Fully non-blocking communication-computation overlap using assistant cores towards Exa-scale computing

Motoki Nakata
National Institute for Fusion Science
322-6 Oroshi-cho, Toki, Gifu, Japan
509-5292
nakata.motoki@nifs.ac.jp

Masanori Nunami
National Institute for Fusion Science
322-6 Oroshi-cho, Toki, Gifu, Japan
509-5292
nunami.masanori@nifs.ac.jp

Shinsuke Satake
National Institute for Fusion Science
322-6 Oroshi-cho, Toki, Gifu, Japan
509-5292
satake@nifs.ac.jp

Yoshihiro Kasai
Fujitsu Limited
1415 Midori-cho, Tsuruga, Nagano
Japan 380-0813
kasai.yoshihiro@jp.fujitsu.com

Shinya Maeyama
Nagoya University
Furo-cho, Nagoya, Japan 464-8601
smaeyama@p.phys.nagoya-u.ac.jp

Tomo-Hiko Watanabe
Nagoya University
Furo-cho, Nagoya, Japan 464-8601
watanabe.tomohiko@nagoya-u.jp

Yasuhiro Idomura
Japan Atomic Energy Agency
178-4 Wakashiba, Kashiwa, Chiba,
Japan 277-0871
idomura.yasuhiro@jaea.go.jp

ABSTRACT

A fully non-blocking optimized Communication-Computation overlap technique using assistant cores (AC), which are independent from the calculation cores, is proposed for the application to the five-dimensional plasma turbulence simulation code with spectral (FFT) and finite-difference schemes, towards Exa-scale supercomputing. The effects of optimization are examined in Fujitsu FX100 (2.62PFlop/s) with 32 ordinary cores and 2 Assistant cores/node, where AC enables us to employ the fully non-blocking MPI communications overlapped by the thread-parallelized calculations with OpenMP Static scheduling with much less overheads. It is clarified that the combination of the non-blocking communications by AC and the static scheduling leads to not only reduction in OpenMP overhead, but also improved load/store and cash performance, where about 22.5% improved numerical performance is confirmed in comparison to the conventional overlap by the master thread communications with dynamic scheduling.

KEYWORDS

Plasma turbulence, Communication-Computation overlap, Assistant cores, Parallel performance

ACM Reference format:
Motoki Nakata, Masanori Nunami, Shinsuke Satake, Yoshihiro Kasai, Shinya Maeyama, Tomo-Hiko Watanabe, and Yasuhiro Idomura. . Fully non-blocking communication-computation overlap using assistant cores towards Exa-scale computing. In *Proceedings of , , ,* 3 pages.
DOI:

,
. .
DOI:

1 INTRODUCTION

Realizing future fusion energy reactor, it is crucial to clarify the physical mechanisms of heat and particle losses due to turbulence in burning plasmas. The first-principle-based gyrokinetic simulation, which solves the time evolution of the plasma distribution function on the five-dimensional phase space, is a promising approach for the turbulent transport in magnetically confined toroidal plasmas, but computationally challenging. In order to establish the predictive turbulence simulations for the fusion burning plasmas with the multiple spatio-temporal scale fluctuations and the multiple particle species, advanced numerical optimizations are indispensable for the next-generation Exa-scale supercomputers. Applying Communication-Computation overlap techniques[1] to the 5-D electromagnetic gyrokinetic Eulerian code GKV[2], the million-cores-class strong scaling with 99.99994% efficiency has been achieved on a Peta-scale super computer system, K computer with 8 cores/node. In this work, we present a more optimized Communication - Computation overlap by utilizing "Assistant cores (AC)" to overcome the dynamic scheduling overheads in the thread parallelizations such as OpenMP, for the future Many-core architectures.

2 COMMUNICATION-COMPUTATION OVERLAP

The numerical cost of point-to-point and/or collective communications in the fluid-type simulation codes with multi-dimensional domain decomposition is a crucial issue for scalabilities of the code. In order to mask these communication costs by the independent computations, the communication - computation overlap technique[1] has been developed by utilizing the OpenMP dynamic scheduling, where the master thread performs the Send/Recv and/or AlltoAll communications during the finite-difference and/or spectral(FFT)

computations by the other threads. The overlap technique is successfully applied to large scale simulations on K computer with multiple cores (~ 8 cores). However, in recent Many-core processors, the scheduling overhead in thread parallelizations is one of concerns[3], and it may degrade the scalability performance in the Communication-Computation overlap based on OpenMP dynamic scheduling. Here we propose a solution with an optimized overlap technique to avoid this issue.

3 THE GKV CODE

From the numerical aspects of view, the GKV code performs computational fluid dynamics (CFD) calculations in the 5-D phase space (x, y, z, v, μ) for each particle species (s). The code mainly consists of three parts: spectral calculations in method in x and y [using the parallel FFT with the 3/2 de-aliasing rule], finite difference calculations in z, v and μ, a field solver (including integrations over v, μ and s), where the computations are parallelized by using the OpenMP/MPI hybrid parallelization. Since five-dimensional domain decomposition is applied for y, z, v, μ and s, the MPI communications are data transpose for the parallel FFT in x and y, point-to-point communications in z, v and μ, and reduction over v, μ and s. The most time consuming part is the parallel FFT, where the data transpose often degrades the scalability.

