Chapter 4

Large-scale Parallel Numerical Computing Technology Research Team

4.1 Members

Toshiyuki Imamura (Team Leader)

Yiyu Tan (Research Scientist)

Daichi Mukunoki (Research Scientist)

Shuhei Kudo (Postdoctoral Researcher)

Takuya Ina (Technical Staff)

Tetsuya Sakurai (Senior Visiting Researcher, University of Tsukuba)

Daisuke Takahashi (Senior Visiting Researcher, University of Tsukuba)

Franz Franchetti (Senior Visiting Researcher, Carnegie Mellon University)

Yusuke Hirota (Visiting Researcher, Tokyo Denki University)

Sarah Huber (Visiting Researcher, Bergische Universtät Wuppertal)

Martin Galgon (Visiting Researcher, Bergische Universtät Wuppertal)

Andreas Marek (Visiting Researcher, Max-Plank Computing & Data Facility)

Takeyuki Harayama (Intern, University of Tsukuba)

Chen Yen-Chen (Intern, University of Tokyo)

Ryuki Shimotori (Intern, University of Yamaanashi)

Aya Motohashi (Assistant)

4.2 Research Activities

The Large-scale Parallel Numerical Computing Technology Research Team conducts research and development of numerical software for the national flagship systems, K computer and the supercompter Fugaku. In particular, we are focusing on significant technical issues when we face at extensive computing, such as, large-scale, highly parallel and high-performance. In general, simulation programs require various numerical techniques to solve systems of linear equations, to solve eigenvalue problems, to compute and solve non-linear equations, and to do fast Fourier transforms. From a mission critical point of view, it is natural for us to develop and deploy highly-parallelized and scalable numerical software integrated over a software framework dedicated on K and Fugaku as well, but not limited on a target platform. It comprises above-mentioned software components (numerical software) in order to run a specific simulation code which come from scientific and engineering domain problems.

Also, the K- and Fugaku- related issues were supposed to be also our challenging works such as communication reducing and avoidance, cooperation with advanced devices, fault detection and recovery, and precision-aware computing (higher-or-reduced/variable-or-mixed accuracy).

Since 2016, we have added three new research themes as recipes for long-term research goals;

1. investigation of unexplored and conventional numerical fields,

2. precision-aware computing and numerical reproducibility,

3. acceleration on some emerging devices such as an FPGA.

We are going to complete our mission through a tighter collaboration among trilateral communities from computational science (simulation), computer science (hardware and software), and numerical mathematics (scheme and theories). Our final goal is to establish fundamental techniques to develop numerical software libraries for next-generation supercomputer systems based on active and vigorous cooperation within R-CCS.

For the first topic to investigate the conventional numerical algorithm and new research on unexplored filed, we have investigated the eigen/singular-value solver, three-dimensional parallel FFT, and mixed-precision HPL kernel, namely HPL-AI benchmark, on the part of which is also related to the precision awareness. Three studies cover the second topic; minimal-precision computing, accurate and reproducible BLAS, and DGEMM emulated by Tensor Cores. For the third topic, we have conducted task-parallelism on an FPGA for matrix-matrix multiplier and a super-realistic acoustic simulation by the field-rendering method using FPGA's.

4.2.1 Application of the Jacobi Rotation Kernel for the eigen- and singular value computations

The Jacobi eigen-/singular value decomposition method (the Jacobi method) is accurate and also has a simple computation pattern, which is easy to parallelize. However, its high computational cost damages its benefits. In FY2019, we developed the Jacobi Rotation Kernel (JRK), which takes the core computation part of the Jacobi method with high efficiency and lower computational. Therefore, JRK can mitigate the defect of the Jacobi method. Since 2020, we have started to research to apply JRK to the Jacobi methods and evaluate its performance [11, 35]. Fig. 4.1 and 4.2 compare the performance of our implementations of the Jacobi method with that of LAPACK's eigen-/singular value solvers, respectively. The figures show that JRK improves the performance of the Jacobi methods and reduce the gap between other eigen-/singular value computation methods.

