Analysis of the efficiency PETSc and PETIGA libraries in solving the problem of crystal growth

Ilya Starodumov[1], Evgeny Pavlyuk[2], Leonid Klyuev[3], Maxim Kovalenko[4], and Anton Medyankin[1]

[1] Laboratory of Multi-scale Mathematical Modelling, Ural Federal University, 620075 Ekaterinburg, Russia, ilya.starodumov@urfu.ru
[2] Department of Computational Mathematics, Ural Federal University, 620075 Ekaterinburg, Russia, evgeny.pavluk@urfu.ru
[3] Immers Ltd., Michurinskiy Ave, 19, bld. 3, 119192, Moscow, Russia, www.immers.ru, l.klyuev@immers.ru
[4] The Program Systems Institute of RAS, 152020 Pereslavl-Zalessky, Russia, kovalenko@botik.ru

Abstract. We present an analysis of high performance computational method for solving the problem of crystal grows. The method uses PETSc and PETIGA C-language based libraries and supports parallel computing. The evolution of calculation process was studied in a series of special computations are obtained on innovative cluster Immers. The results of research confirm the high efficiency of the proposed algorithm on multicore computer systems and allow us to recommend the use of PETSc and PETIGA for solving high order differential equations.

Keywords: PETSc, PETIGA, parallel, high perfomance, Immers, crystal grows, isogeometric analysis

1 Introduction

Applications involving differential operators of order greater than two have not historically lent themselves well to finite element analysis[3]. The variational statements of such problems involve second derivatives, necessitating the use of a globally C^1- continuous basis. The difficulty in constructing general setting bases has relegated the study of such equations to the subject of finite-differences and spectral methods, both of which are viable methods, but far more limited than FEA in their scope and flexibility. With isogeometric analysis, we have a higher-order accurate, robust method with tremendous geometric flexibility and compactly supported basis functions, all while maintaining the possibility of higher-order continuity. Thus, it is a convenient technology for the study of equations involving higher-order differential operators[8].

1.1 Phase field models

Two different approaches have been used to describe phase transition phenomena: sharp interface models and phase-field(diffuse-interface) models[7]. Tradi-

tionally, for describing the evolution of interfaces, such as the liquidsolid interface, has been used sharp-interface models. Such an approach requires the resolution of a moving boundary problem, separate differential equations hold in each phase, and certain quantities may suffer jump discontinuities across the interface. Phase-field models provide an alternative description for phase-transition phenomena by approximating the interface as being diffuse such that it does not need to be tracked explicitly.An alternative description for phase-transition phenomena was provided by phase-field models. Such models can be derived from classical irreversible thermodynamics. Developed by K. R. Elder et al. as recently as 2002[5, 4] the PFC model, which shares many features with the CDFT (the classical density functional theory) of freezing, was presented as an extension of the PF models to study processes with smaller length scales.Essential progress has been made in the simulation of the parabolic PFC-equation[9–11], special efforts are required to solve numerically the modified (hyperbolic) PFC-equation due to the second-order time derivative of the equation. One of the challenges to PFC has been modeling different close-packed crystal structures[6]. Such a task in three-dimensional case will be considered in the current article further.

1.2 The modified phase field crystal problem

The modified phase field crystal model describes a continuous atomic density field $\phi(x,t)$ and it is expressed by the sixth order in space and second order in time equation:

$$\tau\frac{\partial^2\phi}{\partial t^2}+\frac{\partial\phi}{\partial t}=\nabla^2\mu, \tag{1}$$

where t is the time, τ is the relaxation time of the atomic flux to its stationary state, and μ represents the chemical potential, which may be obtained from the free-energy functional

$$\mathcal{F}[\phi,\nabla\phi,\nabla^2\phi]=\int_\Omega\left[f(\phi)-|\nabla\phi|^2+\frac{1}{2}(\nabla^2\phi)^2\right]\mathrm{d}\Omega, \tag{2}$$

associated to the domain Ω. The chemical potential is simply the variational derivative of the free-energy functional $\mathcal{F}$, namely

$$\mu(\phi)=\frac{\delta\mathcal{F}}{\delta\phi}=f'(\phi)+2\nabla^2\phi+\nabla^4\phi. \tag{3}$$

The function f represents the homogeneous part of the free energy density, and takes on the form

$$f(\phi)=\frac{1-\epsilon}{2}\phi^2+\frac{\alpha}{3}\phi^3+\frac{1}{4}\phi^4. \tag{4}$$

Here, $\epsilon=(T_c-T)/T_c$ is the undercooling, where T and T_c are the temperature and critical temperature of transition, respectively. α is a coefficient which is a measure of metastability.

2 Computational experiments

The modified PFC equation is a hyperbolic differential equation of the sixth order. The solution of this equation from the computational point of view is not an easy task. Therefore, we developed special numerical algorithm using a C-language code based on the PETIGA library[2]. This software can be represented as an extension of PETSc[1] that adds the utilization of IGA capability. PETSc is a suite of data structures and routines that provide frames to develop large-scale application codes on parallel computers and consists of parallel linear and nonlinear equation solvers and time integrators[12]. Therefore we created a program with abilities for the use on high perfomance computational cluster. To assess the performance of the computational program, we developed a series of experimental tasks. The main task was to investigate the efficiency of the algorithm parallelization. Thus, we made 3 types of experiments on homogeneous HPC cluster Immers. The cluster configuration includes 5 computational nodes connected by Infiniband QDR network of 40 GB/sec. Each compute node consists of two 14-core processor Intel E5-2697 v3 and 64GB of DDR4 RAM. In all experiments, the program calculated the task for 1 time step.

