

Adaptive OpenMP Task Scheduling Using Runtime APIs and Machine Learning

Abstract

Task-based programming models adopt different scheduling strategies to exploit parallelism in irregular applications. These scheduling strategies differ in terms of exploiting data locality, maintaining load balance, and minimizing overhead. OpenMP tasks allow programmers to express unstructured parallelism at a high level of abstraction and make the runtime responsible about the burden of scheduling parallel execution. For irregular applications, the performance of task scheduling cannot often be predicted due to the nature of application, the used compiler, and the platform/architecture dependencies. In this work, we introduce an automatic, portable, and adaptive runtime feedback-driven framework (APARF) that combines standard low-level tasking runtime APIs, a developed profiling tool, and a hybrid machine learning model. We employ APARF to select the optimum task scheduling scheme of any given application using similarity analysis through the correlation between the captured runtime APIs with low profiling costs. Our hybrid model predicts the best scheduling strategy for a variety of unseen applications with an average accuracy of 93%, while maintaining a 100% training accuracy. An average performance enhancement of 25% was obtained compared with the default configuration, when APARF was applied on different unseen programs. APARF was examined against a real application (Molecular Dynamics), where we achieved up to 31% performance improvement. Compared to Intel, PGI and GNU compilers, our predicted scheme achieved better performance in most cases.