Use of CUDA Parallel Computing Technology in Modeling of Solid Mineral Deposits

D.I.Eryemin[1], Yu.A.Ponyatov[1], N.I.Yagfarova[1]

[1]*Institute of Space Technique and Technology, Almaty, Kazakhstan*

denis.e@bk.ru, user_for_all@mail.ru, nadiya1006@gmail.com

Abstract. *This article describes CUDA technology of high-performance parallel computing and its application in the modeling of mineral deposits.*
The article describes CUDA technology of high-performance parallel computing. It provides the concept, features of the technology architecture, the principle of development of applications based on CUDA.
The article presents algorithms for modeling mineral resources deposits using CPU and GPU, as well as an algorithm for data distribution of a deposit's block model for computing by CUDA.
The article contains analysis of the results of software for simulation mineral deposits, based on CUDA technology.
The results showed that the calculations based on CUDA technology of parallel computing, allow significant acceleration of calculating of deposit's block model and enables more detail modeling. Conducted research shows that CUDA parallel computing technology is well suited for block modeling of solid mineral deposits, allowing more efficient and cost-effective simulation.

Keywords

CUDA parallel processing technology, modeling of mineral deposits, CPU, GPU.

Introduction

Development of the mining industry led to current abundance of special software and information systems, which assist in solving issues of deposits development. However, these software products for automated modeling of deposits require significant computing time, they are expensive and do not comply with accepted methods of estimating reserves and defining categories of reserves. Significant resource to increase efficiency lies in the ability of companies to make optimal production decisions, which are mainly dependent on timely relevant information. Given that at the moment the existing system cannot provide accurate data in a short period of time, this problem can be solved by using modern information technologies.

Capabilities of new information technologies and modern electronic component base allows modeling of mineral deposits on a higher qualitative and quantitative level. In order to develop highly accurate models of mineral deposits, modern advances of information technology were considered, for example high-performance computing based on CUDA technology and cloud computing.

CUDA technology is the most promising technology of using emerging computing power, which combines software and hardware means for high performance parallel computing. Compared with the traditional approach to the organization of general-purpose computation, this technology, using capabilities of graphics API, has the following advantages in this area:

- CUDA application programming interface (CUDA API) is based on the standard C programming language;
- 16 kb of memory, shared between threads, can be used for user-established cash with higher bandwidth than when retrieving data from common textures;
- More efficient transactions between CPU memory and video memory;
- Full hardware support for integer and bitwise operations;
- Support for compiling GPU code by means of Low Level Virtual Machine (LLVM). [1].

1 Algorithm of mineral deposit modeling

Using CUDA parallel computing technology allows more efficient and cost-effective modeling of mineral deposits. To this end, software for imitation modeling of ore bodies of solid mineral deposits using polygonal method on the basis of CUDA parallel computing technology was developed.

The essence of the software implementation of deposit block model calculation is that each block is assigned properties inherent to the nearest sample to this block. The software consists of two subsystems: the I/O subsystem and the calculation subsystem. The first subsystem performs reading source data from a file, calls calculation functions and writes the result to a file. The computing subsystem performs the following actions: determines the number of blocks in the block model, distributes data among threads, allocates and initialized memory, creates threads, indicates memory volume for the thread, executes algorithm of calculating the block model of the deposit. [2]

The developed software can perform calculations using both CPU and the GPU. In both cases, the software uses multi-threaded model of execution. The main difference of a GPU and CUDA technology is the number of threads used. For example, calculations on CPU use from 2 to 16 threads, but GPU calculations may use up to 65535^3 threads. [3]. Because of that, parallel computing using GPU is preferable because the computation time can potentially be much shorter than when using CPU.

As part of the testing of the developed software sample, series of test calculations of the block model were performed. The following values were taken as the initial data:

For calculations, one profile was taken, the block model of which is represented by a two-dimensional array A, consisting of 6 million blocks (x = 3000, y = 2000);

Exploratory data is a one-dimensional array B, consisting of 600 elements with values (x, y, c), where *x* and *y* correspond to coordinates of the sample, and *c* corresponds to the commercial mineral content in the sample;

Also, to model ore bodies of solid minerals deposits using the polygonal method on the basis of CUDA parallel computing technologies, two parameters were added: the number of video cards blocks and the number of threads.

This example is a special case of the deposit model. It is used for testing and determining of existing hardware performance and verification of the calculations.

To simulate a situation close to reality it was assumed that out of 600 samples, 100 samples represent conditional commercial mineral content values from 1 to 9. These values may be interpreted in the relative values of the commercial mineral content in samples. The remaining 500 samples have commercial mineral content values equal to 0.

An example of a block model of the profile is shown in Figure 1.1. In general, one block may contain several samples. That is why the calculations use coordinates of a block geometric center as its coordinates.

A

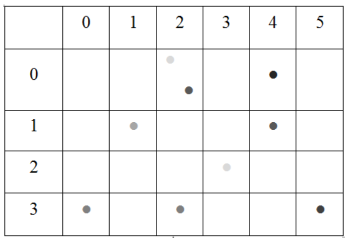

Fig. 1.1 – General view of the block mode of the profile

Exploratory data is shown on Figure 1.2.

B

(0,2,1)	(0,2,5)	(0,4,9)	(1,1,3)	(1,4,8)	(2,3,1)	(3,0,2)	(3,2,4)	(3,5,7)

Fig. 1.2 – Exploratory data

1.1 Algorithm for simulation study of ore bodies of solid minerals deposits using polygonal method on the CPU

Calculation the commercial mineral content in each block of the block model of the algorithm for simulation study of ëore bodies of solid minerals deposits using polygonal method on the CPU is shown in Figure 1.3.

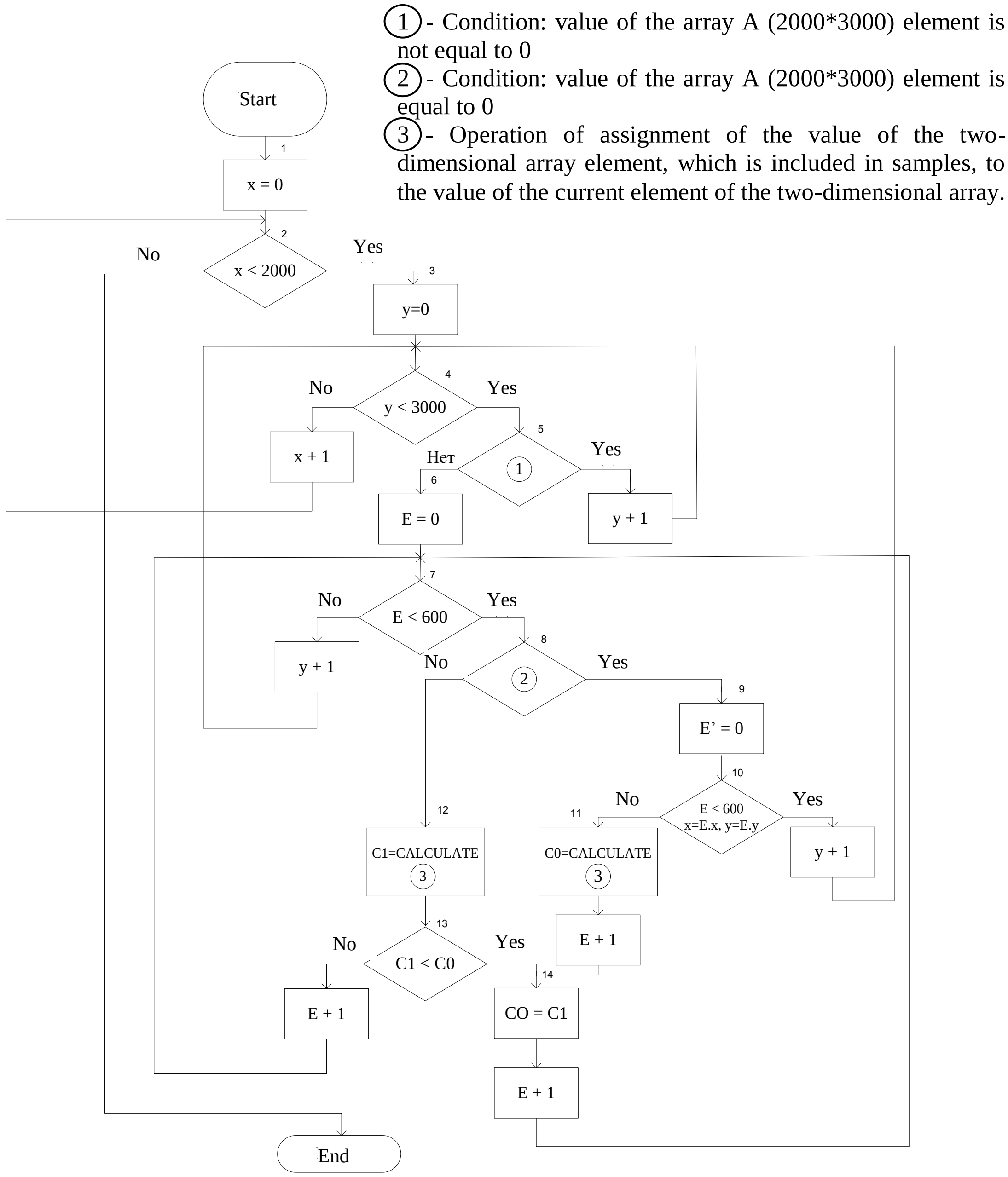

Fig. 1.3 – Algorithm for computing on CPU

In the second part of the algorithm A, procedure of actions is as follows:

1. Viewing each element (block) of two-dimensional array A (profile) by *x*;
2. If the index of an element equals to 2000, go to end. If the index is less than 2000, continue;
3. Viewing each two-dimensional array by *y*;
4. If the index of an element equals to 3000, go to 2. If the index of an element is less than 3000, go on;
5. Check the commercial mineral content in the block. If commercial mineral content is not equal to 0, return to 4. If it equals to 0, then go on;
6. Viewing each element of the array B (array of samples);
7. If the index of an element equals to 600, return to 4. If the index is less than 600, continue algorithm, going on;
8. If the value of the current two-dimensional array is 0, go to 9. Otherwise, go to 12;
9. Viewing each element of the array A;
10. If the index of the current element is less than 600, and the coordinates of this element does not belong to values of the one-dimensional array (an array of samples), go to 11. Otherwise return to 4;
11. Calculate the distance from this block of the polygon to the sample. Let us assume that the distance to the first sample taken is minimal. Assign the value of the current sample to this block of the polygon;
12. Find distances to each of the following samples;
13. Compare them with minimal assumed distance;
14. If the newly calculated distance is less than the minimal, assign it a minimal value.

Thus we are left with only the value of the sample, the distance to which the smallest. After going through all the iterations (steps 2, 4, 7), the algorithm stops.

1.2 Algorithm for simulation study of ore bodies of solid minerals deposits using polygonal method on the basis of CUDA parallel computing technology

The initial data of the algorithm for simulation study of ore bodies of solid minerals deposits using polygonal method on the basis of CUDA parallel computing technology is similar to the initial data for the algorithm using CPU, except for the addition of two parameters: the number of video card blocks and the number of threads.

The block diagram of the algorithm for calculation of the commercial mineral content in each block of the model based on the CUDA parallel computing technology is shown in Figure 1.4.

The algorithm performs the following steps:

1. Viewing each element of the array B (the array of samples);
2. If the index of an element equals to 600, go to end. If the index is less than 600, continue algorithm, going on;
3. If the current element of the two-dimensional array equals to 0, go to 4. Otherwise, go to 7;
4. Viewing each element of the array A;
5. If the index of the current element is less than 600, and the coordinates of this element does not belong to values of the one-dimensional array (an array of samples), go to 6. Otherwise return to 2
6. Calculate the distance from this block of the polygon to the sample. Let us assume that the distance to the first sample taken is minimal. Assign the value of the current sample to this block of the polygon;
7. Find distances to each of the following samples;
8. Compare them with minimal assumed distance;
9. If the newly calculated distance is less than the minimal, assign it a minimal value.

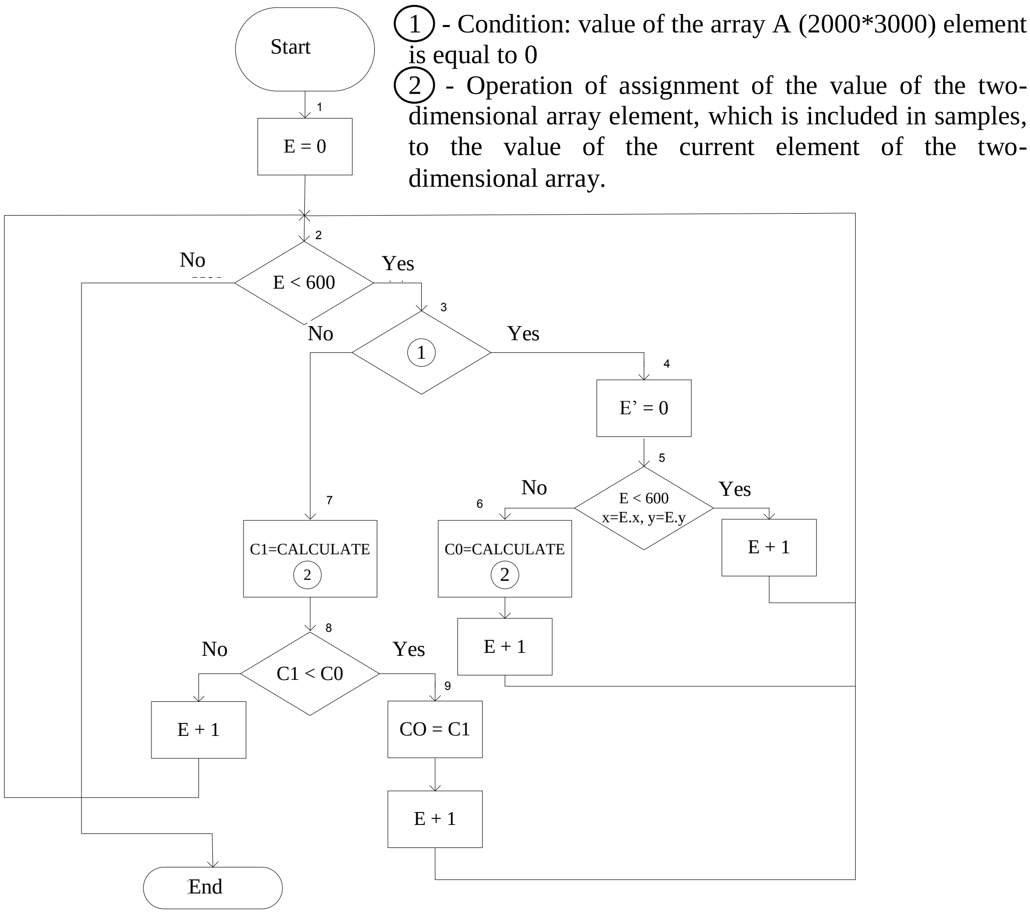

Fig. 1.4 – Algorithm for computing using CUDA technology

Thus we are left with only the value of the sample, the distance to which the smallest. After going through all the iterations the algorithm stops.

1.3 Algorithm for distribution of data in the deposit block model

Due to the large volume of data of the deposit and existing restrictions of a video card, the block model of the deposit in some cases cannot be handled in a single iteration. Based on that, it is necessary to have a queue for processing blocks of the deposit block model, which provides maximum load to the video card. To implement this possibility, algorithm of distribution of the deposit block model to several video cards by parts was developed for computing using CUDA, which consists of the following stages:

- Selection of the number and type of used video cards;
- Determination of video card characteristics of a video card (dimension of the grid, dimension of the block, i.e. the number of threads, which can be processed in one block of CUDA);
- Checking the ability to handle the volume of incoming data. If the amount of the deposit data exceeds the amount of available memory of the video card, the deposit is divided into parts, which the video card can handle, and calculations are iterative in nature;
- The number of blocks in the deposit block model is divided by the number of threads, which one CUDA block can process, thus the number of CUDA blocks, which must be calculated, is determined;
- Calculation of blocks of the deposit model using the polygonal method.

Algorithm for distribution of the deposit block model data for processing using CUDA is shown in Figure 1.5.

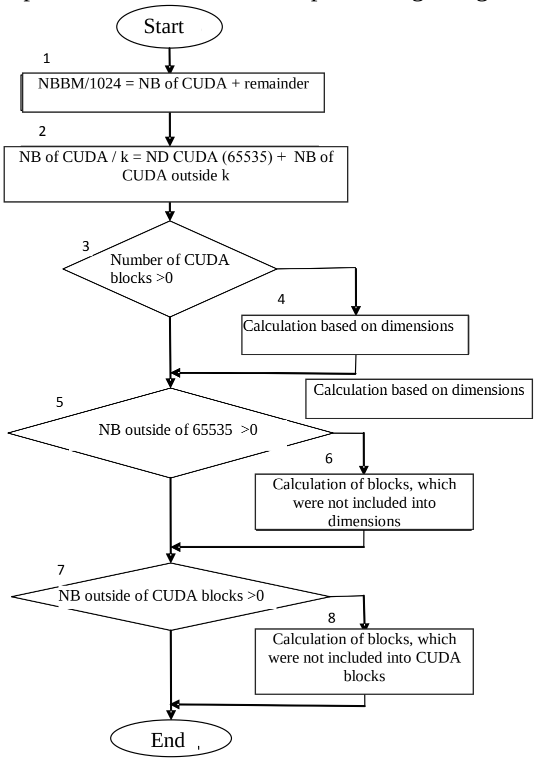

Fig.1.5 – Algorithm for allocation of the deposit block model data for calculation using CUDA

1. Divide the number of blocks of the block model (NBBM) by *n* (the maximum number of threads in a one CUDA block). We get two numbers: the number of blocks (NB) of CUDA and the number of blocks in the block model, which were not included in CUDA blocks;
2. Divide number of CUDA blocks by *k* (the maximum number of blocks that can fit into the block space of CUDA). We get the number of dimensions (ND) by *k* and the number of CUDA blocks, which were not included in dimension *k*;
3. If the number of dimensions is greater than zero, then go to 4, otherwise go to 5;
4. Perform calculations by dimensions, i.e. process those blocks that are included in this dimension;
5. If the number of blocks, which are not included into the dimensions greater than zero, go to 6, otherwise go to 7;
6. Perform calculation of blocks that are not included in the multiplicity *k*;
7. If the number of blocks that are not included in CUDA blocks is greater than zero, go to 8, otherwise go to end;
8. Perform calculation of blocks that are not included in CUDA blocks.

Thus, processing of the entire deposit is performed by forming a queue of the calls to the video accelerator. For processing, the graphics accelerator initially receives the volume of data divisible by *k*, then receives the rest of data in the range less than *k*, which are included into CUDA blocks, and the rest of data from division by *n*.

Based on the research, the following conclusions can be made:

- The developed study method allows reliable and quantified to determination of benefits of algorithms for simulation study of mineral deposits using polygonal method on CPU and based on CUDA technology of parallel computing on the GPU.
- The selected polygonal method is a good example for testing performance of hardware, software and algorithmic support the CUDA parallel computing technology during solving the problem of simulation study of ore bodies in solid mineral deposits.
- In accordance with the analysis it was defined that polygonal method in a raster representation is optimal for modeling of ore bodies in mineral deposits.

2 TESTING. ANALYSIS OF RESULTS

Within the framework of testing softare for simulation study of ore bodies in solid mineral deposits using the polygonal method, a series of experimental calculations of a deposit block model were perfomed according to the structure on the CPU and GPU. The most representative results are given below.

2.1 Tests on CPU

Generation of a block model of the deposit with four threads on the CPU.

Calculation time of each variant of generation of the block model of the deposit is shown in Figure 2.1.

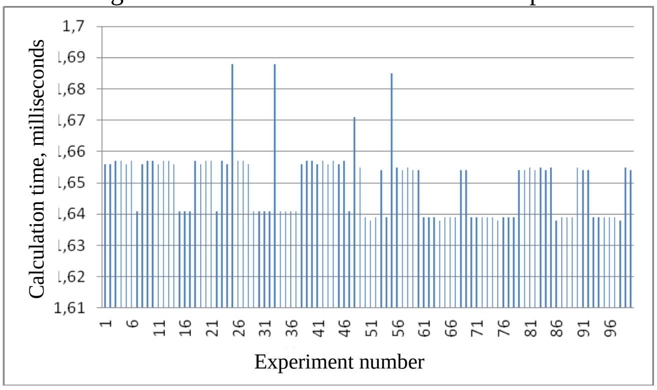

Fig. 2.1 – time of calculation of each variant in the series of generation of the deposit block model, which is comuted usingthe algorith with four threads on the CPU

The function of distribution of the number of variants in the series to the time intervals is shown in Figure 2.2.

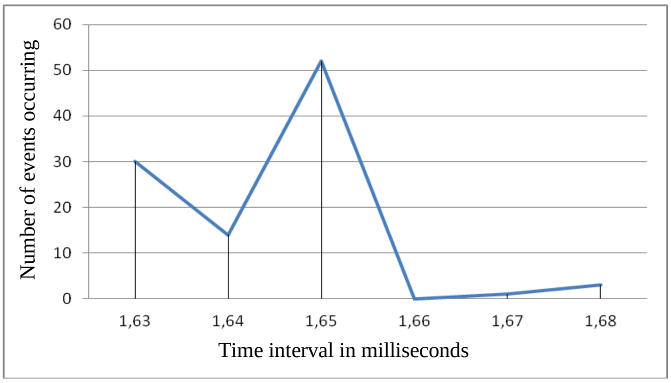

Fig. 2.2 – Function of distribution of the number of variants in the series to the time intervals when generating a block model of the deposit, calculated using the algorithm with four threads on CPU

2.2 Experiments on GPU

Generation of the block model of the deposit using the algorithm with three Tesla C2050 GPU computing processors.

Calculation time of each variant of the deposit block model is shown in Figure 2.3.

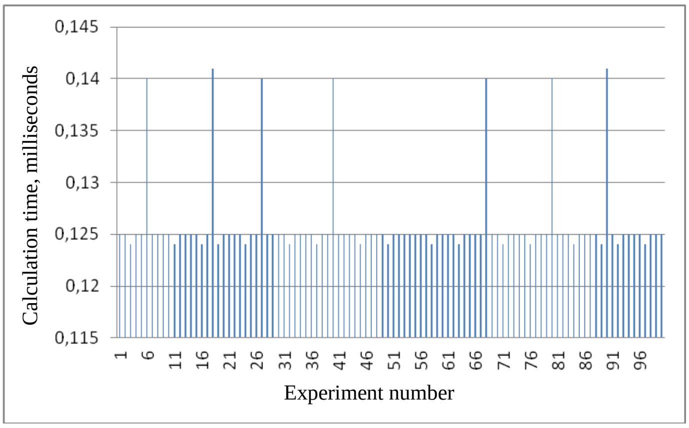

Fig. 2.3 –Time of feneration of each variant of the deposit block model, calculated using three Tesla C2050 GPU computing processors

Function of distribution of the number of variants in the series to time intervals is shown in Figure 2.4.

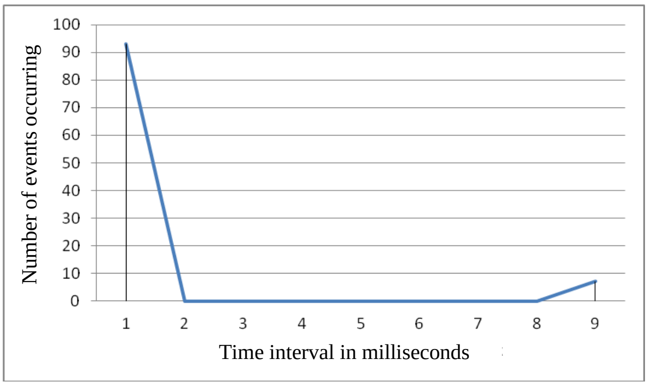

Fig. 2.4 Function of distribution of the number of variants in the series to time intervals of generation of the block model of the deposit, calculated on three Tesla C2050 GPU computing processors

2.3 Analysis of results

Average processing time for each experiment is shown in Table 1.

Table 1 – Average processing time of each of the calculations s

Processing unit		**Number of threads/devices**	**Average time of processing of one variant in the series, ms**
CPU		1	8.76501
		4	1.64967
		8	1.68113
GPU	GeForce	1	0.8135
	Tesla	1	0.32292
		2	0.17815
		3	0.1259

Minimum time of calculation:
- On CPU = 1.64967 ms (algorithm with four threads per CPU);
- On GPU = 0.1259 ms (algorithm with three Tesla C2050 GPU computing processors).

During the study, it was determined that modeling of ore bodies in mineral deposits using polygonal method on the basis of CUDA technology, unlike calculations performed by the central processor unit of a computer, provides better performance of calculations, speed and thus enables more detailed models of deposits.

Conclusion

Based on experiments it was found that modeling of mineral deposits using CUDA technology does not require expensive graphic accelerators, such as the Tesla C2050, since using several (relatively inexpensive) GeForce graphic accelerators provide computation time comparable with the time of calculations using Tesla graphic accelerators. The computing system can be build on the basis of GeForce graphic acclerators, whih are more than 4 times cheaper than Tesla C2050 GPU computing processor and brings considerable savings during implementation of the project. In addition, it was found that adding devices leads to an almost linear increase in performance, which confirms correctness of the chosen area of research in modeling deposits using CUDA parallee computing technology.

References

[1] Vladimir Frolov. Introduction to CUDA. Computer Graphics and Multimedia web magazine. Issue №6(1)/2008. http://cgm.computergraphics.ru/issues/issue16/cuda.
[2] Chris Kasperski. Technique optimization programs. Efficient use of memory. Il.CUDA Programming Guide 1.1: 464 p, St. Petersburg.: BHV-Petersburg, 2003.
[3] Katharina Henneböhl, Marius Appel, Edzer Pebesma. Spatial interpolation in massively parallel computing environments. – Münster 2011.