2.1 First experiment

The purpose of the first experiment is evaluation of dependence the complexity of the problem from the amount of finite elements. Fixed parameters for this experiment are: computational domain size 160x160x160 and using of 5 nodes including 28 processor cores on each node. Variable parameter is a grid size: 10x10x10, 20x20x20, 40x40x40, 80x80x80, 160x160x160. Results of the experiment are presented in the following figures:

a) Growth of the size of the problem and calculation time

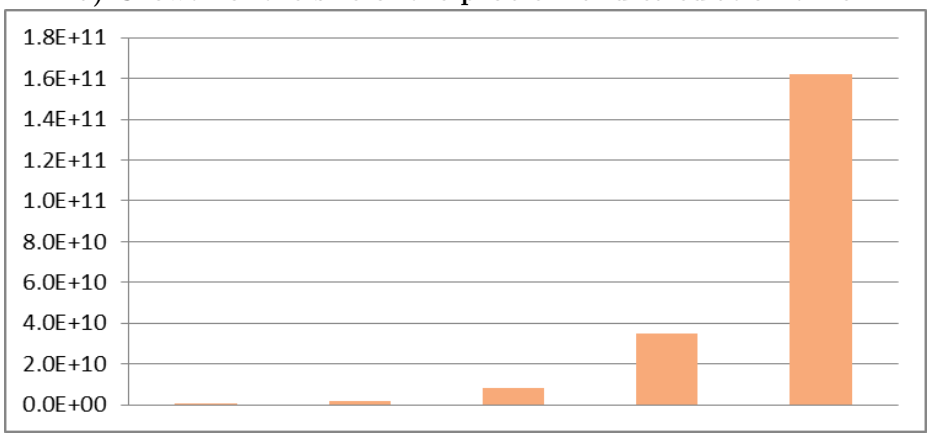

Fig. 1: Total memory used, bytes.

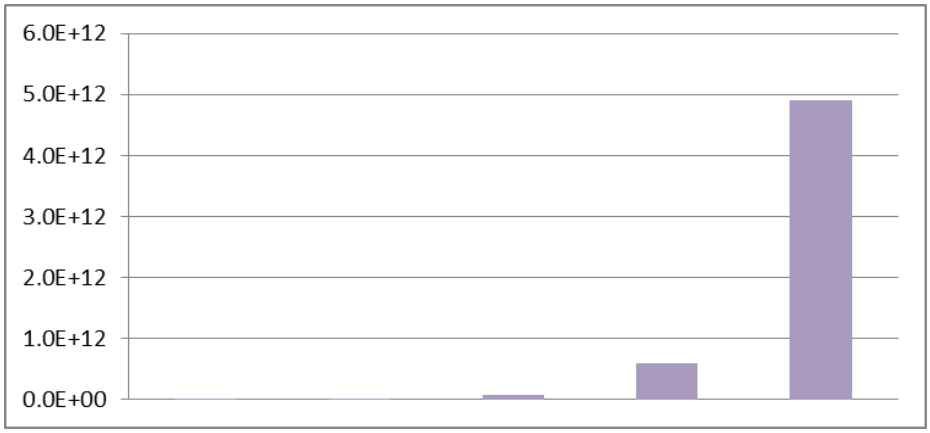

Fig. 2: Total flops.

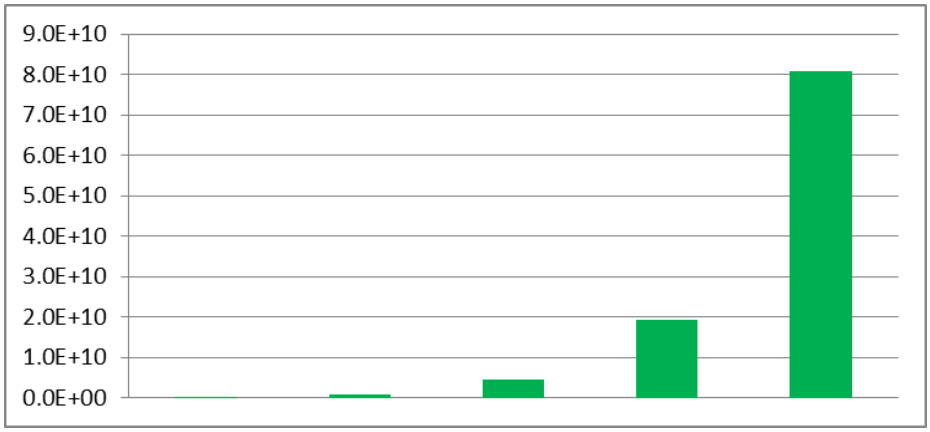

Fig. 3: MPI message total lengths , bytes.

