captured by a logic analyzer or Rapita’s data logger, RTBx. For software capture, an on-target buffer.
Multi core WCET analysis and Scheduling analysis with RapiTime and RapiTask

Collecting the execution time from a specific task within RTEMS means eliminating the time spent executing other (pre-empting) tasks and interrupts from any time measurements.

For each core, a trace file is generated. This contains information about all of the tasks which were executed on the core. Eliminating time in other tasks requires an instrumented context switch mechanism. This is achieved by adding an instrumented function into several hooks which are called when a switch context operation occurs, such as init, create, switch or stop. In such functions, the involved tasks are identified and captured to enrich the analysis.

In situations where the execution time of several tasks is of interest, RapiTime can collect a trace of multiple tasks concurrently. This can then be demultiplexed into reports on separate tasks.

Using the same mechanism and the same produced trace, RapiTask can provide useful information about the multi core system under analysis; a high-level task overview, rare timing events and timing behavior.

Glossary

Execution time:
The time spent executing a specific thread between two points. This time excludes time spent executing any pre-empting threads or interrupts.

Instrumentation point identifiers (Ipoint Ids):
Constant values that allow an external view of the application’s progress to be made. Under RapiTime, Ipoint Ids do not need to be unique.

Response time:
The elapsed time between some initial stimulus to a system and that system’s response. In the case of a multi-threaded application, this may include the time spent executing several tasks and/or interrupts.

Worst-case execution time (WCET):
The longest possible execution time for a given thread.

Worst-case response time (WCRT):
The longest possible response time that can occur for a given system. Note for non real-time applications this can be unbounded.