

The introduction of tasks in the OpenMP programming model brings a new level of parallelism. This also creates new challenges with respect to its meanings and applicability through an event-based performance profiling. The OpenMP Architecture Review Board (ARB) has approved an interface specification known as the “OpenMP Runtime API for Profiling” to enable performance tools to collect performance data for OpenMP programs. In this paper, we propose new extensions to the OpenMP Runtime API for profiling task level parallelism. We present an efficient method to distinguish individual task instances in order to track their associated events at the micro level. We implement the proposed extensions in the OpenUH compiler which is an open-source OpenMP compiler. With negligible overheads, we are able to capture important events like task creation, execution, suspension, and exiting. These events help in identifying overheads associated with the OpenMP tasking model, e.g., task waiting until a task starts execution or task cleanup etc. These events also help in constructing important parent-child relationships that define tasks’ call paths. The proposed extensions are in line with the newest specifications recently proposed by the OpenMP tools committee for task profiling.