Distributed Computing using GPU

Polina Mishchenko
Faculty of Automatics and computer facilities
Novosibirsk State Technical University
Novosibirsk, Russia
mpv-777@ya.ru

Alexander Nezvanov
Faculty of Automatics and computer facilities
Novosibirsk State Technical University
Novosibirsk, Russia
nezvanovml@mail.ru

Marat Khairetdinov
Faculty of Automatics and computer facilities
Novosibirsk State Technical University
Novosibirsk, Russia
xajretdinov@corp.nstu.ru

***Abstract*—This article shows results of a research of efficiency of implementation of distributed computing using graphics processing unit (GPU), methods of involvement of graphic processors in the course of functioning of distributed computing systems. Experiments were made in the context of application of the discussed approach to integration of separate calculators for a solution of applied resource-intensive tasks, in particular a problem of search of statistically significant represented biological characteristics of genes from the set. As the main means of the analysis the permutable (randomized) test, known in biological information science, is used.**

Keywords— distributed computing, graphic processor, permutable test, CPU, GPGPU, OpenCL

I. Introduction

Use of supercomputer technologies for modeling of difficult processes and systems became an integral part of researches practically in all branches of science and technicians. Parallel implementations of computational modeling are the high-performance tool in computing experiments. In this work approach on use of graphic processors as means of parallelization at a solution of some tasks is considered. Today there are two explicit leaders in development of graphic processors and technologies of work with them – NVIDIA, ATI. Each of them offers tools for effective use of opportunities of the video card used as the coprocessor to the central processor. The efficiency of such "sheaf" is confirmed in a number of works [1, 2]. NVIDIA CUDA and Open CL are considered as the most perspective software. In view of closeness of the CUDA software package the OpenCL technology is of the greatest interest. OpenCL is a technology of creation of parallel programs with use of different architecture of graphic processors. It includes the programming language based on C99 standard and the application programming interface of API-Application Programming Interface. Open CL provides parallelism at the level of instructions and at the level of data and is implementation of the technology of sharing of resources of the central and graphic processors. This technology is completely open standard, its use is not assessed with runtime royalty fees.

The offered method of integration of diverse computing nodes into the uniform heterogeneous environment means combination of territorially remote calculators connected to the Internet. At the same time, separate workstations and local area networks can act as calculators [3]. It should be noted that traditionally parallel computings are implemented or within one local area network or by the organization of remote access to high-performance clusters. [4]. Feature of the offered method of the organization of calculations is algorithmically put decentralization of process control of calculations that allows to increase fault tolerance of computing environment in general [5]. The idea of combination of computing potential of the central and graphic processors of personal computers in the conditions of their geographically distributed structure and heterogeneity of architecture is the cornerstone of the developed technology. Computing loading at the same time can be distributed both for regular, and for a background operation mode.

II. Analysis of Technologies of Application of GPU

At the beginning of the 2000th years in the world of the computer industry there was a GPGPU technology (from English general-purpose graphics processing units) which actively develops also until today. Its basis is transfer of homogeneous algorithmic calculations on the graphic processor which, unlike the central processor possesses a set of the kernels allowing to parallelize calculations. Application of GPU allows to win considerably in productivity per watt of power consumption. Thanks to architectural features of GPU are optimized for parallel processing of data. Now there are developments in which both processors are united in one device called by APU (from English accelerated processing unit). In the computer industry there are several large products which implement the concept of GPGPU. Everything them can conditionally be divided into three categories: hardware-dependent (CUDA), program-dependent (the C AMP) and universal (OpenCL, OpenACC). All of them allow developers to create the applications using the graphic processor as computing block. Let's present short characteristic of each of technologies.

CUDA (abbr. from English Compute Unified Device Architecture) [6,7] – the technology allowing to make on GPU of calculation of general purpose at this GPU actually acts as the powerful coprocessor. NVIDIA CUDA technology perhaps the only development environment in programming language C which allows developers to create the software for a solution of difficult computing tasks for smaller time, thanks to the multinuclear computing power of graphic processors. CUDA gives to the developer the chance at discretion to organize access to a set of instructions of the graphic accelerator and to manage his memory, to organize on it parallel computings. The technology makes available to the developer the low-level, distributed and high-speed access to the equipment, giving the chance of creation of serious high-level tools, such as compilers, debuggers, mathematical libraries, software platforms.