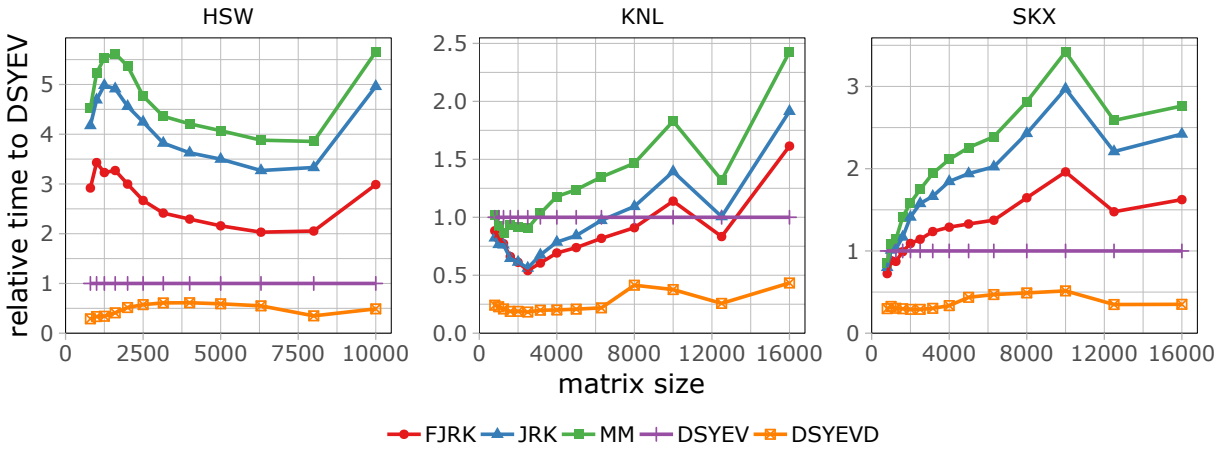


Figure 4.1: The relative computations time of eigenvalue solvers compared with the standard solver in LAPACK, DSYEV, on three different CPUs, Intel Haswell (HSW), Knights Landing(KNL), and Skylake-X (SKX). 'FJRK' and 'JRK' are the Jacobi method with JRK, and 'MM' refers to without JRK. 'DSYEVD' is the fastest solver in LAPACK. The Jacobi method is about 2–5 times slower than DSYEV without JRK. With the faster version of JRK, 'FJRK', the gaps reduce to about 1.5–3.

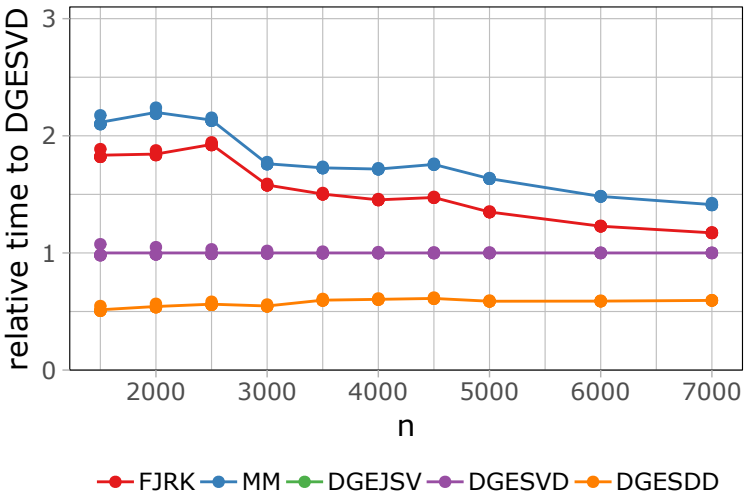


Figure 4.2: The relative computation time of singular solvers compared with the standard solver in LAPACK, DGESVD. The results are not as significant as the previous figure, Fig. 4.1, but we can observe the benefit of 'FJRK'.

4.2.2 Parallel three-dimensional FFT on a massive-scale system like K

The demand for large scale 3D FFTs will never change as we move from K to Fugaku. In order to overcome the weakness of the 3D FFT, we have been introducing a new implementation to enable the simultaneous execution of all-to-all and FFTs by proactively scheduling internal tasks with batching. This study's results were presented in ParCo2019 [4], which was done collaboratively with Prof. Yokokawa from Kobe University, and partly under the leadership of visiting scholars; Prof. Franchetti from Carnegie Mellon University and Prof. Takahashi from University of Tsukuba.

