Multi-DaC Programming Model

A variant of Multi-BSP Model for divide-and-conquer Algorithms

Abdorreza Savadi
Ferdowsi University of Mashhad (FUM), Mashhad, Iran
savadi@um.ac.ir

Morteza Moradi Hossein Deldari
Islamic Azad University-Mashhad Branch (IAUM), Mashhad, Iran
moradi.edu@gmail.com/hd@um.ac.ir

Abstract

Nowadays, the evolution of multi-core architectures goes towards increasing the number of cores and levels of cache. Meanwhile, current typical parallel programming models are unable to exploit the potential of these processors efficiently. In order to achieve desired performance on these hardwares we need to understand architectural parameters appropriately and also apply them in algorithm design. Computational models such as Multi-BSP, illustrate these parameters and explain adequate methods for designing algorithms on multi-cores. One of the most applicable categories of problems is Divide-and-Conquer (DaC) that needs to be adapted by such model for implementing on these systems.

In this paper, we have attempted to make a mapping between DaC tree and the Memory Hierarchy (MH) of multi-core processor. Multi-BSP model inspired us to introduce Multi-DaC programming model. Analogous to Multi-BSP analysis, lower bounds for communication and synchronization costs have been presented in the paper respecting DaC algorithms. This work is a step towards making multi-core programming easy and tries to obtain correct analysis of DaC algorithm behavior on multi-core architectures.

Categories and Subject Descriptors D.1.3 [Concurrent Programming]: Parallel Programming; D.3.2 [Language Classifications]: Concurrent, Distributed and Parallel Languages; D.4.8 [Performance]: Modeling and prediction

General Terms Multi-BSP, parallel, algorithms, Divide-and-Conquer, multi-core architectures, Skeletal Programming, cache, Memory Hierarchy

Keywords Multi-BSP, parallel, Divide-and-Conquer, multi-core architectures, cache, Memory Hierarchy

1. Introduction

The multi-core architectures were introduced as a remedy to avoid the limitations which were made during development of parallel architectures such as high energy consumption, heat dissipation, and the memory-wall problem. Traditional programming models don’t consider existence of shared memory hierarchy on a processor therefore using these models will produce undesired results. In void of proper multi-core software development models, designing efficient programs have become awkward, so there is a demand for revising current models or introducing new ones.

On the other hand, it is predicted that in the forthcoming years the number of cores on a chip will increase in large numbers and MH will get larger in more levels respectively. Hence, if these architectures want to stabilize their place among parallel systems have to be accompanied by appropriate programming tools and models because the complexity of efficient algorithm design for them will get higher soon.

In order to simplify programming, it seems we should distinguish efficient system parameters to be able to add a layer over architecture to hide its complexity from programmers. Programming models can well show efficient performance parameters and are able to lessen the difficulty of algorithm design by explaining the style of programming. So, designing a robust flexible model that can fit to all needs and future architectures is a high goal.

In this paper, we attempted to step forward towards achieving the mentioned goal by presenting a programming model for Divide-and-Conquer (DaC) algorithms which are widely applied species of algorithms. The proposed model is based on adapting DaC to a computational model for multi-cores called Multi-BSP. The benefit of our model is that it can anticipate the cost of algorithm execution on hardwares which have a hierarchical shared memory.

In the following section the related works are presented. In Section 3 we introduce the basic description of our model. In Section 4 we describe mapping algorithm and in Section 5 scheduling algorithm used in the model is presented. Lower bounds for estimating the costs are demonstrated in Section 6. Two case studies, merge sort and convex hull, are evaluated in Section 7. The conclusions and future works appear in the last section.

2. Related Works

With the advent of multi-cores, the need to exploit their potential was appeared. One of the efforts undertaken for fulfilling the above need is modeling the hierarchical memory parallel architectures to make the ability of their performance analysis. Computational models for parallel and distributed systems such as PRAM[8], LogP[6], QSM[13], and BSP[17] can be the apt candidate for the purpose, but as they have not included the hierarchical memory on the scale of a single chip, they do not have the required efficacy for these architectures. Recent works have been done on hierarchical single-core processors. For example, PEM is a two level multi-processor model in which, the first level is related to private cache while the second level is an external shared memory among processors. The model is based on two previous models of Ideal Cache Model[1] and Two Level I/O Model[9]. Multi-core Cache Model[5] is more extended than PEM. In stride of evolution of multi-core architectures, it is determined that with the increase in