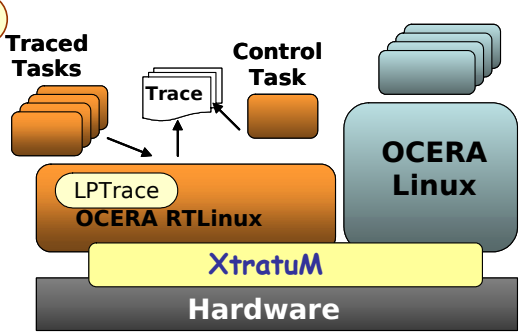
LPTracing and Metrics



As realtime applications become bigger and more complex, the availability of event tracing mechanisms becomes more important in order to perform debugging and runtime monitoring. Recently, IEEE has incorporated tracing to the facilities defined by the POSIX® standard. The result is called the POSIX Trace standard.

Light Weight Tracing is an reduced implementation of the POSIX Trace which permit to capture the traces of a embedded real-time operating system and the running applications.

The tracing facility plays an important role in the OCERA architecture. Besides its primary use as a debugging and tuning tool, the tracing component jointly with the application-defined scheduler component constitute the key tools for building fault-tolerance mechanisms. This is the first implementation of the Standard.

Features

- Traces can be analyzed on-line and off-line
- Off-line traces can be processed by **Metrics** which extract predefined or used-defined metric
- Metrics implements an automaton and generates a **Metrics** file
- Metrics are generated using a XML format

The contents of a trace stream can be analyzed while the tracing activity takes place or they can be analyzed later, once the tracing activity has been completed. Events stored in a trace stream represent system actions such as context switches, hardware interrupts, state changes, etc. In order to extract metrics from these events, it is necessary to know how the execution of the system generates these events, and normally this information is only known by the system designer. In order to solve this problem, this component implements a metric extraction engine and provides an application interface for using this engine. This interface allows the programmer to obtain predefined system metrics from trace streams without it being necessary for the programmer to know the system implementation.

Metric Report

Example of a html file obtained from the XML **Metrics** process

Table of Contents

Distributions for Metric JOB_RESPONSE_TIME_CLOCK_NANOSLEEP

Metric Description: Time elapsed since a job is activated until this job ends. This metric asumes that the clock_nanosleep function is used to release a job.

Detailed Distributions

Table 2.50. Thread 2 Distribution

N	Min	Max	Max – Min	Mean	Std.Dev.
666	0.000174976	0.000197248	0.000022272	0.000189063	0.000002831