TECHNOLOGIES FOR ACCELERATING CALCULATIONS IN TOKAMAK SIMULATION MODELS [[1]](#footnote-1)\*)

DOI: 10.34854/ICPAF.2022.49.1.173

Scopintsev D., Dokuka V., Khayrutdinov R.

SRC RF TRINITI, Moscow, Troitsk, Russia. [scopintsev.d.a@triniti.ru](mailto:scopintsev.d.a@triniti.ru), [khayr@triniti.ru](mailto:khayr@triniti.ru), [dokuka@triniti.ru](mailto:dokuka@triniti.ru)

The paper describes technologies for accelerating calculations when using application software packages implemented in the FORTRAN language for modeling plasmophysical processes in tokamaks.

At the same time, the difficulty in modeling is due to processes that are different in nature, which have spatial and temporal scales that differ by many parameters. That is why it is necessary to combine various specialized codes into a single system, which together make it possible to design a tokamak installation and analyze its operation [1].

One of the important tasks is to predict the behavior of the plasma cord during the experiment — the so-called calculation of the discharge scenario. The development of appropriate computer codes and conducting computational experiments with their help is a task of extreme computational complexity [2].

Simulation models of a tokamak reactor include modeling not only the plasma-physical states of the discharge, but also the formation of tools (algorithms and program codes) for components and systems included in the plasma control circuit. Such as, a plasma parameter recovery module based on magnetic measurements, a winding power supply system, a plasma magnetic control system designed to control the current, position and shape of the plasma, etc. [3].

Model codes place very serious demands on computing resources. The problem arises of distributing computations across an array of computing cores, which is possible when using a CUDA-enabled compiler for Fortran, for example, Portland Group Inc. (PGI), and using CUDA libraries with modern GPU architecture [4].

Arrays should be transferred to GPU memory for parallel processing, and then returned back. Therefore, they must be static. Overall performance is limited by these operations. The appearance of a new NVLink bus and HBM2 memory, which can be accessed by the controlling CPU and GPU, should remove this restriction.

The code was modified using directives to the compiler. The implementation algorithm was tested on scenarios where each equation was solved separately and iterations were carried out for each moment of time, until the required absolute convergence accuracy was achieved. The convergence time when using the GPU has been reduced by an order of magnitude.

References

1. Kostomarov D.P., Zaitsev F.S., Shishkin A.G., Sychugov D.Yu., Stepanov S.V., Suchkov E.P., “Virtual tokamak” software toolbox. Вестн. Моск. Ун-та. Сер. 15. Вычисл. матем. и киберн. 2011. №4 c. 48-56.
2. Зайцев Ф.С. Математическое моделирование эволюции тороидальной плазмы. М.: МАКС Пресс, 2005. С. 524.
3. R.R. Khayrutdinov and V.E. Lukash, J. Comput.Phys. 109, 193 (1993).
4. Рутш Г., Фатика М. CUDA Fortran для инженеров и научных работников./ пер. с англ. Слинкин А.А. – М.: ДМК Пресс, 2014. – 364 с.

1. \*) [abstracts of this report in Russian](http://www.fpl.gpi.ru/Zvenigorod/XLIX/E/ru/IL-Scopintsev.docx) [↑](#footnote-ref-1)