4 OVERLAP WITH NON-BLOCKING COMMUNICATIONS BY ASSISTANT CORES

In a Japanese national project, FLAGSHP2020, Post-K computer is being developed towards Exa-scale computing, and Scalable Many Core (SMaC) architecture[4] with "Assistant cores (AC)" will be applied. In this work, we develop an optimized overlap with fully non-blocking communications by the assistant cores, instead of the calculation core, where Fujitsu FX100 (Plasma Simulator at NIFS, 2.62PFlop/s) with 32 ordinary cores and 2 assistant cores/node is used to examine the effects of the optimization in GKV kernel. It should be noted that AC is not only for reducing the OS Jitter, but also for the improved performance: one can employ fully non-blocking MPI communications, such as ISend/IRecv, IAllreduce, and IAlltoAll, on the assistant cores. Then all the ordinary cores can perform the computations with OpenMP static scheduling, where the scheduling overhead is much smaller than that in the OpenMP dynamic case. Blue Gene/Q also has the similar architecture of 16 cores with two redundant cores as a spare core and/or a core offloading I/O operation[5].

As shown in Fig. 1, the numerical performance is compared among the cases for MS(D): overlap by the master thread communications with OpenMP dynamic scheduling, AC(D): overlap by the assistant core communications with OpenMP dynamic scheduling, and AC(S): overlap by the assistant core communications with OpenMP static scheduling, where the chunk size is set to the unity for the MS(D) and AC(D). The problem size of about 10 billion grid points is used for the measurements, where 432 MPI processes with OpenMP thread parallelization for 16 cores/process are assigned on FX100. Then we found successful overlap by AC, indicating ~ 22.5% improvement of the performance from MS(D) to AC(S), where all the MPI processes show the similar improvement (but shown only for rank0 in Fig. 1). By using the precise program

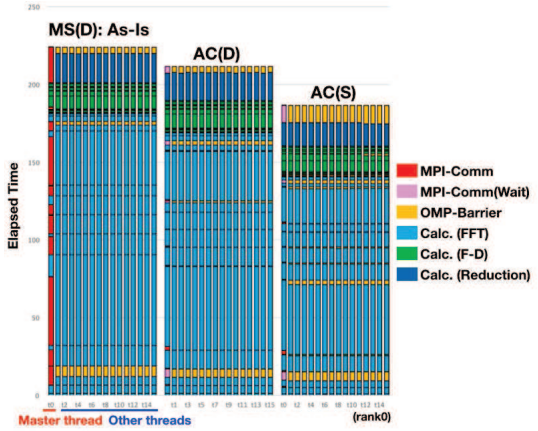

Figure 1: Comparison of the numerical costs among MS(D)(As-is), AC(D), and AC(S) in GKV kernel.

analyzer, it is clarified that the static scheduling leads to not only reduction in OpenMP overhead, but also improved load/store and cash performance (including hardware/software prefetch) with the continuous memory access for larger chunk size. Here, the OpenMP overhead is evaluated by the difference between thread parallelized loop and non-parallelized loop with no computations. It is also found that, when the chunk size in AC(D) is optimized to the value near the OpenMP static one, the comparable performance to AC(S) is achieved. Thus, one can make use of both of AC(S) and AC(D) flexibly in wide applications on Many-core systems.

5 SUMMARY

In this work, a fully non-blocking optimized Communication - Computation overlap technique is presented, then we found that: (i)Assistant cores (AC) enable us to employ fully non-blocking MPI communications in the code, (ii)In the overlap by AC, all the OpenMP threads concentrate their computations, and the usage of static scheduling leads to 22.5% improvement of the numerical performance, (iii)Chunk-size optimization with AC enables us to use flexibly the static/dynamic scheduling of the thread parallelization in future Many-core system. By applying the present overlap technique, we realized a 5-D turbulence simulations for complex geometry plasma with both deuterium ions and electrons[6], and high performance computing significantly accelerates the plasma and fusion research.

6 ACKNOWLEDGEMENTS

This work is supported by NIFS-Fujitsu collaborations on Plasma Simulator (FX100), the MEXT Grant No. 17K14899, and the MEXT Grant for Post-K project: Development of innovatives clean energy, Core design of fusion reactor(6-D). Numerical simulations were performed by Plasma Simulator in NIFS.

REFERENCES

[1] S. Maeyama, Y. Idomura, et al., "Improved strong scaling of a spectral/finite difference gyrokinetic code for multi-scale plasma turbulence", Parallel Computing 49, 1 (2015)

[2] T.-H. Watanabe and H. Sugama, "Velocity-space structures of distribution function in toroidal ion temperature gradient turbulence", Nuclear Fusion 46, 24 (2006)

[3] T.A.J. Ouermi and A. Knoll et al., "OpenMP 4 Fortran Modernization of WSM6 for KNL", Proceedings of PEARC17 (2017)

[4] T. Shimizu, "FUJITSU HPC and the Development of the Post-K Supercomputer", Exhibitor Forum in SC16 (2016)

[5] https://www-03.ibm.com/systems/technicalcomputing/solutions/bluegene/

[6] M. Nakata, M. Nunami et al., "Isotope Effects on Trapped-Electron-Mode Driven Turbulence and Zonal Flows in Helical and Tokamak Plasmas", Physical Review Letters 118, 165002 (2017)