Some aspects of studying of High Performance Technologies by Moldavian young researchers

Mircea Petic[1,2], Inga Titchiev[1]
Irina Frinea[1], Nicolai Iliuha[1],Veaceslav Macari[1]

[1]Institute of Mathematics and Computer Science, Chisinau, REPUBLIC OF MOLDOVA, E-mail: inga.titchiev@gmail.com, irina.frinea@math.md, nicolai.iliuha@renam.md, vmacari@yandex.ru

[2]Faculty of Computer Sciences, Alexandru Ioan Cuza University of Iasi, ROMANIA, Iasi, E-mail: petic.mircea@gmail.com

Abstract – Aim of this article is to present an overview of the project for young researchers - "Systems and Technologies of Distributed Processing of Information and evaluating the effectiveness of their Use" and show which are the goals of project what has been achieved until moment and what are the expected results. The description of available at the Institute of Mathematics and Computer Science HPC infrastructure, adopted methodology and stages of achievment are given. A special compartiment is devoted to workflow Petri nets in modeling of parallel processes. At the end we present a method of image template matching.

Key words – distributed systems, parallel algorithms, distributed processing technologies, High Performance Computing (HPC), OpenCV.

I. Introduction

Many efforts have been made in order of modeling of hardware and software components. Especially in the last four decades a formal model of software component were performed. Many of these efforts have been focused on the analysis, specification and description of sequential programs. Formal description of systems with concurrent processes has become a new research direction. The use of parallel processing techniques reduce costs, processing time, increase application performance and led to their successful use in various fields (bioinformatics, physics, mathematical modeling, web servers and database, in business decisions, medicine).

In this context, the idea emerged was to develop a project that would help young researchers in obtaining experience required for more complex projects related to parallel programming.

The paper is structured in the following way: first we will present some technical equipment available at the Institute of Mathematics and Computer Science to run applications with parallel technologies. Then the description of the adopted methodology and stages of achievment are given. A special compartiment is devoted to workflow Petri nets in modeling of parallel processes. At the end we present a method of image template matching.

II. Technical equipment

To train young professionals getting started with cluster systems, to create and run applications with different technologies of parallel computing at the Institute of Mathematics and Computer Science the 48 core IMI-RENAM cluster is used [1]. At this cluster on virtualization platforms next Home Training Infrastructures were deployed:

- MS Windows Compute Cluster 2003, 4 Nodes, 12 Cores (CPUs: QuadCore Intel Xeon E5335 2,0 GHz, QuadCore Intel Xeon E5310 1,6 GHz) to run different tasks serial, parallel, parametric sweep and task flow;
- Grid-Site (Linux): MD-02-IMI, 4 Worker Nodes, total 16 VCPU, 1 GB RAM per 1 VCPU – to test applications and prepare them for porting from local clusters to EGI GRID and to HP-SEE regional resources [2];
- on Virtual Machine: 64 bit Scientific Linux 6.3; Intel(R) Parallel Studio XE 2011 (4 cores, 4 Gb RAM) – for compiling and debugging of applications.

For GPU-oriented applications was prepared workstation with 2 GPU (QuadCore Intel Core 2 Quad Q6600, RAM 4 Gb, CUDA on nVIDIA GeForce 8600 GT, nVIDIA GeForce GTS 450)

III. Used methodology

In order to achieve research the following methods will be used:

1. Open MP - to control explicit and shared memory parallelism with multiple threads of execution,
2. CUDA - support the communication of "peer-to-peer" between GPUs or a single server stations work, allowing easier and faster programming systems "multi-GPU". Provides a collection of algorithms C++ open-source and data structures that facilitate easy programming for C++ developers [3].
3. MPI (Message Passing Interface) is a specification for those who use computer resources with non-shared memory like Grids.

Main resources of the European HPC infrastructure support parallel programming paradigms like MPI and OpenMP.

IV. Stages of achievement

For 2013 are planned the execution of two stages [1]:

1. Theoretical foundations of distributed processing systems and information technologies.
2. Assess to the systems and technologies for distributed processing of information. Training program to get experience in parallel programming and various HPC resources using.

The first stage involves the study and analysis of theoretical and practical problems specific to parallel and distributed computation, study of parallel and distributed architectural models. Familiarity with specific technologies for parallel and distributed computing like OpenMP, MPI, CUDA.

The second stage involves the identification of problems related to the field of distributed processing systems and technologies. Carrying out the analysis and evaluating the effectiveness of parallel/distributed computing systems in different conditions.

V. Workflow Petri nets in modeling of parallel processes

For modeling parallel processes by means of Petri Nets first of all we will examine how a single process can be represented by a Petri net, and then combining corres-

ponding Petri nets for each processes we obtain the corresponding Petri net system with concurrent processes. One process is described by a program. This program can be written in many languages, in which they represent two separate aspects of the process: computation and control.

The computation dealing with arithmetic and logical operations, current inputs and outputs and general handling of memory locations and values. On the other hand control not deals with the values or performed computations, but only with the order in which they were made. Petri nets can be used to model control structures of software, sequences of instructions and information how and computations but not values.

A standard representation of the control structure of a program is using a logical scheme. A logical scheme represent the control flow in a program. The structure of the program, is a uninterpretable logical scheme. Thus, showing how a program logical scheme can be represented by a Petri net, we showed how a uninterpretable program can be represented by a Petri net. A logical scheme of the program is very similar with a Petri net because it is consists of nodes (by two types: the decision, represented by rhombus, and computing, represented by rectangles) and arcs joining them.

A simple way to execute a program represented by the logical scheme is to introduce a token representing the current instruction. When is executing the instruction, the token moves in scheme. In order of modeling more accurately information flow we consider workflow Petri nets [4]. Nodes from logical scheme which represents input/output blocks will be presented by source place/sink place. Nodes which represents assignments (computing) blocks will be presented by transitions because all of them execute some action. Node from logical scheme which represents decision blocks will be presented by one place (precondition), two transition and two places (postconditions).

VI. Image template matching

A particular case of using systems and technologies that process information distributedly is the processing of images (namely, pre-processing, recognition of forms, identification of some specific features) that are obtained after an ultrasouns examination.

Another experiment deals with template matching. It is a technique for finding areas of an image that match to a template image. In this case we need two primary components:

1. Source image – the image in which we expect to find a match to the template image;

2. Template image – the patch image which will be compared to the template image.

We move the patch one pixel at a time (left to right, up to down). At each location, a metric is calculated so it represents how appropriate the match at that location is (or how similar the patch is to that particular area of the source image). So, our goal in this experiment is to detect the highest matching area.

In our case we used:

1. the OpenCV function matchTemplate to search for matches between an image patch and an input image;

2. the OpenCV function *minMaxLoc* to find the maximum and minimum values (as well as their positions) in a given array.

The reduction of waiting time plays a vital role in taking a decision about giving fast and adequate treatment.

Conclusion

Efficient processing of parallel algorithms in time and space can bring significant benefits in various fields such as medicine, biology, astronomy, forensics which are based on the use and interpretation of specific images, also can be used in the training of specialists (education) of different areas of interest.

The expected outcome of the project would be the development of parallel processing algorithms in order to solve specific problems (eg. parallel processing of images in order to increase performance and speed up medical examinations).

Further research will be directed at identifying specific problems that can be solved using systems and technologies of distributed processing of information.

Acknowlegement

This paper was elaborated within the project 13.819.18.06A Systems and Technologies of Distributed Processing of Information and evaluating the effectiveness of their Use.

References

[1] I. TITCHIEV, N. ILIUHA, “Rising Skill of young researches of Moldova in using High Performance Technologies”, Proceedings of the Conference “Computer Science and Information Technologies” (CSIT2013), Yerevan, Armenia, 23-27 September 2013, ISBN 978-5-8080-0797-0, pp. 423-424.

[2] P. BOGATENCOV, B. RYBAKIN, G. SECRIERU, N. ILIUHA, “Utilization of the Computational Resources Provided by HP-SEE Project”. International Workshop on Intelligent Information Systems. Proceedings IIS. 20-23 August, Chisinau, IMI ASM, 2013, pp. 60-64. ISBN 978-9975-4237-1-7

[3] B. RYBAKIN, “Parallel programming for graphics accelerators.” Under edition by Academician V.B. Betelin. Moscow, SIISI RAS, 261 p., 2011.

[4] W. AALST, K. HEE. ”Workow Management - Models, Methods,and System”, MIT Press, 2002.