978-1-7281-2896-2/19/$31.00 ©2019 IEEE

OpenCL (Open Computing Language) [8] — the framework giving to the programmer opportunities of writing of programs for implementation of parallel computings which can be started on CPU, GPU and chips of FPGA. Language C (C99 standard) and the interface of creation of programs is its part. OpenCL provides parallelism at two levels: instructions and data, is implementation of GPGPU technology. Completely open standard, for its use is not required a payment. It is possible to OpenCL to use as with processors (the most part of modern processors supports the standard), the video cards AMD, and with video cards from NVIDIA.

C++ AMP (Accelerated Massive Parallelism) [9] – library from Microsoft corporation which uses DirectX 11 for implementation of parallel computings with use of language C. Allows to transfer calculations to GPU, having made minor changes in the code of the program if the code is difficult for execution on the graphic accelerator, then it will be automatically executed on the central processor. Start of programs on any graphic processors supporting the DirectX 11 standard is possible, but works only under Windows that narrows the field of possible application.

OpenACC (Open Accelerators) [10] — the standard of parallel programming developed jointly by the Cray, CAPS, Nvidia and PGI companies. The standard describes a set of the directives of the compiler intended for simplification of creation of heterogeneous parallel programs as on the basis of the central processor, and graphic. As well as earlier standard Open MP, OpenACC is used for annotation of fragments of programs in the C, C and Fortran languages. By means of a set of directives of the compiler, the programmer notes code locations, which should be carried out in parallel or on the graphic processor, designates what of variables are the general and what individual for a flow, etc. Software to syntax is similar to OpenMP. The OpenACC standard allows the programmer to abstract from features of initialization of the graphic processor, questions of data transmission on the coprocessor and back, thereby reducing the required level of knowledge in the field of programming and considerably simplifying development.

The C++ AMP technology does not meet the requirement of cross-platform, which is required at start of calculations on diverse computers. The OpenACC standard meets the requirement, but is high-level and does not allow making tracking of runtimes of separate operations, such as data transfer from the calculator in the RAM and back and also GPU kernel operating time.

On the basis of the made analysis of technologies for GPU the OpenCL standard which satisfies to both requirements of cross-platform, and providing an opportunity to launch calculations on the central processor or the integrated graphic kernel without change of a program code was selected.

III. Problem of Definition of the Permutable Test

The solution of a number of large scientific tasks is connected with the analysis of large volume of data. The problem of genetic determination of genes connected with studying of a problem of heredity [11] is among them. For a solution of this task, use of graphic processors is reasonable [12-14]. The algorithm of the permutable test intended for search of statistically significant represented properties of genes consists of the following steps:

1) The list of genes associated is set:
   a) With the list of qualitative properties of these genes (by means of which selections of genes are set);
   b) With the list of the measured quantitative characteristics of genes;
2) For each of Boolean properties of genes according to the list of quantities the value of statistics

$$G_0(x_1, x_2, x_3, \dots, x_n) = \sum_{i=0}^{n} x_i, x_i \in D \qquad (1)$$

   where x – the measured quantitative characteristic of a gene (expression level, evolution speed, etc.), D – a set of the observed objects meeting an observation condition is calculated;
3) Accidental shift of quantitative characteristics between genes is carried out;
4) For the same genes with accidentally rearranged values

$$x_1, x_2, x_3, \dots, x_n \qquad (2)$$

   is calculated value of the same statistics

$$G_u^{'}(x_1^{'}, x_2^{'}, x_3^{'}, \dots, x_n^{'}) = \sum_{i=0}^{n} x_i^{'}, x_i^{'} \in D \qquad (3)$$

   for each of Boolean properties of genes, index u answers number of shift;
5) Procedures of shift and calculation of statistics 3) - 4) repeat U times;
6) Error probability at a deviation of a null hypothesis

$$p\_value = G\_u/G\_0 \qquad (4)$$

   is defined as a share of values Gu, exceeding $G\ 0$.