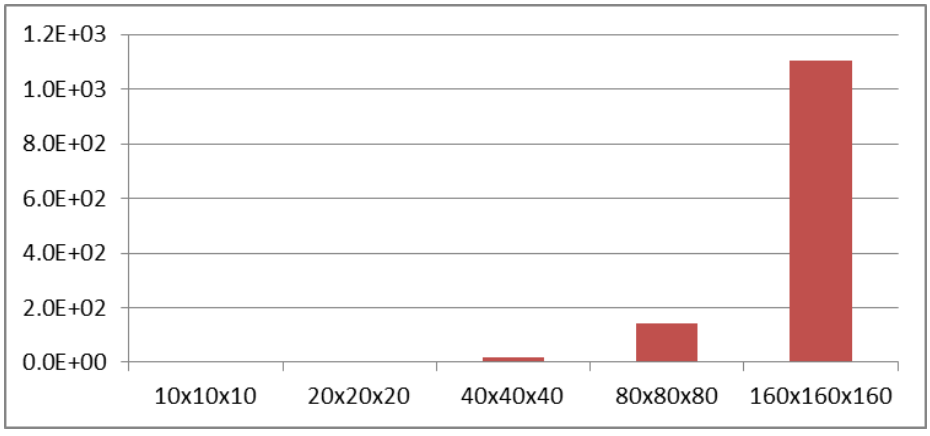

Fig. 4: Maximum computational time for a single core, sec.

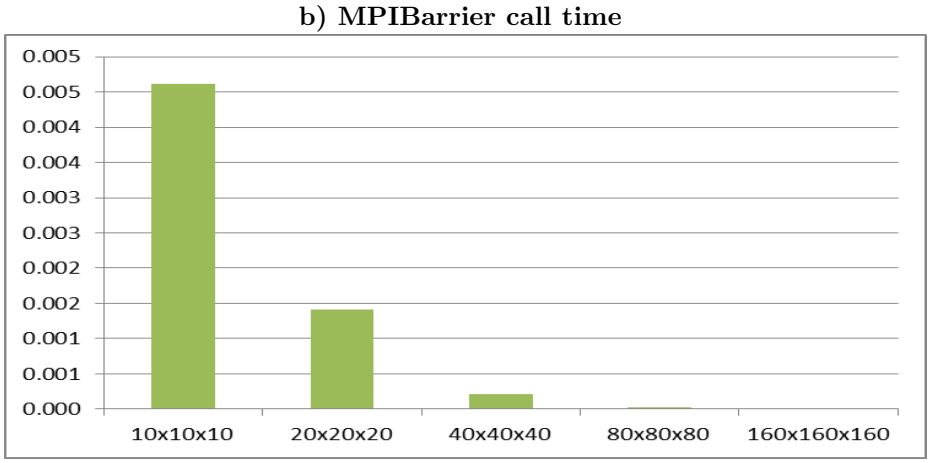

Fig. 5: Percentage of MPIBarrier call time in the maximum computation time for a single core.

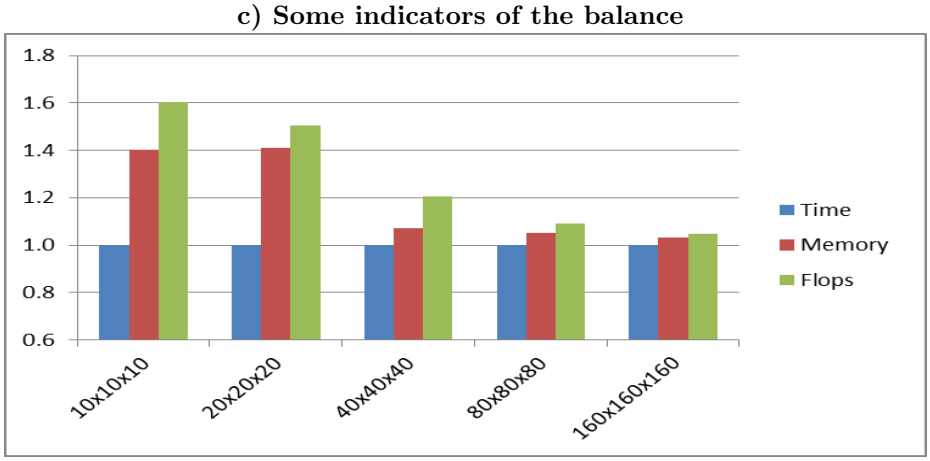

Fig. 6: [Blue]The ratio of the maximum time for a single core to a minimum. [Red]The ratio of the maximum memory usage to a minimum during the calculation. [Green]The ratio of the maximum flops indicator to a minimum during the calculation.

