

Remote control system for high-performance computer simulation of crystal growth by the PFC method

Evgeny Pavlyuk¹, Ilya Starodumov², Sergei Osipov³

¹ Department of computational mathematics and computer science, Ural Federal University, 620000 Ekaterinburg, Russia

² Laboratory of Multi-Scale Mathematical Modelling, Ural Federal University, 620000 Ekaterinburg, Russia

³ School of Natural Sciences and Mathematics, Ural Federal University, 620000 Ekaterinburg, Russia

E-mail: evgeny.pavlyuk@urfu.ru

Abstract. Modeling of crystallization process by the phase field crystal method (PFC) - one of the important directions of modern computational materials science. In this paper, the practical side of the computer simulation of the crystallization process by the PFC method is investigated. To solve problems using this method, it is necessary to use high-performance computing clusters, data storage systems and other often expensive complex computer systems. Access to such resources is often limited, unstable and accompanied by various administrative problems. In addition, the variety of software and settings of different computing clusters sometimes does not allow researchers to use unified program code. There is a need to adapt the program code for each configuration of the computer complex. The practical experience of the authors has shown that the creation of a special control system for computing with the possibility of remote use can greatly simplify the implementation of simulations and increase the performance of scientific research. In current paper we show the principal idea of such a system and justify its efficiency.

1. Introduction

Numerical modeling of physical processes is often associated with the solution of complex differential systems of equations. Such tasks can be solved only with the use of computer technology. In the case of complex three-dimensional simulations, a typical personal computer may not suffice, which requires modeling on high-performance computing systems. The phase field crystal method is one of such algorithms. The PFC model has been formulated in a parabolic form [1–3]. Then it has been extended to the form described by partial differential equation of a hyperbolic type (MPFC) [4–6, 6–10]. The MPFC model can be expressed using the free-energy functional

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



where Ω is the computational domain, in which it's possible to define the chemical potential as

$$\mu(\phi) = \frac{\delta \mathcal{F}}{\delta \phi} = \frac{df}{d\phi} + 2\nabla^2 \phi + \nabla^4 \phi, \quad (2)$$

where ϕ describes conserved order parameter. The free energy density $f(\phi)$ represents the homogeneous part of energy and in the present report it is written in the form of Landau-de Gennes:

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

Here, ϵ is the dimensionless overcooling and α is the measure of metastability. Finally, the modified (hyperbolic) phase field crystal model can be expressed by the following equation:

$$\tau \frac{\partial^2 \phi}{\partial t^2} + \frac{\partial \phi}{\partial t} = \nabla^2 \mu, \quad (4)$$

where t is the time, τ is the relaxation time of the atomic flux to its stationary state.

2. HPC researches of the MPFC task

The research of the MPFC equation (4) is associated with a complex computational task. The necessity to solve differential equations of the 6th order in space and 2nd order in time requires the use of special numerical algorithms [10]. Analysis of the effectiveness of this method presented [11, 12]. that, despite the high rates of parallelization of computing processes, the method requires significant computer resources. This means that the implementation of simulations is necessary to use high-performance computing cluster. Access to such clusters, as a rule, has a number of significant limitations, which makes it difficult to carry out simulations, especially in a continuous mode. Among these limitations are the following:

- 1) The differences in the configuration of software and hardware do not allow for the calculation without the preliminary settings of computing programs, postprocessors and additional user's applications.
- 2) In the case of a large amount of data with results (in the calculations by the MPFC method this volume can reach hundreds of gigabytes), there is a need for a large data storage. Practice experience presents that usually HPC centers are not ready to provide the storage quota of significant capacity.
- 3) Regular transfer of large amounts of data from a cluster to a user's computer requires a stable high-speed network connection.
- 4) The need for special software to process the results of calculations limits the user in using only a preconfigured computer.

Additionally, the following system requirements are relevant:

- 1) View the task status and the system components statuses (for example, nodes load).
- 2) Task results postprocessing (with a choice of tools).
- 3) Ability to view the task results without the need to use special soft.
- 4) The ability to exchange data between clusters to distribute one task between different clusters.
- 5) Monitoring the task execution and restarting, if necessary, to ensure fault tolerance.
- 6) Ability to duplicate the critical elements of the system to ensure fault tolerance.
- 7) Performance of component interactions.
- 8) Scalability to support large numbers of components and interactions among components: users, clusters, tasks.
- 9) Modifiability of components to meet changing needs (even while the application is running).
- 10) Simplicity of a uniform interface.

3. Remote control system of computations for simulation

To solve the described problems the following special system was proposed in Figure 1:

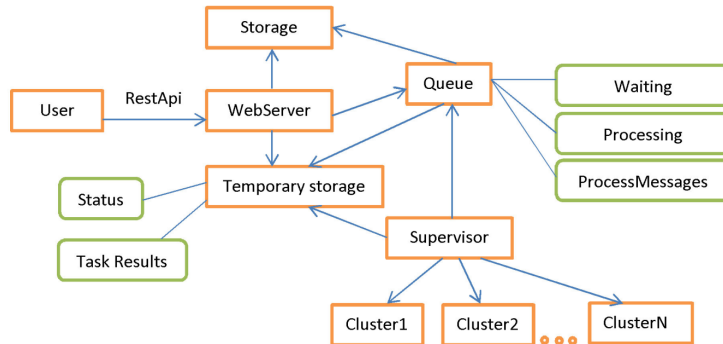


Figure 1. The remote control system of computations for simulation.