It is possible to implement this algorithm by means of data view in the form of the matrixes containing the functional description of genes in the cells. Due to exchanges between all computing devices, it is possible to organize search of all genetic combinations. At the same time evenly to divide the available basic data between all calculators that is important at a large number of genetic data. Thus, each computer will have a part of the general data set in the memory, and will receive optional data, which are required for testing (in matrix representation columns and lines) from other calculators. In tables 2 and 3 we presented input and output parameters of an algorithm of the permutable test.

TABLE I. Input Parameters of an Algorithm of The Permutable Test

Parameter	Description and Format	
	Description	*Representation format*
List of genes	Identifier list of the studied genes	Character
Property list of genes	The identifier list of properties of genes associated with a set of	Character

Parameter	Description and Format	
	Description	*Representation format*
	genes. Here functional summaries of GeneOtology act as identifiers of properties of genes	
List of characteristics of genes	The list of the measured quantitative characteristics of genes associated with a set of genes	Numerical

TABLE II. OUTPUT PARAMETERS OF AN ALGORITHM OF THE PERMUTABLE TEST

Parameter	Description and Format	
	Description	*Representation format*
List of genes	Identifier list of the studied genes	Character
Property list of genes	The identifier list of properties of genes associated with a set of genes. Here functional summaries of GeneOtology act as identifiers of properties of genes	Character
List of p-values	The list of the calculated p-values for each property of a gene in each statistical group associated with property set of genes.	Numerical

IV. TESTING

The tape algorithm of matrix multiplication as it well gives in to parallelization is applied to testing of efficiency of use of graphic processors and corresponds to logic of distributed computing. Here each calculator has only a part of data and for calculation of the part of a task; it needs exchange with other calculators. Each calculator calculates the part of elements of the first line of a matrix, which is available for him. After calculations of n machine transfers the available part of a matrix In following and accepts a part of a matrix In from the previous machine. Process repeats until each calculator does not receive from all missing parts of a matrix, and, respectively, will not transfer the part of a matrix, which is available for it to all rest. As a result, there is p of identical branches of a parallel algorithm, each of which works with the part of data.

Experiments are decided to be made by means of the API SMPI system of modeling SimGrid, which allows imitating work of RVS. SimGrid is the framework implementing modeling of the distributed high-performance AF. SimGrid can be used for assessment of abstract algorithms or describe and debug the real distributed applications. The system allows exploring the areas Grid, DataGrid, IaaS Clouds, clusters, supercomputers and peer-to-peer systems. From the technical point of view of SimGrid is library with which interaction is organized so that the user wrote the programs with the purpose to construct own simulator.

The SimGrid system provides a number of key abstractions and functionality, which can be used for creation of simulation for certain application areas and/or topology of computing environment. The system carries out the modeling managed by events. The most important component of process of modeling - modeling of a resource. It is supposed that resources have two technical characteristics: a delay (time in seconds to get access to a resource) and the rate of service (the number of units of the operations, which are carried out for unit of time). SimGrid provides mechanisms to simulate technical characteristics as constants or based on trace. It means that the delay and rate of service of each resource can be simulated by a vector of values with a time stamp or trace. Traces allow making modeling of any computing errors including as those that are inherent in real resources. In effect, trace is used to make background load of resources, which are divided on time with other users.

SMPI is one of program interfaces (API) of a system of modeling SimGrid which was created over internal API SimGrid, SIMIX. SIMIX provides access to a modeling kernel - SURF in which simulation models are implemented. SMPI supports the MPI applications written in programming language C which should be connected to SMPI library to make modeling.

Modeling of SMPI works in the one-line mode with each process of MPI working in its own flow. However, these flows work consistently under control of a kernel of modeling of SimGrid. Thus, the kernel of modeling is completely consecutive. This design solution because of the known problems of parallel discrete modeling of an event. The potential lack of a consecutive kernel – time of modeling can strongly increase with a modeling scale.