c) Network activity

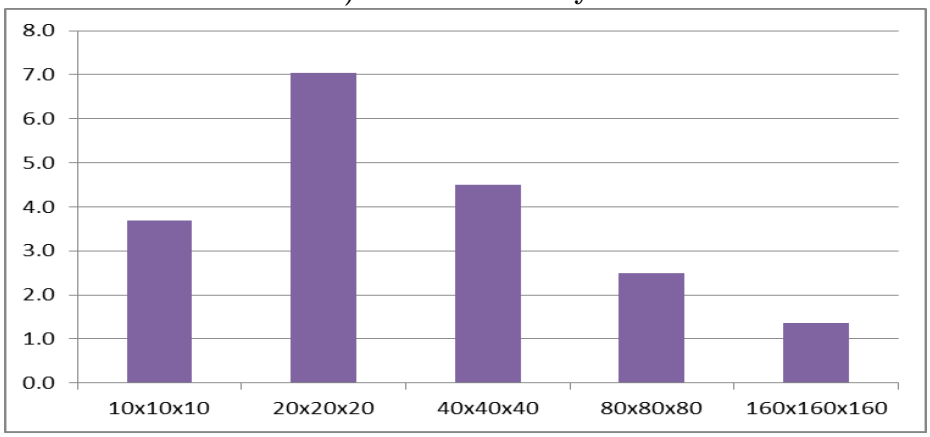

Fig. 7: Percentage of MPI messages in maximum computation time for a single core.

2.2 Second experiment

The purpose of the second experiment is the assessment of the efficiency of the algorithm parallelization by increasing the number of computing nodes. Fixed parameters for this task are: computational domain size 160x160x160 and the grid size 50x50x50. Variable parameter is the number of nodes: from 1 to 5 nodes included 28 processor cores on each node. Results of the experiment are presented in the following figures:

a) Growth of the size of the problem and calculation time

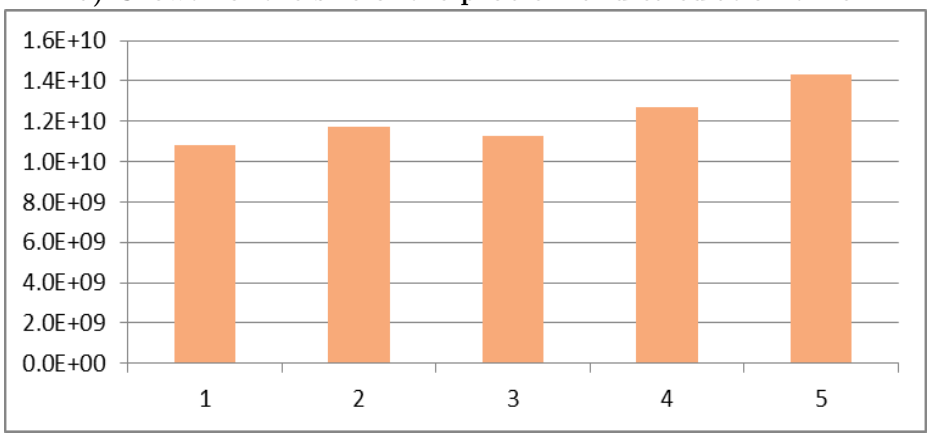

Fig. 8: Total memory used, bytes.

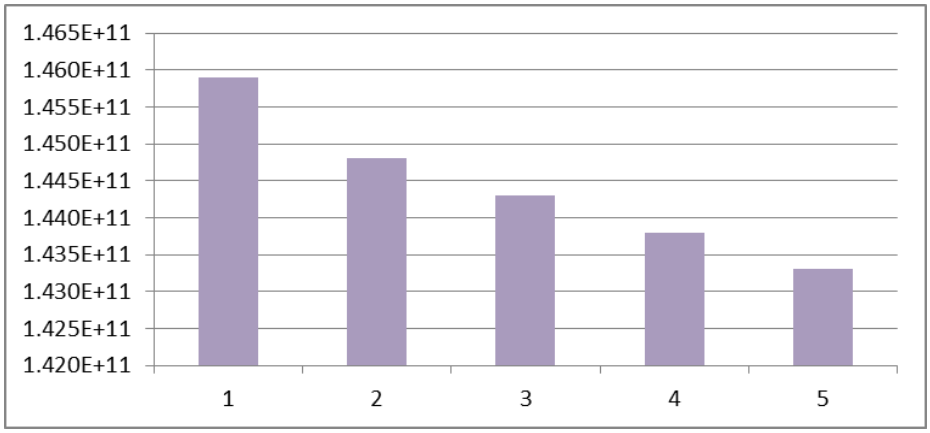

Fig. 9: Total flops.

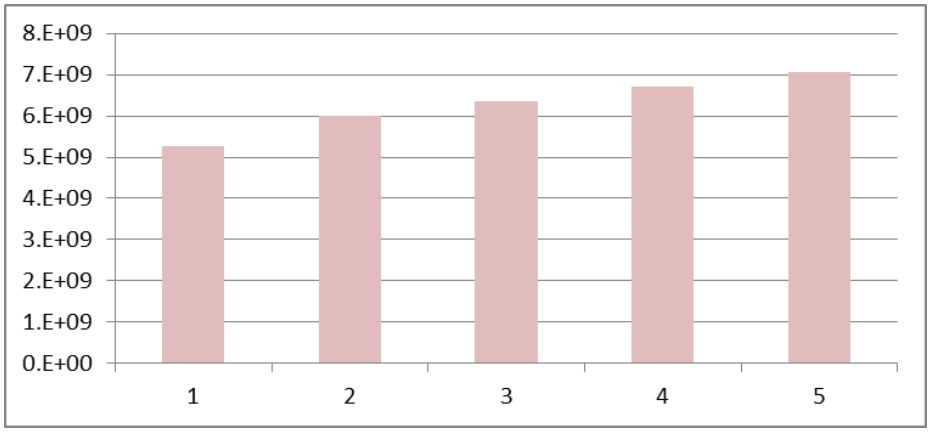

Fig. 10: MPI message total lengths , bytes.

