September 2016

State-Aware Concurrency Throttling

Daniele DE SENSI [a,1], Peter KILPATRICK [b] Massimo TORQUATI [a]
[a] *Computer Science Department, University of Pisa*
[b] *Computer Science Department, Queen's University Belfast*

Abstract. Reconfiguration of parallel applications has gained traction with the increasing emphasis on energy/performance trade-off. The ability to dynamically change the amount of resources used by an application allows reaction to changes in the environment, in the application behavior or in the user's requirements. A popular technique consists in changing the number of threads used by the application (*Dynamic Concurrency Throttling*). Although this provides good control of application performance and power consumption, managing the technique can impose a significant burden on the application programmer, mainly due to state management and redistribution following the addition or removal of a thread. Nevertheless, some common state access patterns have been identified in some popular applications. By leveraging on this knowledge, we will describe how it is possible to simplify the state management procedures following a Concurrency Throttling operation.

Keywords. Power-Aware Computing, Concurrency Throttling, Data Stream Processing

1. Introduction and Motivation

In recent years power consumption management has become a major concern for data center operators due to economic cost, reliability problems and for environmental reasons [16]. To dynamically control the power consumption of an application, a commonly used approach consists in changing the amount of resources allocated to the application (i.e. its *configuration*). This can be achieved by acting on the number of cores allocated to the application [3], on the clock frequency of these cores [2], on the types of cores used by the application [18] or on a combination of those [11]. The application is monitored throughout its execution and the configuration is adapted by considering the user preferences, the interference caused by external factors (e.g. other applications) and possible workload fluctuations.

Concurrency throttling is one of the most commonly used techniques [7,4,17,19] and consists in dynamically changing the number of threads used by the application. By reducing the number of threads, if we had one thread for each core, the application may release some cores, thus allowing the operating system to shut them down and so reduce the overall power consumption of the computing architecture. In some cases, like for example in OPENMP, this can only be done between two parallel sections (e.g. between two different *parallel for*s). On the other hand, other approaches apply this technique

[1] Corresponding Author: Daniele De Sensi, Computer Science Department, University of Pisa, Italy; E-mail: desensi@di.unipi.it.