Distinctive feature and indisputable advantage of SMPI is still that it works with not modified MPI programs that allows to carry out simulation of real MPI programs on different models of high-performance AF. Therefore, development processes of balancing in the conditions of heterogeneity and modeling highly productive AF will integrate in one action.

By means of a task of computing power of each of nodes RVS productivity is established. As the studied GPU Nvidia GeForce 1080 is selected. Calculations on several video cards with use of MPI technology are made for comparison of actual capacity of the selected video card with the design capacity of RVS.

The order of carrying out experiments can be described as follows:

1) Testing of calculations on 5 GPU by means of MPI technology, with matrixes dimension: 3000x3000, 10000x10000, 15000x15000.
2) Testing of calculations for CPU by means of SimGrid, with matrixes of the same dimension.
3) Testing of the RVS model by means of SimGrid with productivity parameters, the relevant GPU and matrixes of the same dimension. Contrastive analysis of the received results and forming of the conclusion.

V. RESULTS OF EXPERIMENTS

A. Testing of calculation on GPU using MPI and OpenCL technologies.

By results of testing GPU performance data further used as reference data on actual productivity in the course of modeling by means of the GPU sets are obtained. Results are represented in table III.

TABLE III. COMPUTATION TIME AT INVOLVEMENT OF GPU AND MPI TECHNOLOGY, SEC.

Matrix size	Number of calculators	
	1 calc.	*2 calc.*
3000x3000	28.3	10.1
5000x5000	149.7	29.5
10000x10000	1202.7	126.4
15000x15000	1624.4	424.8

B. Testing of calculation on CPU.

By results of modeling it is possible to note that at increase in number of calculators acceleration of calculations is observed. At the same time the greatest coefficient of acceleration upon transition from 1 to 20 calculators, equal 46, is received at calculation with dimension of matrixes of 15000x15000 elements. Results are represented in table IV.

TABLE IV. COMPUTATION TIME WHEN MODELING CPU BY MEANS OF SIMGRID, SEC.

Matrix size	Number of calculators			
	1 calc.	*5 calc.*	*10 calc.*	*20 calc.*
3000x3000	319.7	65.9	36.4	25.3
5000x5000	1589.4	301.3	160.9	98.6
10000x10000	12150	2380.6	1229.2	686.2
15000x15000	40561.2	7995.8	1425.4	869.2

C. Testing of the RVS model by means of SimGrid software with productivity parameters, the relevant GPU.

For comparison of compliance of productivity of model with the actual productivity of GPU multiple tests were carried out and by comparison of results of real starts with results of start of model was and the required productivity of model is set.By results of modeling it is possible to note interesting feature: the maximum acceleration was reached when using 10 calculators (for matrixes by dimension of 3000x3000, 5000x5000 and 10000x10000 elements) and 5 calculators (for matrixes dimension of 15000x15000 elements). It can be explained with the fact that GPU reaches maximum capacity at the maximum loading, and at increase in number of calculators division of one "big" task into a set small considerably increases that increases the number of exchanges and calculations of small data volume. It is caused by structure of exchanges – "everyone with everyone" which is caused by directly tape algorithm. Proceeding from these results it is possible to conclude that use more than 10 calculators with GPU for a solution of a problem of multiplication of matrixes of the studied dimension is inexpedient. Results are represented in table V.

TABLE V. COMPUTATION TIME WHEN MODELING GPU BY MEANS OF SIMGRID, SEC

Matrix size	Number of calculators				
	1 calc.	*5 calc.*	*10 calc.*	*20 calc.*	*50 calc.*
3000x3000	31.6	8.8	7.9	11.1	26.7
5000x5000	146.8	36.4	27.6	31.9	65.1
10000x10000	1175	119.35	98.6	116.3	238.6
15000x15000	1569.8	389.1	511.1	529.4	538.3

In comparison with computation time on CPU, GPU show the greatest acceleration when using one or 5 calculators with GPU, at further increase in number of calculators with GPU in experiments with matrixes of the considered dimension recession of acceleration concerning CPU is observed.