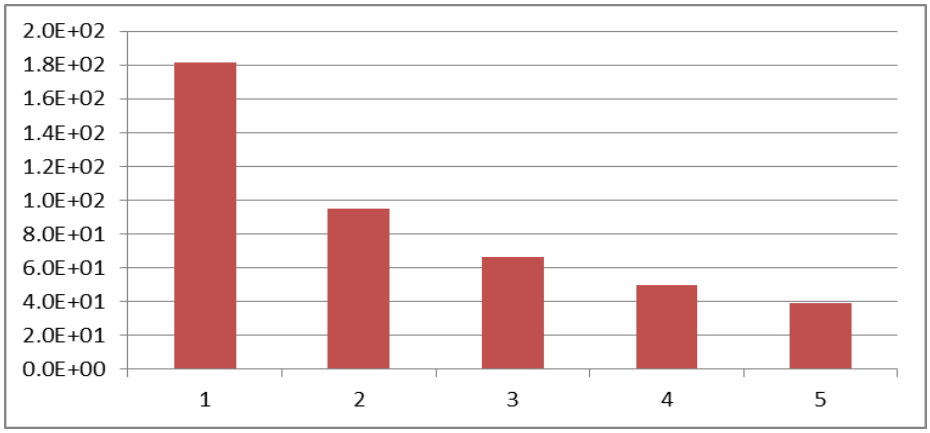

Fig. 11: Maximum computational time for a single core, sec.

b) MPIBarrier call time

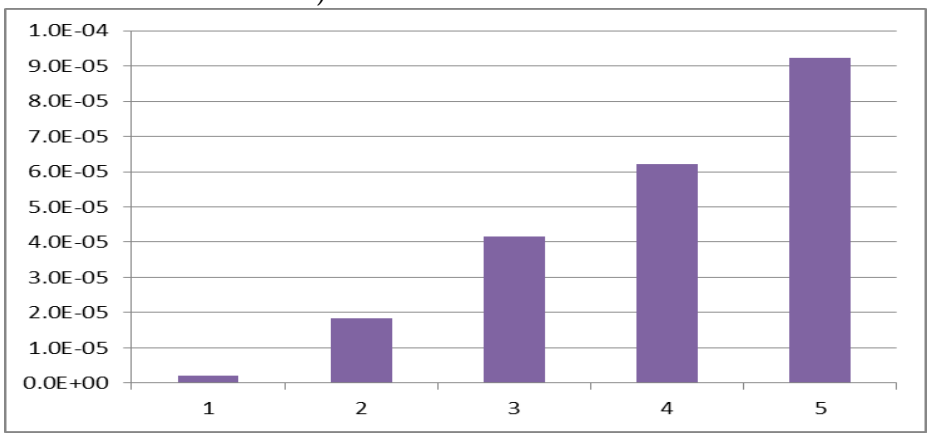

Fig. 12: Percentage of MPIBarrier call time in the maximum computation time for a single core.

c) Some indicators of the balance

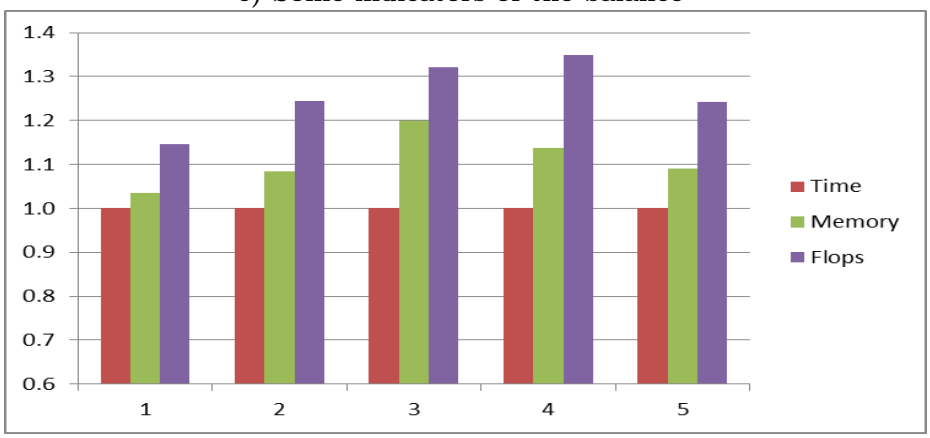

Fig. 13: [Red]The ratio of the maximum time for a single core to a minimum. [Green]The ratio of the maximum memory usage to a minimum during the calculation. [Blue]The ratio of the maximum flops indicator to a minimum during the calculation.

