Chapter 7
CPU-GPU Computing:
Overview, Optimization, and Applications

Xiongwei Fei
Hunan City University, China

Wangdong Yang
Hunan University, China

Kenli Li
Hunan University, China

Keqin Li
State University of New York, USA

ABSTRACT

Heterogeneous and hybrid computing has been heavily studied in the field of parallel and distributed computing in recent years. It can work on a single computer, or in a group of computers connected by a high-speed network. The former is the topic of this chapter. Its key points are how to cooperatively use devices that are different in performance and architecture to satisfy various computing requirements, and how to make the whole program achieve the best performance possible when executed. CPUs and GPUs have fundamentally different design philosophies, but combining their characteristics could avail better performance in many applications. However, it is still a challenge to optimize them. This chapter focuses on the main optimization strategies including "partitioning and load-balancing", "data access", "communication", and "synchronization and asynchronization". Furthermore, two applications will be introduced as examples of using these strategies.

INTRODUCTION

As processor speeds steadily increase, high energy consumption and heat dissipation become harder to mitigate. At the same time, design engineers must devise processors with multiple cores to satisfy the demand for high performance. Central Processing Units (CPUs) and Graphics Processing Units (GPUs) have evolved to support more cores than ever, but the development of the two processor types follows different philosophies. CPU development focuses on low latency by using sophisticated control, while GPU development aims for high performance by using a greater number of simple cores. GPUs often serve as coprocessors with CPUs. In many supercomputers, such as Tianhe and Titan, CPUs and GPUs cooperate together to produce powerful computing. For example, each computing node of the Tianhe-1A has two Intel® Xeon® X5670 CPUs and one Nvidia Tesla™ M2050 GPU. In personal computers, the typical combination of the CPU and GPU provides low price and high performance.

DOI: 10.4018/978-1-5225-0287-6.ch007

The popularity of such heterogeneous systems necessitates adaptation of applications to optimize their performance. Three types of adaptation are necessary. First, because of the multiple cores, the applications must be adapted for parallel processing to maximize use of available resources. Second, effort should be made to combine the properties of the CPU and the GPU to partition and map tasks. Third, focus on specific techniques to enhance performance as much as possible. In this chapter, four techniques are introduced:

- Workload balancing and distribution between the CPU and the GPU.
- Efficient use of hierarchical memories, such as hiding data access latency, coalescing data access, efficient shared memory, virtual addressing between GPU and CPU, etc.
- Reduction of communication overhead, such as by zero copy or streaming data transmission.
- Asynchronization such as concurrent copying and execution, sub-task pipeline, and so on.

This chapter discusses CPU and GPU heterogeneous hardware and hybrid OpenMP and CUDA software. They are two sides of a coin, so this chapter discusses the architecture of the CPU and the GPU, and the mixing method of OpenMP and CUDA. Performance measurement techniques are described to evaluate the performance of applications running on such heterogeneous computers. Specifically, the metrics of execution time, bandwidth, occupancy, and speedup will be provided. Furthermore, two synthesized examples –parallel Sparse Matrix-Vector multiplication (SpMV) and Advanced Encryption Standard (AES) –are provided to demonstrate the use of some of these techniques on heterogeneous computers and to provide some inspiration for development of hybrid parallel applications. High-performance applications for heterogeneous computing environments can be achieved with a thorough understanding of the essential properties of the hardware and the application and the use of specific optimization techniques.

The remainder of this chapter is organized as follows: the next section briefly discusses the advances of heterogeneous and hybrid computing models. Then, section "HETEROGENEOUS AND HYBRID COMPUTING MODELS" describes in more details the heterogeneous and hybrid computing models, including GPU and CPU hardware architectures, and CUDA and OpenMP platforms. After that, section "OPTIMIZATION STRATEGIES" provides some optimization strategies, such as partitioning and load-balancing, data access optimization, reducing communication in heterogamous and hybrid environments; performance evaluation is also provided in this section. Section "APPLICATIONS" offers two typical applications of heterogeneous and hybrid computing models. One of these applications involves hybrid parallel matrix computing, and the other involves hybrid parallel encrypting /decrypting. Finally, the chapter concludes.

BACKGROUND

Heterogeneous and hybrid computing refers to the use of various cores and resources cooperating to accomplish a common computing task. In common usage, "heterogeneous computing" denotes that the hardware has a mixed architecture; whereas, "hybrid computing" means that the software is split into various parts and mapped to the hardware.

Heterogeneous hardware architecture has developed to address four issues or limitations of traditional homogeneous computing: