P045

Hybrid Technologies For Computation Of Enhanced Oil Recovery Problem Using Mobile Devices

D.Zh. Akhmed-Zaki (al-Farabi Kazakh National University), B.S. Daribayev* (al-Farabi Kazakh National University), D.V. Lebedev (al-Farabi Kazakh National University), T.S. Imankulov (al-Farabi Kazakh National University)

Summary

Recently heterogeneous computational systems consisting of supercomputers, FPGA, mobile devices in a state of active evolution. Problems related to enhanced oil recovery among most computational intensive ones. Given paper considers stages of hybrid parallel algorithm development for solving three-dimensional problem of the oil displacement by the method of polymer injection into oil reservoir and stages of creation of system of distributed computations on heterogeneous computational resources using mobile device. System based on using mobile device for input of computational parameters and obtaining data from sensors located directly at production field, their preprocessing using FPGA and transferring through long range and energy efficient wireless communication channels onto mobile device. After determination of computational characteristics mobile device allows to perform computation on remote heterogeneous computational resources which allows to considerably reduce computation time.

System has ability to connect to computational clusters and Grids as well as enterprise cloud services consisting of GPU- and FPGA-based computers.

Implemented parallel algorithms allow to conduct computation on CPUs. Where there are coprocessors (GPU, KNL) available system automatically determine their computational capabilities and distributes computational tasks among them.

If remote high-performance resources not available computations cam be conducted on a local mobile device. There high-performance mobile devices (Xiamoi MiPad, nVidia Shield) which allow to implement parallel algorithms using CUDA technology. Computation results displayed directly on mobile device.

Proposed technology of computation of enhanced oil recovery models allows to conduct more accurate computations and perform them directly near production field which provides quicker response to changes in field condition.

Introduction

Currently due to increased performance of supercomputers and their heterogeneity issues related to their efficient use of resources when dealing with a complex task are common. On the other side performance of mobile devices becoming closer to their desktop counterparts. Given work considers the development of hybrid parallel algorithm using all available resources of the device (CPU, GPU, Intel MIC) for solving one the most resource consuming problems of oil displacement by polymer injection into an oil reservoir for three-dimensional case. Stages of development of the information system for conducting distributed computations for oil production related problems using heterogeneous computational resources or mobile device considered.

The offered system implements mechanism allowing to obtain data from oil production field using a mobile device with wireless data channels. This data can be preprocessed using FPGA. After collection of a data, the system allows conducting computations either on a heterogeneous computational server or using a mobile device. Figure 1 shows the process of Nvidia processors development. One can see, that from 2014 high-performance architectures for mobile devices and GPUs united and this allows to conduct computations directly on a mobile device using GPU. In case computation executed on a mobile and if it supports CUDA technology computations will be on the GPU side which also allows reducing computation time (Virtejanu and Nitu 2013). When conducting computations on remote heterogeneous computational resources implemented parallel algorithm allows automatically distribute load between CPU, GPU and KNL if available.

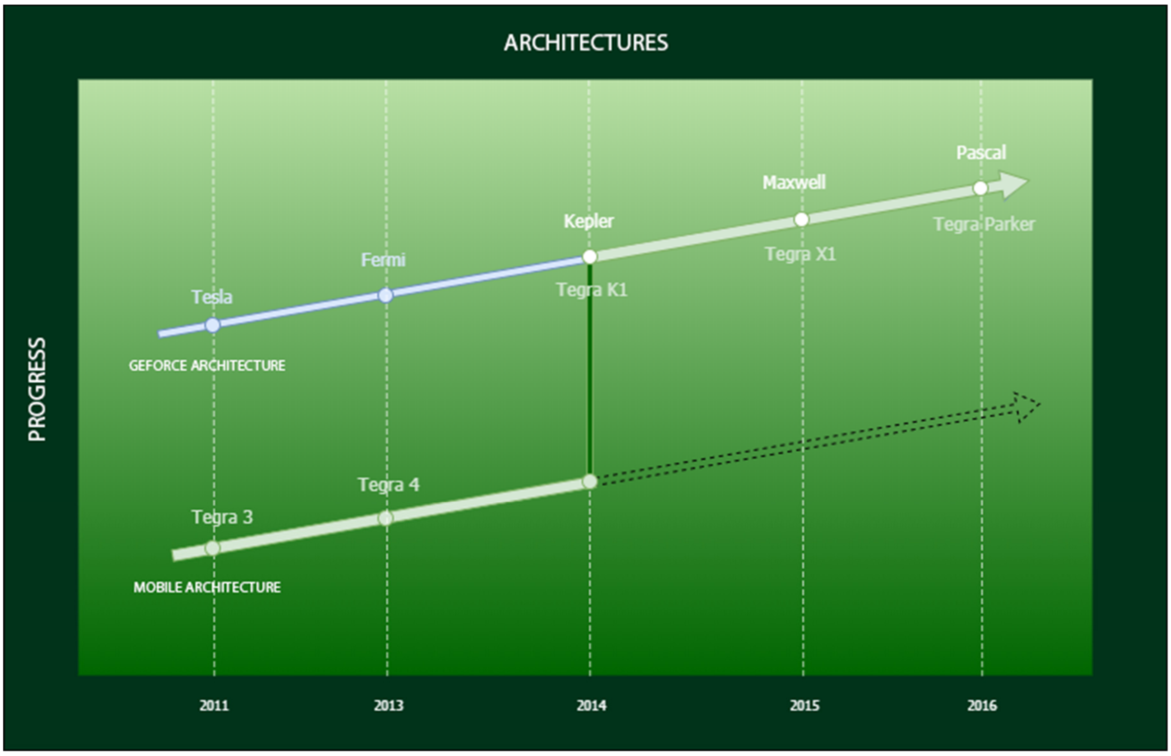

Figure 1 *Nvidia processor architectures development process.*

Thus, heterogeneous computing technologies can help make timely and efficient decisions during field development and exploitation. Moreover, they allow to collect and analyze data in real time using most recent software solutions which guarantees enhanced control over whole process of petroleum field operation.

Mathematical model

In order to test hybrid system based on mobile app three-dimensional model of enhanced oil recovery chosen. The mathematical model of a three-dimensional problem of oil displacement by polymer injection into oil reservoir contains following equations (Babalyan *et al.* 1983; Aziz and Settari. 1979):

$$m\frac{\partial S_w}{\partial t} + div(\vec{v}_w) = 0 \quad (1)$$
$$m\frac{\partial S_o}{\partial t} + div(\vec{v}_o) = 0 \quad (2)$$

$$S_w + S_o = 1$$

where m – porosity of reservoir, S_w, S_o – saturations of water and oil phase, $\vec{v}_w, \vec{v}_o$ – velocities, which is expressed by the following Darcy's law:

$$\vec{v}_i = -K_0\frac{f_i(s)}{\mu_i}\nabla P, \;\; i = w, o \quad (3)$$

$f_i(s), \mu_i$ – relative phase permeabilities and viscosities, K_0 – absolute permeability.

The mass transfer of the polymer in the reservoir during the displacement of oil proceeds through a complex mechanism, since the polymer participates in several parallel processes: the distribution between oil and water, the adsorption from both phases on the surface of the rock from water and oil. The equations for the distribution of the polymer concentration, salt transfer, and the heat transfer equation are described in (Lake 1989; Sorbie 2014).

The relative permeability curves are saturation functions and are taken as follows:

$$f_w(S_w) = S_w^{3.5}; \quad (4)$$
$$f_o(S_w) = (1 - S_w)^{3.5} \quad (5)$$

Thus, it is required to find the pressure, water saturation, flow velocity, polymer concentration, salt concentration and temperature, with initial and boundary conditions:

$$s|_{t=0} = s_0(x), \;\; c|_{t=0} = c_0(x), \;\; a|_{t=0} = a_0(x), \quad (6)$$
$$\left.\frac{\partial s}{\partial n}\right|_{\partial\Omega} = 0; \;\; \left.-D\frac{\partial c}{\partial\Omega} + \vec{v}_{1n}\cdot c\right|_{\partial\Omega} = q_n\cdot\tilde{c}; \;\; \left.\frac{\partial P}{\partial n}\right|_{\partial\Omega} = \gamma\cdot V_p; \left.\frac{\partial T}{\partial n}\right|_{\partial\Omega} = \gamma\cdot V_t \quad (7)$$

where $\partial\Omega$ - boundary of the region. For the numerical solution of problem (1) - (7) we use Jacobi's explicit iterative method (Samarskij 1983).

Architecture

In order to organize parallel computations on several computational devices the architecture of a heterogeneous system developed. This system consists of two parts: the mobile device and computational server. The App for a mobile device developed which manages entire system and distributes computational data between them. The general scheme of operation of developed mobile app shown in Figure 2. Created scheme suitable not only for oil displacement problems but can be applied to a number of problem types related to the development of oil fields. The main advantage of the developed app is that it can be used at any device supporting Android operating system.

The mobile app has to operation modes: network and autonomous. In network mode, mobile app runs user application remotely on a server and after some of the user applications complete their results stored in the server database. At that moment, the mobile app remotely calls web-service which returns calculated data. The advantage of such mode is in faster computation time thanks to the computational power of remote server and hybrid parallel algorithms of a problem. The disadvantage is that given mode requires a constant connection to a remote computational server which sometimes can be impossible to maintain because of lack of connectivity especially if one tries to perform computations at a petroleum field.

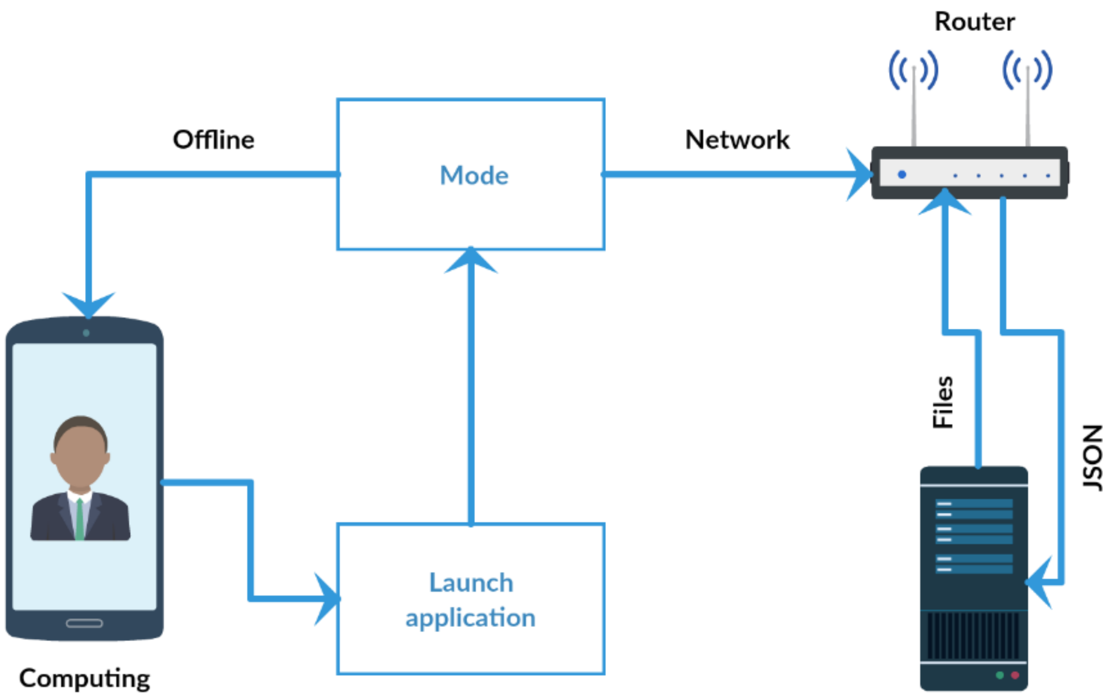

Figure 2 *General scheme of the system operation under mobile app management.*

In offline mode computations executed directly on a mobile device. If the mobile device supports the CUDA technology parallel implementation of a problem executed involving embedded device GPU. If there no CUDA support problem computes using the sequential algorithm. There is a number of devices with Nvidia Tegra K1 processor which support CUDA technology. The main advantage here is that there no need to have any kind of network connectivity. Also using CUDA technology on a mobile device on can achieve considerable performance comparable to desktop solutions of given problem. The disadvantage here is that when using offline mode, it is impossible to run tasks with a large number of computations without applying CUDA (see Table 1).

Device	**TFLOPS**	**GB/sec.**	**Memory (GB)**	**# cores**
Nvidia Tesla K20	3,52	208	5	2496
Intel Xeon Phi 7120A	2,42	352	16	61
Nvidia Tegra K1 (GPU)	0,36	17	4	192

Table 1 *Device characteristics.*

These modes switch automatically depending on server availability on which server side of the program installed. Also for the considered problem of enhanced oil recovery, there is a new implementation under development using OpenCL (David *et al.* 2015) which will make this implementation to be executed on any mobile device.

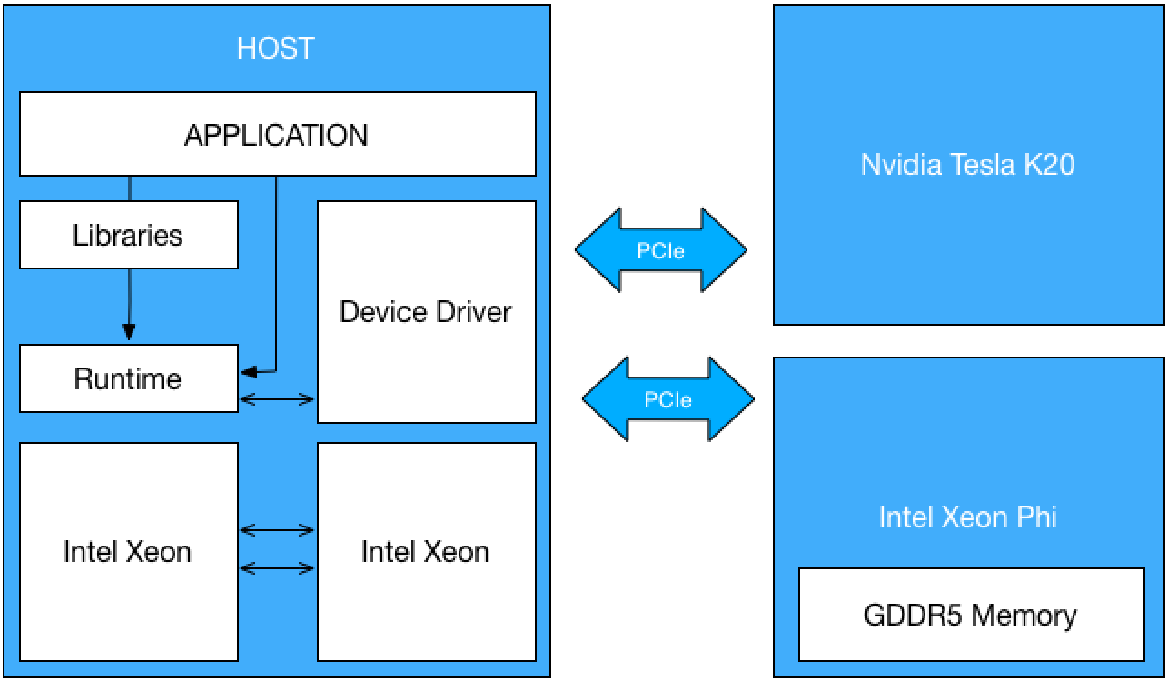

Figure 3 *Architecture of heterogeneous computational system: Intel Xeon+Nvidia Tesla+Intel Xeon Phi.*

Parallel implementation

Two parallel algorithms were implemented: one for mobile device and one for a computational server. We develop parallel algorithms using CUDA technology to be executed on mobile devices and these algorithms already published by us in (Akhmed-Zaki *et al.* 2017; Imankulov *et al.* 2016). On the server side (which represents heterogeneous computational server) we developed the hybrid computational system for numerically solving the problem of oil displacement by polymer injection into an oil reservoir. The heterogeneous computational server contains 4 Intel Xeon CPU with 8 cores each, the GPU with Nvidia Tesla K20 architecture and Intel Xeon Phi coprocessor based on Intel MIC architecture (Yang *et al.* 2017; Huang *et al.* 2017; Govett *et al.* 2017). The architecture of heterogeneous computational server shown in Figure 3. This architecture corresponds to different types of parallel interaction between computational nodes: the distributed memory for interaction between nodes, shared memory for main interaction, parallel SIMD (single instruction with multiple data) within processors and GPUs. The problem is in development of parallel algorithms allowing efficiently use different levels of parallelism, particularly MPI, OpenMP and CUDA which we use in the paper.

In hybrid computational system program automatically defines the presence of necessary resources. If all necessary resources available the MPI efficiently distributes necessary parts of a computational domain between nodes of Nvidia Tesla K20, Intel Xeon Phi, and Intel Xeon processors. Alternatively, the system distributes computational data on a number of existing resources. Scheme of the computational process on the heterogeneous computational server shown in Figure 4.

As the result, the universal hybrid parallel algorithm for heterogeneous computational nodes developed. The advantage of this algorithm is in automatic resource allocation and efficient distribution of computational data between existing resources.

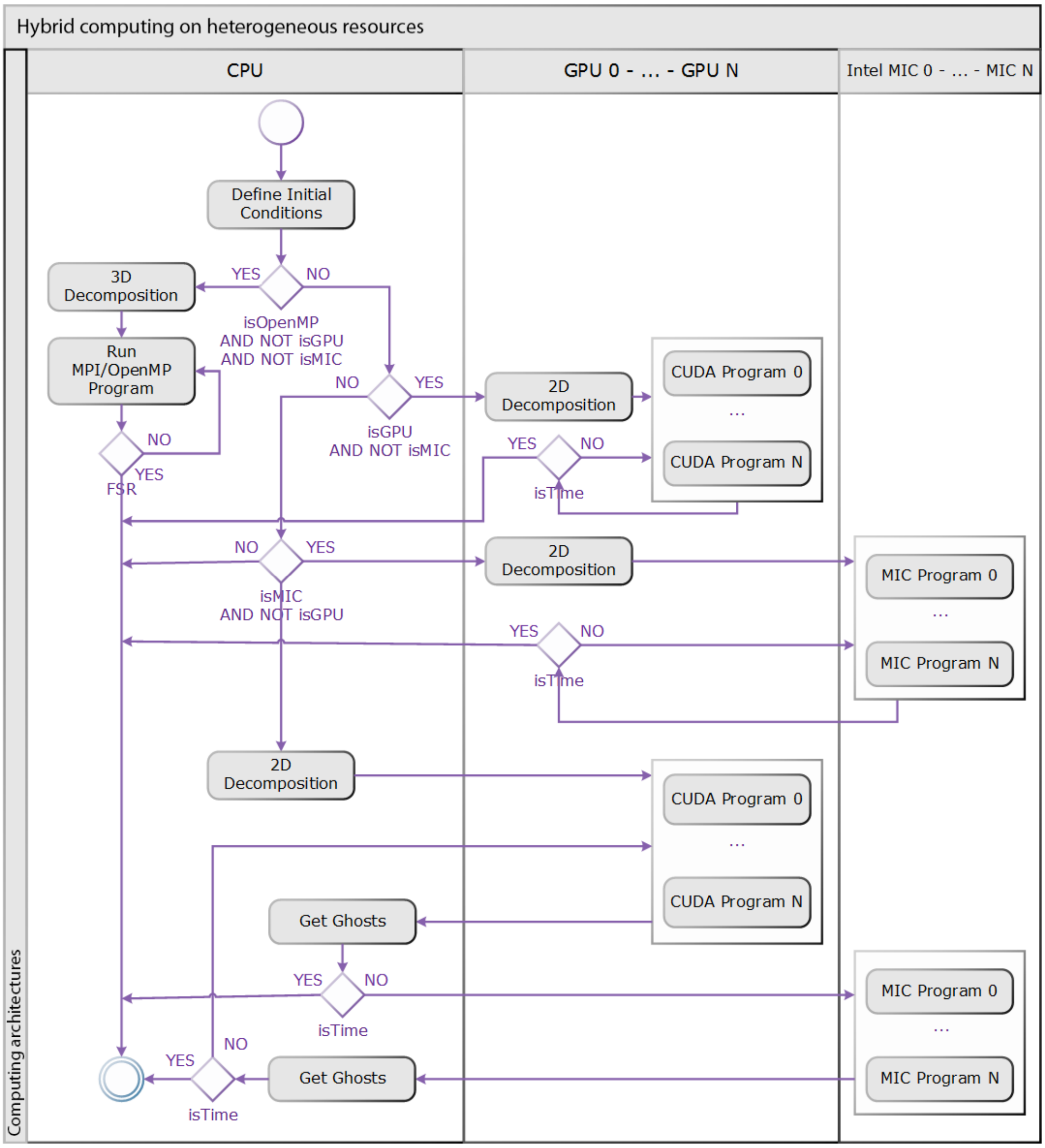

Figure 4 *Computational algorithm on the heterogeneous system (CPU + GPU + Intel MIC).*

Results

Let's assume that NX, NY and NZ are the number of computational nodes in x, y and z directions accordingly in a computational domain (Figure 5). Three-dimensional domain with NXxNYxNZ size will be represented as the NXxNYxNZ size one-dimensional array and stored in general memory. On the GPU side same, one-dimensional data representation used in a global memory. Initial data distribution on a device conducted only once at the beginning of computation using MPI. On the GPU side data stored in the shared memory of the GPU. Computational threads from each block copied from global memory into shared memory. In this case, computations conducted on threads using data from shared memory and computational results written back into global memory just before computational core complete its job. Arithmetic operations within core must be considerably large in order to compensate additional expenses on data copying which leads to efficient usage of the shared memory. One of the methods to achieve that is to increase the size of a block. When using shared memory, one has to consider the restriction of the size of shared memory which currently equals 16 Kb.

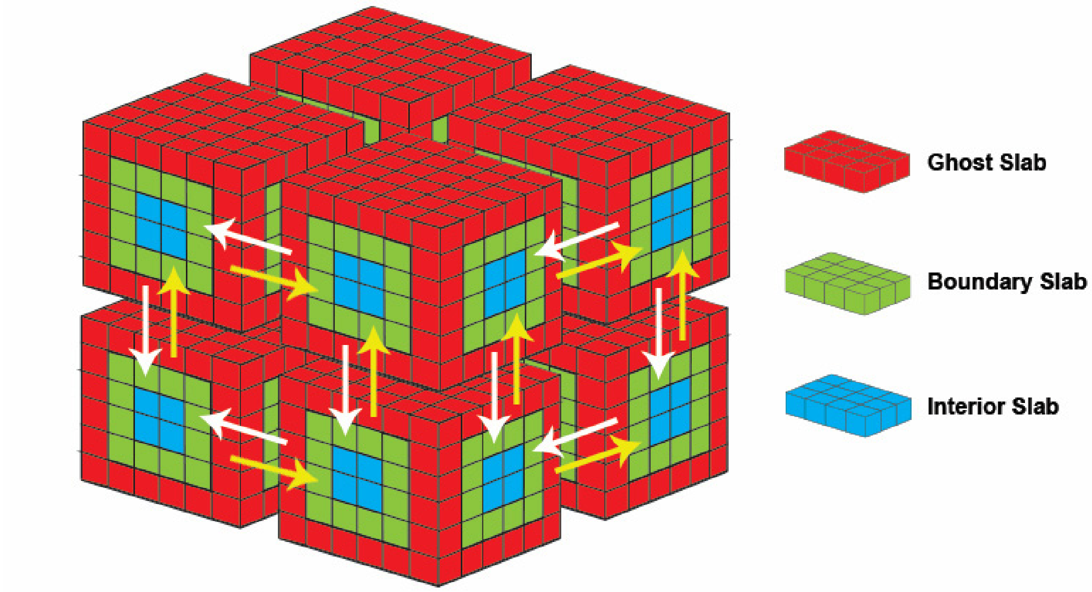

Figure 5 *Three-dimensional computation domain decomposition.*

Problems were tested on two mobile devices: Xiaomi MiPad and Nvidia Shield. Desktop-based computations conducted using PC equipped with GeForce GTX 770 graphics card in order to compare with results of computation on a mobile device. The test has shown that Nvidia Tegra K1 mobile graphics processor can compute given problem as quick as a desktop graphics card. If we take 64x64x64 grid computation time on the mobile graphics processor less than one achieved with GeForce GTX 770. Undoubtedly Tesla K20 graphics card achieved much better timings because of the sheer amount of the computational power. At the size of a grid 256x256x256 mobile device graphics card could not run the problem because it has only 1 Gb of memory and data could not fit into memory at array initialization stage (Table 2).

Grid size	Required memory (Gb)
32x32x32	~0.005
64x64x64	~0.042
128x128x128	~0.344
256x256x256	~2.750

Table 2 *Approximate amount of memory when initializing array within the program for different grid sizes excluding arithmetic operations.*

Analyzing computation times on different grid sizes one can notice that there is not so large gap in performance mobile device equipped with Tegra K1 and Tesla K20 graphics card. This means that complex computation problems of fluid dynamics can be conducted everywhere where there is a mobile device with CUDA technology support. The number of computation runs for the problem of oil displacement allowed to obtain different computation times for various combinations, namely: CUDA (Nvidia Tesla K20), MPI+OpenMP (Intel Xeon Phi) and CUDA+MPI+OpenMP (hybrid computational server). Results of computations on different grids shown in Figure 6.

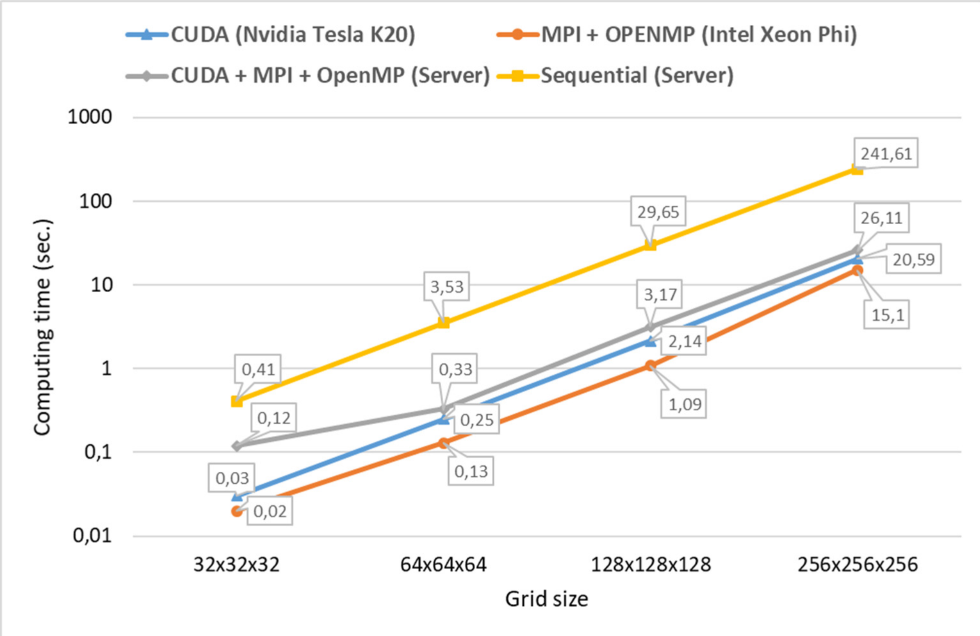

Figure 6 *Computation results of different parallel algorithms.*

Conclusion

Results achieved during preparing given paper are in development of the high-performance algorithm for solving three-dimensional filtration problem using mobile platforms with CUDA technology. The number of testing results allows making the conclusion about the possibility of using mobile devices for conducting high-performance computations. There is the parallel algorithm for numerical solution of three-dimensional filtration problem on heterogeneous computational resources created. Computational resources can be of various nature: GPUs, GPUs, and coprocessors with Intel MIC architecture. As a result of parallel algorithms testing one can notice that on heterogeneous computational resources computations and data efficiently distributed only when appropriate load balancing algorithm applied. Proposed technology of enhanced oil recovery models computation allows performing more accurate computations and conduct them located straight on a production field which allows operatively react to change of field conditions.

Further work

Further work is to implement capabilities related to usage of resources of several heterogeneous mobile devices (McClure *et al.* 2014) and computational nodes consolidating CPUs, GPUs, and coprocessors in order to conduct full heterogeneous computation.

Acknowledgements

This work was performed as part of the grant funding Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan on the topic “Development of a distributed high-performance information system for analysis of oil and gas fields within the “i-fields” concept”.

The authors are grateful to the al-Farabi Kazakh National University and the Scientific Research Institute of Mathematics and Mechanics.

References

Akhmed-Zaki D.Zh., Daribayev B.S., Imankulov T.S., Turar O.N. [2017] High-performance computing of oil recovery problem on a mobile platform using CUDA technology. Eurasian Journal of mathematical and computer applications, 5 (2), 4-13.

Aziz, K., Settari, A. [1979] Petroleum reservoir simulation. 476.

Babalyan, G.A., Levy, B.I., Tumasyan, A.B. and Khalimov, E.M. [1983] Oilfield development using surfactants. Nedra, Moscow.

David R. Kaeli, Perhaad Mistry, Dana Schaa, Dong Ping Zhang [2015] Heterogeneous Computing with OpenCL 2.0. Third Edition. Published by Elsevier Inc., 311.

Duane Storti, Mete Yurtoglu. [2015] CUDA for Engineers: An Introduction to High-Performance Parallel Computing. Addison-Wesley Professional, 352.

Govett, M., Rosinski, J., Middlecoff, J., Henderson, T., Lee, J., MacDonald, A., Wang, N., Madden, P., Schramm, J., Duarte, A. [2017] Parallelization and performance of the NIM weather model on CPU, GPU, and MIC processors. BULLETIN OF THE AMERICAN METEOROLOGICAL SOCIETY, 98(10), 2201-2214.

Huang, M.Q., Lai, C.G., Shi, X., Hao, Z.J., You, H.H. [2017] Study of parallel programming models on computer clusters with Intel MIC coprocessors. INTERNATIONAL JOURNAL OF HIGH PERFORMANCE COMPUTING APPLICATIONS, 31(4), 303-315.

Imankulov T.S., Akhmed-Zaki D.Zh., Daribayev B.S. and Turar. O.N. [2016] HPC Mobile Platform for Solving Oil Recovery Problem. Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2016), 595-598.

Lake, L. [1989] Enhanced oil recovery. 550.

McClure, J.E., Prins, J.F. and Miller, C.T. [2014] A novel heterogeneous algorithm to simulate multiphase flow in porous media on multicore CPU-GPU systems. COMPUTER PHYSICS COMMUNICATIONS, 185(7), 1865-1874.

Sorbie, K. [2014] Polymer-improved oil recovery. 359.

Samarskij, A. [1983] Teoria rasnostnyh shem. 616.

Yang, C.T., Huang, C.W., Chen, S.T. [2017] Improvement of workload balancing using parallel loop self-scheduling on Intel Xeon Phi. JOURNAL OF SUPERCOMPUTING, 73(11), 4981-5005.

Virtejanu, I. and Nitu, C. [2013] Programming Distributed Application for Mobile Platforms using MPI. U.P.B. Sci. Bull., 75(4), 143-148.