c) Network activity

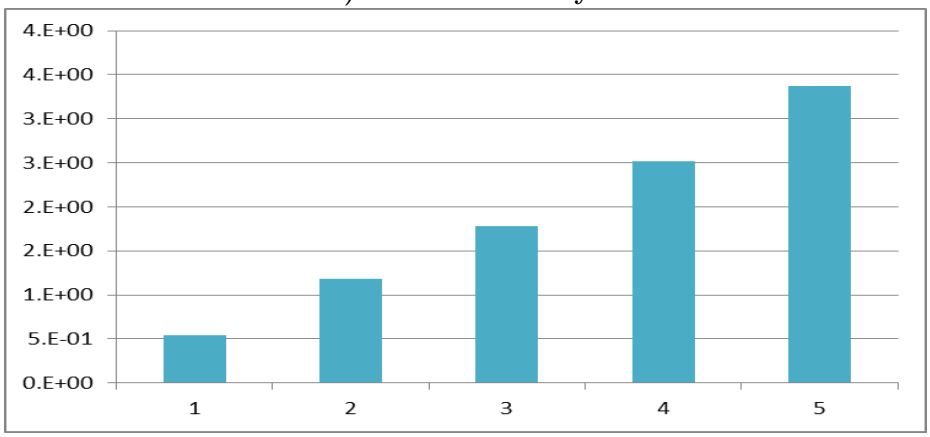

Fig. 14: Percentage of MPI messages in maximum computation time for a single core.

2.3 Third experiment

The purpose of of the third experiment is estimation of efficiency of the algorithm parallelization by increasing the number of cores on single node. Fixed parameters for this task are: only one node, computational domain domain size 160x160x160 and the grid size 50x50x50. Variable parameter is the number of cores on single node: from 1 to 28. Results of the experiment are presented in the following figures:

a) Growth of the size of the problem and calculation time

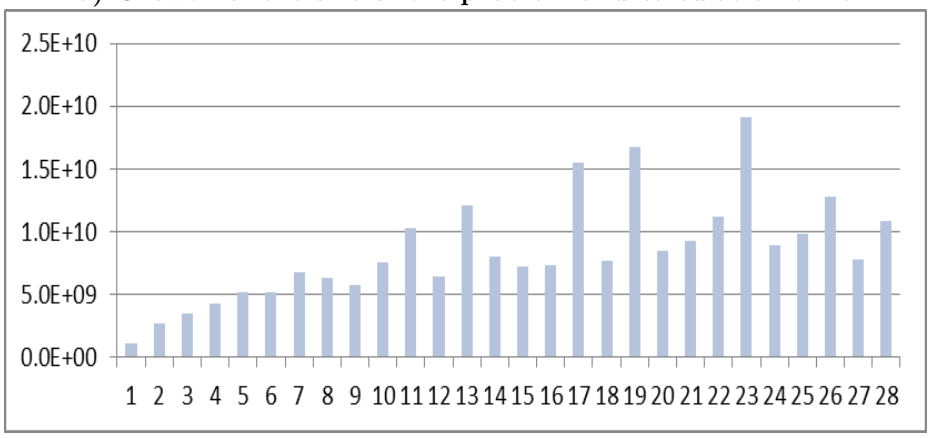

Fig. 15: Total memory used, bytes.

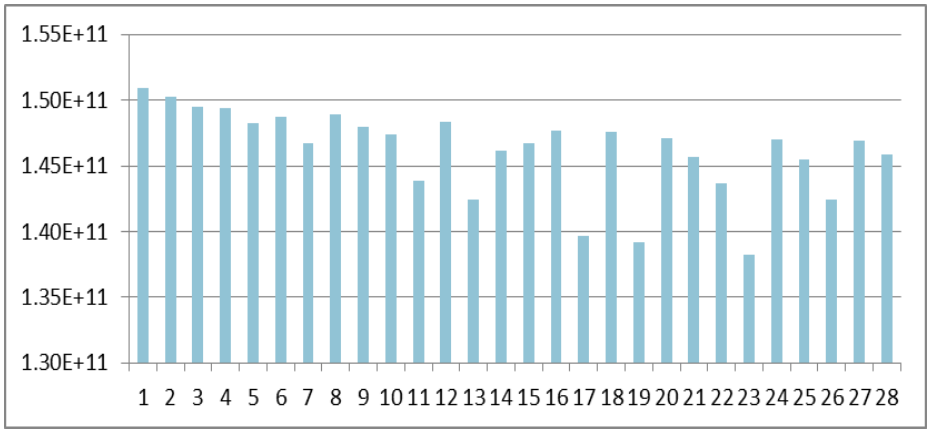

Fig. 16: Total flops.

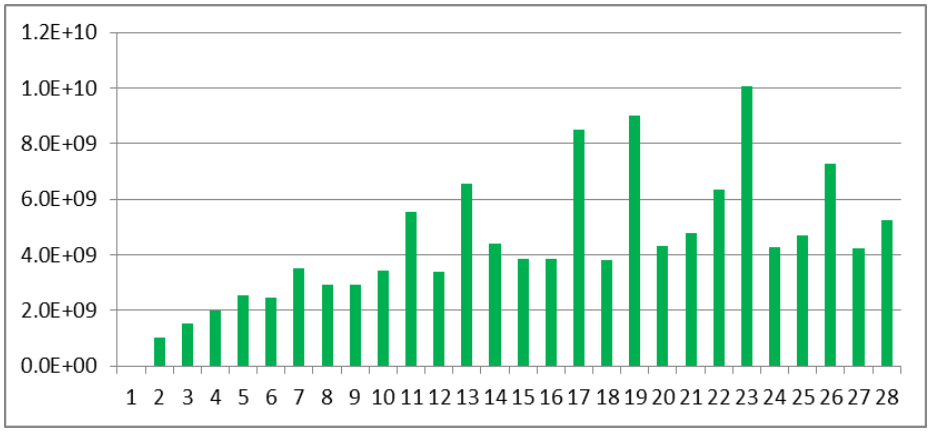

Fig. 17: MPI message total lengths , bytes.

