
How much static knowledge do dynamic schedulers need?

Olivier Beaumont*¹

¹RealOpt (INRIA Bordeaux - Sud-Ouest) – CNRS : UMR5251, Université de Bordeaux, INRIA – France

Abstract

The goal of this talk is to provide an analysis and comparison of static and dynamic strategies for task graph scheduling on platforms consisting of heterogeneous and unrelated resources, such as GPUs and CPUs. Static scheduling strategies, that have been used for years, suffer several weaknesses. First, it is well known that underlying optimization problems are in general NP-Complete, what limits the capability of finding optimal solutions to small cases. Second, parallelism inside processing nodes makes it difficult to precisely predict the performance of both communications and computations, due to shared resources and co-scheduling effects. Recently, to cope with these limitations, many dynamic task-graph based runtime schedulers (StarPU, StarSs, QUARK, PaRSEC) have been proposed. Dynamic schedulers base their allocation and scheduling decisions on the one side on dynamic information such as the set of available tasks, data location and the state of the resources and on the other hand on static information such as task priorities computed from the whole task graph. In this talk, I will consider two different linear algebra kernels, namely Matrix Multiplication and Cholesky Factorization to illustrate the respective advantages and limitations of static and dynamic solutions. In the case of Matrix Multiplication, we will more specifically study the impact of dynamism and runtime strategies on the overall communication amount. In the case of Cholesky Factorization, we will concentrate on the impact of heterogeneous and unrelated resources on resource allocation.

*Speaker