The study showed that communication and computation are possibly comparable under the K computer's specification, and communication and computation work concurrently overlapping each other. Our best achievement was a 45.9% improvement when the CFD domain's grid size was $2,048^3$, and 128 processors on the K computer were utilized. On the other hand, the core of the work insists that the communication part tends to dominate as a steady-state even if the appropriate number of batches is guaranteed. Consequently, it revealed the difficulty in future systems. For example, we already know that the balance between computation and communication varies in Fugaku, which is more communication-intensive than the K computer. Thus, it must be necessary to apply optimizations for batch scheduling, communication methods, and investigation of other advanced methods for parallel/distributed FFTs.

4.2.3 Analysis and development of software of the HPL-AI benchmark

The HPL-AI benchmark is a new benchmark for a supercomputer, similar to the well-known LINPACK benchmark but has been extended to reflect recent hardware's capability for the AI applications. The HPL-AI allows us to use the lower-precision floating-point arithmetic by introducing the mixed-precision technique in the linear solver. Therefore, supercomputers can achieve much better FLOPS than with LINPACK. Although the enormous interest from the HPC communities, because it is new, there is no reference implementation of the HPL-AI benchmark for supercomputer environments. Therefore, we developed software for the benchmark from scratch, with analysis to avoid numerical difficulties in lower-precision arithmetic [43]. We plan to benchmark our software on supercomputer Fugaku in FY2020, and the results will be released at the top500 session during ISC2020.

4.2.4 Minimal-precision computing

The mixed-precision technique utilizing fast low-precision operations is one of the promising approaches to improve the speed and energy efficiency of computations. In this FY, we have started a discussion on the precision-tuning scheme for existing mono-precision codes with RIKEN CCS, Sorbonne University (France),

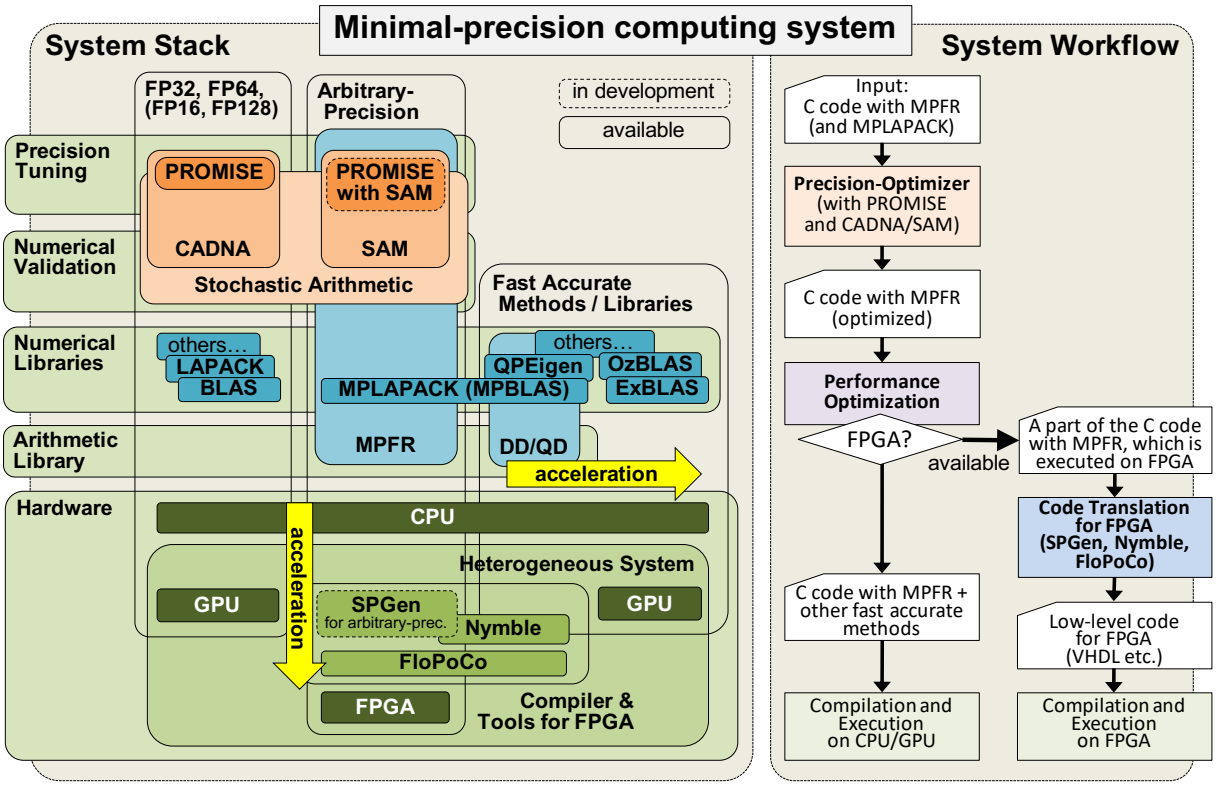


Figure 4.3: System overview required for Minimal-precision computing

and University of Tsukuba. As an intermediate result, we have proposed the minimal-precision computing system.