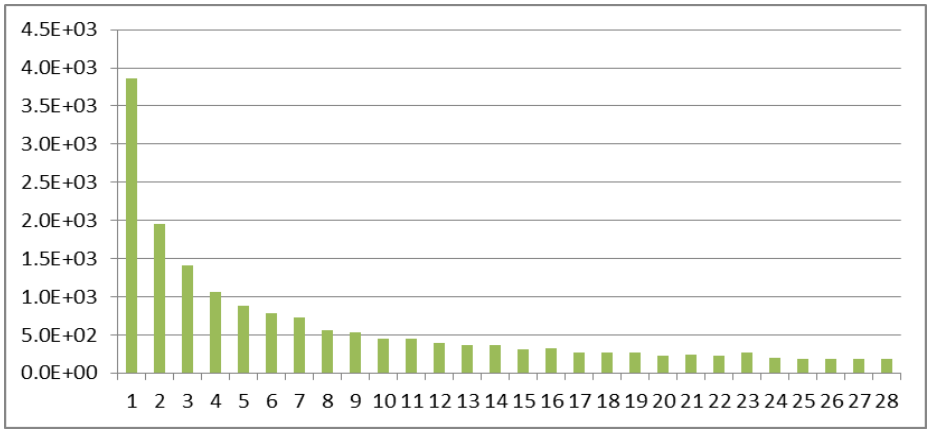

Fig. 18: Maximum computational time for a single core, sec.

b) MPIBarrier call time

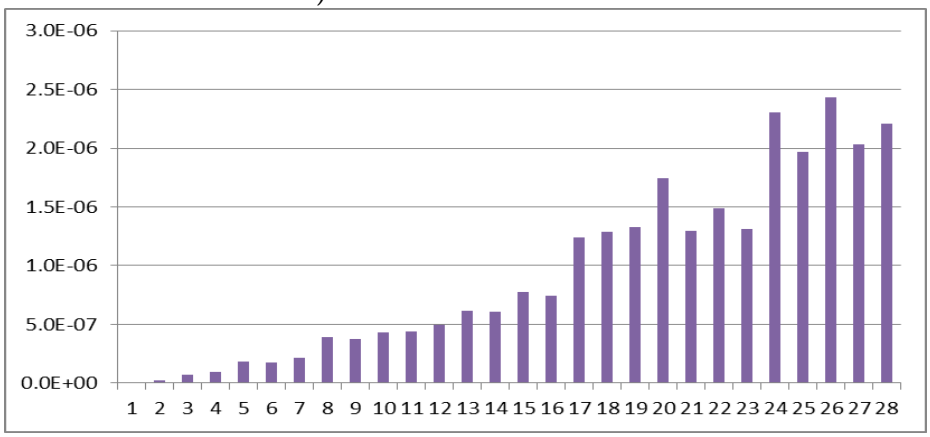

Fig. 19: Percentage of MPIBarrier call time in the maximum computation time for a single core.

c) Some indicators of the balance

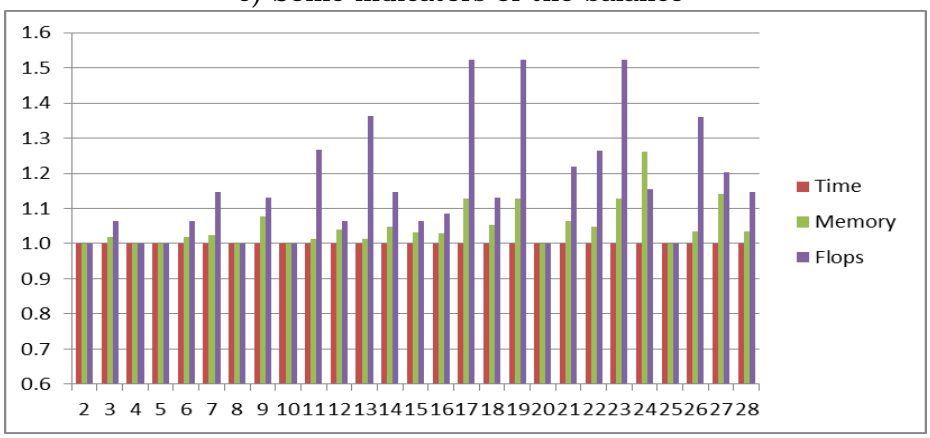

Fig. 20: [Red]The ratio of the maximum time for a single core to a minimum. [Green]The ratio of the maximum memory usage to a minimum during the calculation. [Blue]The ratio of the maximum flops indicator to a minimum during the calculation.

