A Survey on Parallel Computing and its Applications in Data-Parallel Problems Using GPU Architectures

Cristóbal A. Navarro¹,²,*, Nancy Hitschfeld-Kahler¹ and Luis Mateu¹

¹Department of Computer Science (DCC), Universidad de Chile, Santiago, Chile.
²Centro de Estudios Científicos (CECS), Valdivia, Chile.

Received 11 January 2013; Accepted (in revised version) 1 August 2013
Available online 10 September 2013

Abstract. Parallel computing has become an important subject in the field of computer science and has proven to be critical when researching high performance solutions. The evolution of computer architectures (multi-core and many-core) towards a higher number of cores can only confirm that parallelism is the method of choice for speeding up an algorithm. In the last decade, the graphics processing unit, or GPU, has gained an important place in the field of high performance computing (HPC) because of its low cost and massive parallel processing power. Super-computing has become, for the first time, available to anyone at the price of a desktop computer. In this paper, we survey the concept of parallel computing and especially GPU computing. Achieving efficient parallel algorithms for the GPU is not a trivial task, there are several technical restrictions that must be satisfied in order to achieve the expected performance. Some of these limitations are consequences of the underlying architecture of the GPU and the theoretical models behind it. Our goal is to present a set of theoretical and technical concepts that are often required to understand the GPU and its massive parallelism model. In particular, we show how this new technology can help the field of computational physics, especially when the problem is data-parallel. We present four examples of computational physics problems; n-body, collision detection, Potts model and cellular automata simulations. These examples well represent the kind of problems that are suitable for GPU computing. By understanding the GPU architecture and its massive parallelism programming model, one can overcome many of the technical limitations found along the way, design better GPU-based algorithms for computational physics problems and achieve speedups that can reach up to two orders of magnitude when compared to sequential implementations.

AMS subject classifications: 68W10, 65Y05, 68Q10, 68N19, 68M14, 00A79, 81Vxx

*Corresponding author. Email addresses: crinavar@dcc.uchile.cl (C. A. Navarro), nancy@dcc.uchile.cl (N. Hitschfeld), lmateu@dcc.uchile.cl (L. Mateu)
Key words: GPU computing, parallel computing, computing models, algorithms, data parallel, massive parallelism, Potts model, Ising Model, collision detection, $n$-body, Cellular Automata.

Contents

1 Introduction 286
2 Basic concepts 288
3 Performance measures 289
4 Parallel computing models 295
5 Parallel programming models 300
6 Architectures 303
7 Strategy for designing a parallel algorithm 309
8 GPU Computing 310
9 Examples of spatial and tiled GPU compatible problems 314
10 Latest advances and open problems in GPU computing 320
11 Discussion 321

1 Introduction

For some computational problems, CPU-based algorithms are not fast enough to give a solution in a reasonable amount of time. Furthermore, these problems can become even larger, to the point that not even a multi-core CPU-based algorithm is fast enough. Problems such as these can be found in science and technology; natural sciences [52, 108, 116] (Physics, Biology, Chemistry), information technologies [115] (IT), geospatial information systems [11, 66] (GIS), structural mechanics problems [12] and even abstract mathematical/computer science (CS) problems [84, 98, 101, 109]. Today, many of these problems can be solved faster and more efficiently by using massive parallel processors.

The story of how massive parallel processors were born is actually one of a kind because it combines two fields that were not related at all; computational science and video-game industry. In science, there is a constant need for solving the largest problems in a reasonable amount of time. This need has led to the construction of massively parallel super-computers for understanding phenomena such as galaxy formation, molecular dynamics and climate change, among others. On the other hand, the video-game industry is in a constant need for achieving real-time photo-realistic graphics, with the major restriction of running their algorithms on consumer-level computer hardware. The need of realistic video-games led to the invention of the graphics accelerator, a small parallel processor that handled many graphical computations using hardware implemented functions. The two needs, combined together, have given birth to one of the most important hardware for parallel computing; the GPU.