Figure 4.3 shows the system stack of the proposed minimal-precision computing system. Unlike the other existing precision tuning projects, it aims not only to optimize (minimize) the precision for each data in an input code but also to satisfy the demand for reliable computing – accurate and reproducible computations. Moreover, it is a system-level scheme involving both hardware and software stacks. In specific, it combines (1) a precision-tuning method based on a numerical validation method, (2) arbitrary-precision arithmetic libraries, (3) fast and accurate numerical libraries and (4) Field-Programmable Gate Array (FPGA) with high-level synthesis. Therefore, we ultimately aim to provide an arbitrary-precision computing platform with precision tuning.

One of the key technologies in our proposed scheme is the numerical validation based on Discrete Stochastic Arithmetic (DSA). The DSA enables one to obtain the number of correct digits in the computed result statistically by performing a code several times with random rounding. Then, the precision tuning is performed based on the validated result by DSA. Through numerical validation, the demand for reliable computing can be satisfied. This concept is applied even to obtain an accurate result whose accuracy is higher than that can be obtained using double-precision arithmetic. Currently, a DSA implementation for IEEE floating-point standards, CADNA, and a precision tuner based on CADNA, PROMISE, are available (both have been developed by Sorbonne University). We plan to extend and improve both libraries for our scheme.

Our proposed scheme has been presented and discussed in several international conferences (e.g., poster presentations at SC19 [21], HPC Asia 2020 [23] and 2nd R-CCS International Symposium [26], oral presentation at CRE2019 [13], SIAM PP20 [15] and others [20, 40]. Besides, we have organized two international workshops at R-CCS focusing on this topic (LSPANC 2019 June [33] and LSPANC 2020 January [41]). The white paper from the LSPANC2020 January workshop will be published (planned in FY2020 as "White Paper from Workshop on Large-scale Parallel Numerical Computing Technology (LSPANC 2020): HPC and Computer Arithmetic toward Minimal-Precision Computing", HAL-0253631).

4.2.5 Accurate and reproducible BLAS routines and CG method

Ozaki scheme is an accurate and reproducible dot-product/matrix multiplication algorithm based on the error-free transformation for dot-product/matrix multiplication proposed by Ozaki et al. in 2011. The scheme can

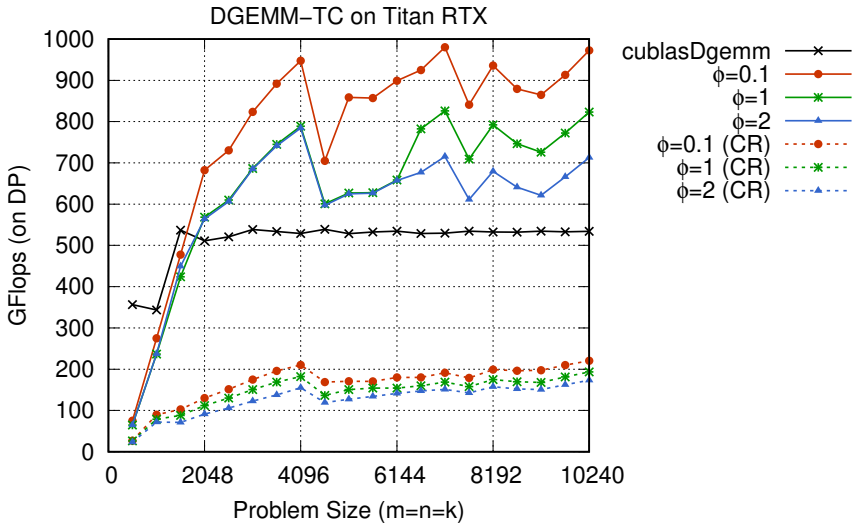


Figure 4.4: DGEMM using Tensor Cores on Titan RTX. "(CR)" is the correctly-rounded version, and without "(CR)" is the DGEMM equivalent accuracy version. The performance depends on the absolute range of the input values. ϕ varies the range: it increases as ϕ increases. When $\phi = 0.1$, the range is about 1E+9.

