RECENT TRENDS IN COMPUTATIONAL SCIENCE AND ENGINEERING

Edited by **M. Serdar Çelebi**

Contents

Chapter 1

High-Performance Computational and Information Technologies for Numerical Models and Data Processing

Darkhan Akhmed-Zaki, Madina Mansurova, Timur Imankulov, Danil Lebedev, Olzhas Turar, Beimbet Daribayev, Sanzhar Aubakirov, Aday Shomanov and Kanat Aidarov

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.73836

Abstract

This chapter discusses high-performance computational and information technologies for numerical models and data processing. In the first part of the chapter, the numerical model of the oil displacement problem was considered by injection of chemical reagents to increase oil recovery of reservoir. Moreover the fragmented algorithm was developed for solving this problem and the algorithm for high-performance visualization of calculated data. Analysis and comparison of parallel algorithms based on the fragmented approach and using MPI technologies are also presented. The algorithm for solving given problem on mobile platforms and analysis of computational results is given too. In the second part of the chapter, the problem of unstructured and semi-structured data processing was considered. It was decided to address the task of n-gram extraction which requires a lot of computing with large amount of textual data. In order to deal with such complexity, there was a need to adopt and implement parallelization patterns. The second part of the chapter also describes parallel implementation of the document clustering algorithm that used a heuristic genetic algorithm. Finally, a novel UPC implementation of MapReduce framework for semi-structured data processing was introduced which allows to express data parallel applications using simple sequential code.

Keywords: fragmented algorithm, high-performance visualization, computational algorithms on mobile platforms, MPI, unstructured and semi-structured data processing, n-gram extraction, MapReduce framework

1. Introduction

With development of computer technology level and high-performance systems across the world, efficiency of solving problems in the field of fundamental and engineering research is increasing. Annual development of mathematical models allows to study physical and chemical processes in greater detail. Modern numerical methods are also being developed for solving applied problems and an amount of calculations are increasing. In this regard, using high-performance computing and computational technologies to solve applied problems with each year becomes more relevant.

In the middle stages of the development of high-viscosity oil fields, the problem of decreasing oil recovery becomes an issue. Increasing oil recovery of reservoirs remains one of urgent tasks at the moment. Methods of injecting polymers and surfactants into an oil reservoir are currently widely used in the oil industry and are considered as one of the effective methods for increasing the oil recovery of reservoirs [1, 2]. Therefore, the problem of oil displacement process by polymer and surfactant flooding was perceived as being a task for given working group.

Parallel implementation of the oil displacement problem and applied method appears to be complex problem of system parallel programming because it requires to provide synchronization of separate computational processes, network data transfer, etc. In order to decrease complexity of such parallel programs, technology of fragmented programming and its implementation called LuNA (Language for Numerical Algorithms) were adopted [3].

Visualization is an integral part of the analysis during the processing of the scientific data. It has a significant role in large-scale computational experiments on modern high-performance engines. The amount of data obtained in such calculations can reach several terabytes. Such system requires a well-designed and implemented client-side visualization module taking into account its client orientation. So such programming module was applied using modern visualization technology Vulkan API [4].

Nowadays full computational potential of mobile devices almost not used because of devices being idle for extended periods during a day. There are number of projects such as Berkeley Open Infrastructure for Network Computing (BOINC) which use excessive computational capabilities of PCs and mobile devices across the globe [5]. While provisioning services for its customers as integrator of numerous computational resources for solving their problems, the processing itself was conducted using only CPUs. Many recent mobile devices are equipped with powerful GPUs generally used for 3D graphics rendering. By efficient usage of mobile GPUs, one can achieve much more performance from a single device therefore increasing overall productivity of such integrational computations. This task requires the mobile software installed to be able to use capabilities offered by GPUs. Following researchers studied issues and possibilities related to exploit GPU capabilities of mobile devices in integrated computations: Zhao [6], Montella et al. [7].

Because of the rapid progress on computer-based communications and information dissemination, large amounts of data are daily generated and available in many domains. The purpose of the research presented in the second part of the chapter is to develop models and algorithms

http://dx.doi.org/10.5772/intechopen.73836

for unstructured and semi-structured data processing using high-performance parallel and distributed technologies.

Today huge amount of information are being associated with the web technology and the internet. To gather useful information from it, these text has to be categorized. Text categorization is a very important technique for text data mining and analytics. It is relevant to discovery various different kinds of knowledge. It is related to topic mining and analysis. It is also related to opinion mining and sentiment analysis, which has to do with knowledge discovery about the observer, the human sensor. The observer based on the content they produce can be categorized. The indexing influences the ease and effectiveness of a text categorization system [8]. The simplest indexing is formed by treating each word as a feature. However, words have properties, such as synonymy and polysemy. These have motivated attempts to use more complex feature extraction methods in text categorization tasks. If a syntactic parse of text is available, then features can be defined by the presence of two or more words in particular syntactic relationships. Nowadays authors [9–11] have used phrases (n-grams), rather than individual words, as indexing terms. In this work, the task was also addressed to n-gram text extraction which is a big computational problem when a large amount of textual data is given to process. In order to deal with such complexity, there was a need to adopt and implement parallelization patterns.

> *The chapter also focuses on research related to the application of genetic algorithm for document clustering. Genetic algorithms make it possible to take into account peculiarities of the search space by adjusting the parameters and selecting the best solutions from the solutions obtained by the population [12–14]. Clustering algorithm is based on the assessment of the similarity between objects in a competitive situation. Since clustering problem solution requires large computational resources parallelization on the stage of genetic algorithm for setting the coefficients in the formula of similarity measures was performed, as well as on the stage of data clustering.*

MapReduce technology has shown a great potential in dealing with large-scale data processing problems [15, 16]. Such batch-oriented MapReduce systems as Apache Hadoop, however, lacks efficiency in dealing with iterative problems. The main bottleneck can be attributed to slow disk operations arising in data storage after current iteration in a distributed file system. Number of solutions that deal with that problem has been proposed in a literature, including ones that propose novel techniques that optimize loops [17] and ones that try to keep static data [18]. Recently introduced novel approaches rely mostly on in-memory processing mechanisms [19, 20]. Also some types of data parallel problems require efficient communication between parallel workers in order to be able to implement specific nature of the data exchange patterns. In such a way, it is necessary to consider other parallel programming models that can be effectively combined with MapReduce.

Partitioned global address space (PGAS) model presents an interesting approach to deal with data communication problem. In PGAS model, a global memory is divided among threads with different choices of memory to thread mappings. Several works introduced different approaches to implement MapReduce functionality in a frame of PGAS model. For example,