VI. CONCLUSION

Within the research of efficiency of implementation of distributed computing using GPU in a solution of applied tasks the analysis of the existing technologies of involvement of graphic processors for calculations on the basis of which the decision on the choice of technology which will be used for application development for start in heterogeneous RVS was made is led and also a number of the corresponding recommendations is developed. The problem of genetic determination of genes connected with studying of a problem of heredity acts as reference and applied.

The developed parallel algorithm making multiplication of matrixes with transfer of calculations from the central processor on graphic was tested in RVS with the different number of calculators and with involvement of graphic processors. On the basis of the received results model parameters for the SimGrid system by means of which simulation of RVS with a large number of the calculators using both the central processor, and graphic was made were picked up.

The results received during the made experiments allow drawing conclusions on expediency of use of graphic processors for calculations of this look in comparison with the central processor. It is experimentally proved that the acceleration coefficient concerning CPU can both increase, and to decrease depending on specifics of a task, but in general use of graphic accelerators is justified and allows receiving gain of productivity in RVS.

The reported study was funded by RFBR according to the research project № 18-37-00360, RFBR and government of Novosibirsk region according to the research project №18-47-540006p_a.

REFERENCES

[1] M. Bernaschi, A. Di Lallo, R. Fulcoli, Emanuele Gallo, L. Timmoneri, "Combined Use of Graphics Processing Unit (GPU) and Central Processing Unit (CPU) for passive radar signal & data elaboration", Radar Symposium (IRS), 2011, IEEE, Leipzig, Germany.

[2] Kruglov V.N., Papulovskaya N.V., Chiryshev A.V. "Advantages of sharing of the CPU and GPU device", Basic researches, number 8-2 2014, pp. 296-304.

[3] Mishchenko P. V., A.V. Belyashov, "The program organization of highly productive distributed computing systems with programmable structure", the Scientific bulletin of Novosibirsk State Technical University, 2016, 2 (63), pp. 115-128.

[4] Glinsky B.M. "Siberian supercomputer center of collective use", Information science problems, vol. 3, 2013. Page 72-84.

[5] P.S. Goncharuk, A.S. Lykov, P.V. Mishchenko, "Module of increase in fault tolerance of distributed computing systems", Intellectual analysis of signals, data and knowledge: methods and means: Saturday. station Vseros. Internatianal science conference, Novosibirsk, 14-17 October 2017, Novosibirsk, NGTU publishing house, 2017, pp. 129-134.

[6] "Parallel computings of CUDA", URL: http://www.nvidia.ru/object/cuda-parallel-computing-ru.html (date of the address 30.10.2018).

[7] Boreskov A. V., "Parallel computings on GPU. Architecture and program CUDA model", Moscow, publishing house of the Moscow university, 2012, 336 pages.

[8] "The open standard for parallel programming of heterogeneous systems", URL: https://www.khronos.org/opencl/ (date of the address 28.10.2018).

[9] "General information about C AMP", URL: https://msdn.microsoft.com/ru-ru/library/hh265136.aspx?f=255&MSPPError=-2147217396 (date of the address 28.10.2018)

[10] "OpenACC", URL: https://www.openacc.org/get-started (date of the address 19.04.2018) .

[11] Yakimenko A.A., Gongbing K.V., Hayretdinov M.S., "Search of the represented characteristics of genes", experience of implementation of the permutable test with use of graphic processors, Avtometriya, vol. 1, T.50, 2014, pp. 123-129.

[12] Dematté L., "GPU computing for systems biology", Brief Bioinform. 2010, vol. 11(3), pp. 323-333.

[13] Van Hemert J.L. "Monte Carlo randomization tests for large-scale abundance datasets on the GPU", Comput Methods Programs Biomed, 2011, vol. 101(1), pp. 80-86.

[14] Shterev I.D., "permGPU: Using graphics processing units in RNA microarray association studies", BMC Bioinformatics, 2010, vol. 11. p. 329.