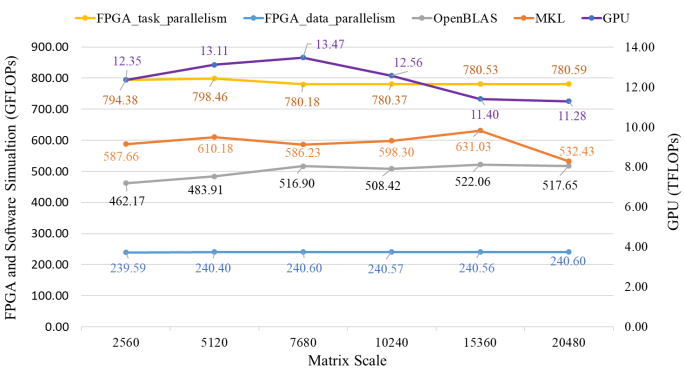
realize tunable accuracy, including correct-rounding, and ensure bit-level reproducibility regardless of the computational environment, even between CPUs and GPUs. We have been developing accurate and reproducible linear algebra kernels using the Ozaki scheme on CPUs and GPUs since 2018 under the collaboration of Prof. Takeshi Ogita (Tokyo Woman's Christian University) and Prof. Katsuhisa Ozaki (Shibaura Institute of Technology) (the development started when the main developer, Daichi Mukunoki worked at Tokyo Woman's Christian University). In this FY2019-2020, we published a paper on accurate and reproducible BLAS routines on CPUs and GPUs based on the Ozaki scheme at PPAM2019 [6]. Also, a part of the results was presented at the poster session in Russian Supercomputing Days 2019 [18], LSPANC [34, 42], SIAM CSE19 [14], ATOS22 [31], and other domesctic seminars [37, 38]. Moreover, in this FY2019-2020, we have developed accurate and reproducible sparse iterative solvers based on the conjugate gradient (CG) method using the Ozaki scheme on CPUs and GPUs. The result was presented at ENUMATH 2019 [10].

4.2.6 DGEMM using Tensor Cores

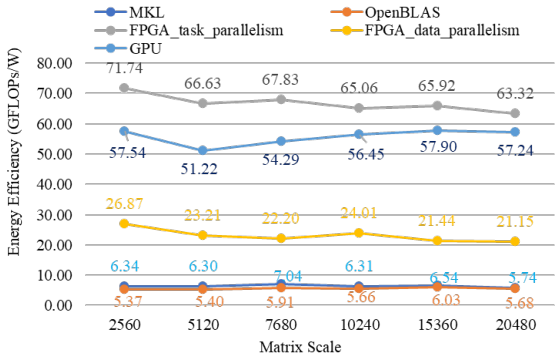
We have developed a double-precision dense matrix multiplication routine (DGEMM) on NVIDIA Tensor Cores based on the Ozaki scheme. Tensor Cores are special processing units that perform 4×4 matrix multiplications on FP16 inputs with FP32 precision and return the result on FP32. We have modified the Ozaki scheme for computing FP64 values using Tensor Cores. This study aims to enhance the potential of Tensor Cores or AI-oriented processors supporting fast low-precision operations for general-purpose workloads. In this FY2019-2020, we first developed a correctly-rounded implementation. The result was presented at the poster session in HPC Asia 2020 [25]. After that, we developed an enhanced implementation that can achieve standard DGEMM equivalent accuracy. The result was submitted as a paper to ISC2020 (which was accepted to publication in LNCS 12151). Figure 4.4 shows the performance of DGEMM using Tensor Cores on a Titan RTX.