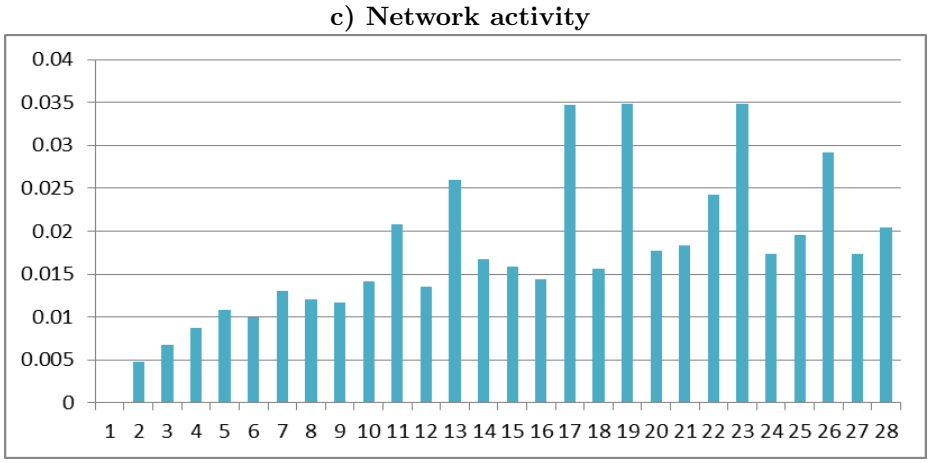

Fig. 21: Percentage of MPI messages in maximum computation time for a single core.

3 Conclusions

During computing the tasks the cluster Immers showed the best performance 4.45E+09 Flops/sec, 3.68E+09 Flops/sec and 8,03E+08 Flops/sec for the first, second and third experiments, respectively.The experiments results suggest the following conclusions:

3.1 Estimating the size of the problem and the computation time

In the first experiment, the total memory usage, the total number of operations and the total size of messages MPI grow exponentially. These indicators are rising at roughly the same speed. In the second experiment, the total amount of memory used and the total size of MPI messages almost unchanged. The total number of operations is slightly reduced, and this process requires further study. The computation time is also reduced with a logarithmic rate as expected. In the third experiment, there is a large variation in the total amount of memory used - this effect requires further study. Variations in the total number of operations and the total amount of MPI message, apparently associated to variations from memory. Computational time is change expected.

3.2 Balancing

In the first experiment, variation in the memory up to 40%, the number of operations up to 60%. In the second experiment, variation in the memory up to 20%, the number of operations up to 35%. In the third experiment, variation in the memory up to 28%, the number of operations up to 55%. The indicators seem to be unexpectedly large and require further research.

3.3 Overhead

The overhead of synchronization, as expected, increases with increasing amounts of computing nodes and decreasing the size of the grid. Meanwhile, the proportion of execution time MPI Barrier in total computation time is less than 0.001%, which is quite a bit.

References

1. Balay, S., Abhyankar, S., Adams, M., Brown, J., Brune, P., Buschelman, K., Eijkhout, V., Gropp, W., Kaushik, D., Knepley, M., et al.: Petsc users manual revision 3.5 (2014)
2. Collier, N., Dalcin, L., Calo, V.M.: Petiga: high-performance isogeometric analysis. preprint arXiv:1305.4452 (2013)
3. Cottrell, J., Hughes, T., Bazilevs, Y.: Isogeometric Analysis: Toward Integration of CAD and FEA. Wiley (2009)
4. Elder, K.R., Grant, M.: Modeling elastic and plastic deformations in nonequilibrium processing using phase field crystals. Physical Review E 70(5), 051605 (2004)
5. Elder, K.R., Katakowski, M., Haataja, M., Grant, M.: Modeling elasticity in crystal growth. Physical review letters 88(24), 245701 (2002)
6. Galenko, P., Gomez, H., Kropotin, N., Elder, K.: Unconditionally stable method and numerical solution of the hyperbolic phase-field crystal equation. Phys. Rev. E 88(1), 013310 (2013)
7. Gomez, H., Calo, V., Bazilevs, Y., Hughes, T.: Isogeometric Analysis of the Cahn-Hilliard phase-field model. Comput. Methods Appl. Mech. and Eng. 197, 43334352 (2008)
8. Gomez, H., Hughes, T., Nogueira, X., Calo, V.: Isogeometric Analysis of the isothermal Navier-Stokes-Korteweg equations. Comput. Methods Appl. Mech. and Eng. 199(25-28), 1828–1840 (2010)
9. Gomez, H., Nogueira, X.: An unconditionally energy-stable method for the phase field crystal equation. Computer Methods in Applied Mechanics and Engineering 249, 52–61 (2012)
10. Tegze, G., Bansel, G., Tóth, G.I., Pusztai, T., Fan, Z., Gránásy, L.: Advanced operator splitting-based semi-implicit spectral method to solve the binary phase-field crystal equations with variable coefficients. Journal of Computational Physics 228(5), 1612–1623 (2009)
11. Tóth, G.I., Tegze, G., Pusztai, T., Tóth, G., Gránásy, L.: Polymorphism, crystal nucleation and growth in the phase-field crystal model in 2d and 3d. Journal of Physics: Condensed Matter 22(36), 364101 (2010)
12. Zhang, J.: A petsc-based parallel implementation of finite element method for elasticity problems. Mathematical Problems in Engineering vol. 2015, 7 (2015)