Web-server process user commands. Commands are passed through http requests. For example,

- Get system status : `GET /status`
- Get task status: `GET /status?tag="taskname_20170130"`
- Run task (precisely put in task queue): `PUT /task?tag="taskname_20170130"&clusters="clustername"¶meters='{ "node_count"=3, "time_limit"=10000}'&d @ "Problem_Conditions.csv"`
- Get task result: `GET /result?tag="taskname_20170130"`
- Add cluster to cluster set: `POST /add_cluster?parameters='{ "ip": "192.168.99.100", "name": "Cluster1" }'`

This format allows to create console and web clients, to use the system as a server for calculations in some other software system. In addition, the web server can process a command to postprocess task results (the handler can be on a separate computer), save the results in the storage, view processed results via the web interface with ParaviewWeb. This allows to view the results from any computer that has just a browser.

Task queue is a system of robust storing and transmitting messages between system components based on AMQP (Advanced Message Queuing Protocol). The queue performs routing, guarantees delivery, distributes data flows, subscribes to the required message types. In addition, the queue must support horizontal scaling. The system have three separate queues:

- Waiting – for tasks that are waiting to be launched;
- Processing – for already running tasks. You may need to restart such a task;
- ProcessMessages – there will be data to exchange between clusters.

Using such a queue will eliminate data loss cases due to the impossibility of executing the request. Even if the whole system or its individual components are stopped, the data of user requests and intermediate calculations will not be lost and the calculations can be continued from the point of stop.

Temporary Storage is a key-value data base. It is for storing queue data, and ancillary data. Requirements for this storage are following:

- The ability to run copies on several independent nodes to ensure fault tolerance;
- Access speed is primarily important for tasks that exchange data between clusters.

Supervisor is a system that controls the set of clusters. Its main tasks are:

- 1) Start / Stop tasks. Each cluster can have specific control scripts. The supervisor calls these scripts through a single interface;
- 2) Collect the statuses of each cluster, server components and individual tasks, and save them to the Temporary Storage;
- 3) Collecting the task results and logs and saving in Temporary Storage for debugging and starting the task from the checkpoint load balancing;
- 4) Control over the task execution time and emerging problems.

The system has a modular architecture, each module can be replaced if necessary, without having to stop and restart the entire system.

4. Conclusions

In this paper, the problem of modeling crystal growth by the highly effective MPFC method on high-performance computing systems is considered. The authors describe the difficulties encountered by researchers in carrying out serial computer simulations aimed at solving complex problems. As a solution to these problems, the authors propose a scalable and fault-tolerant system for controlling computations, the concept of which is described in this paper. The system makes it possible to run tasks on a variety of clusters with a variety of configurations, allowing a variety of postprocessing of the results of calculations and having a convenient API (application programming interface) that allows to create console and web clients, use the system as a server for calculations in some other software system.

References

- [1] Elder K R, Katakowski M, Haataja M and Grant M 2002 *Physical Review Letters* **88** 245701
- [2] Elder K R and Grant M 2004 *Physical Review E* **70** 051605
- [3] Elder K R, Provatas N, Berry J, Stefanovich P and Grant M 2007 *Physical Review B* **75** 064107
- [4] Stefanovic P, Haataja M and Provatas N 2006 *Physical Review Letters* **96** 225504
- [5] Stefanovic P, Haataja M and Provatas N 2009 *Physical Review E* **80** 046107
- [6] Galenko P K, Danilov D A and Lebedev V G 2009 *Physical Review E* **79** 051110
- [7] Galenko P K and Elder K R 2011 *Physical Review B* **83** 064113
- [8] Wang C and Wise S M 2011 *SIAM Journal on Numerical Analysis* **49** 945–969
- [9] Baskaran A, Hu Z, Lowengrub J S, Wang C, Wise S M and Zhou P 2013 *J. Comput. Phys.* **250** 270–292 ISSN 0021-9991
- [10] Galenko P K, Gomez H, Kropotin N V and Elder K R 2013 *Phys. Rev. E* **88**(1) 013310
- [11] Starodumov I, Pavlyuk E, Klyuev L, Kovalenko M and Medyankin A 2015 *Ceur-WS Proceedings* **1513** 109–122
- [12] Starodumov I, Pavlyuk E, Abramov S, Klyuev L, Galenko P and Alexandrov D 2016 *Bulletin of Udmurt University. Mathematics, Mechanics* **26** 445–450 URL <http://dx.doi.org/10.1063/1.4959750>

5. Acknowledgments

Authors acknowledge support from the Russian Science Foundation under project number 16-11-10095.