4.2.7 FPGA-based matrix multiplier with task parallelism

Matrix multiplication requires computer systems have huge computing capability and data throughput as problem size is increased. In this research, an OpenCL-based matrix multiplier with task parallelism is designed and implemented by using the FPGA board DE5a-NET to improve computation throughput and energy efficiency. The matrix multiplier is based on the systolic array architecture with 10×16 processing elements, and all modules except the data loading modules are autorun to hide computation overhead. When data are single-precision floating-point, the proposed matrix multiplier averagely achieves about 785 GFLOPs in computation throughput (4.5(a)) and 66.75 GFLOPs/W in energy efficiency (4.5(b)). Compared with the Intel's OpenCL example with data parallelism on an FPGA, software simulations with the Intel MKL and OpenBLAS



(a) Computation throughput



(b) Energy efficiency

Figure 4.5: Performance of the FPGA-based matrix multiplier

libraries carried out on a desktop with 32 GB DDR4 RAMs and an Intel i7-6800K processor running at 3.4 GHz, the proposed matrix multiplier averagely outperforms by 3.2 times, 1.3 times, and 1.6 times in computation throughput, and by 2.9 times, 10.5 times, and 11.8 times in energy efficiency, respectively, even if the fabrication technology of the FPGA is 20 nm while it is 14 nm in CPU. Compared with the TITAN V GPU (12 nm), the proposed matrix multiplier is significantly defeated in computing performance, but it wins in energy efficiency. The related results were presented at the conference SIAM PP 2020 [16], ParCo 2019 [3], and ISC 2019 (poster) [17], CYGNUS project [19], and LSPANC2020Jan [44].

4.2.8 FPGA-based acceleration of FDTD sound field rendering

Finite difference time domain (FDTD) schemes are widely applied to analyse sound propagation, but are computation-intensive and memory-intensive as sound space is increased. Current sound field rendering systems with FDTD schemes are mainly based on software simulations on personal computers (PCs) or GPUs. In this research, an accelerator is designed and implemented using FPGA for sound field rendering. Unlike software simulations on PCs and GPUs, the FPGA-based sound field rendering system directly implements wave equations by reconfigurable hardware. Furthermore, a sliding window-based data buffering system is adopted to alleviate external memory bandwidth bottlenecks. Compared to the software simulation carried out on a PC with 128 GB DDR4 RAMs and an Intel i7-7820X processor running at 3.6 GHz, the proposed FPGA-based accelerator takes half of the rendering time and doubles the computation throughput even though the clock frequency of the FPGA system is about 267 MHz and has only 8 GB DDR3 on-board RAMs. The related results were published on the conference USE 2019 [22], DAFx'19 [7], HPC Asia 2020 [24], R-CCS Intl.Sympo [27].

4.2.9 Other activities

In FY2019-2020, the following studies were also conducted, and we published papers:

- Communication-avoiding PCG algorithm on a real-3D CFD code (ScaLA19 [1], R-CCS Intl.Sympo [28]),
- Performance optimization of GPU-based LOBPCG solver (ParCo19 [5]),
- Preliminary work on numerical linear algebra (JLESC [8], and [32]),
- Some of collaborative works with DL4Fugaku project (R-CCS Intl.Sympo [29]),
- Some review or summary talks for numerical libraries on Fugau or other modern systems (JLESC [8, 9], JIFT [12], R-CCS Intl.Sympo [30], other domestic seminars [36, 39]).

4.3 Schedule and Future Plan

The most significant task in FY2019 and FY2020 was to complete the numerical software on the supercomputer Fugaku, and we recognized almost tasks were done in the final stage of development. As reported in the annual

report in FY2018-2019, we have updated the research milestone on a short-term and long-term as well. The must-do-topics should be rementioned again here.

- **Communication avoiding algorithm:**

  We continue to investigate the communication avoiding algorithms and methods in any view of numerical linear algebra. The technologies will be applied on the algorithms of the existing CA-XX linear solvers, both dense and sparse eigenvalue solvers, the multi-dimensional FFT routines, and so on.

- **Precision-and-power aware computing:**

  We have studied high/mixed/reduced-precision numerical software. Recently, reduced-precision computing enforced by deep-learning becomes a significant driving force not only on industry/consumer's market but the HPC community. What is more, other critical issue came from power-saving was pointed out in the annular report last year. We will corporate with the stochastic and approximate approach and arbitrary precision arithmetic with the help of a reconfigurable device to reduce the cost of floating point operations and the total volume of the required hardware as well as energy consumption.

In addition to the above-mentioned issues, various international endeavors have been identified with the completion of the Fugaku project. For example, it is urgent to validate the numerical software developed in different parts of the world and to evaluate the performance of the standard software as the responsibility of the world's leading-edge computer institutions. In particular, the software products developed under the ECP initiatives in the US, SLATE and FFT-X will be actively implemented within the framework of DOE-MEXT, and through joint research agreements between CMU and R-CCS, respectively.

Furthermore, we also need to promote the log-term-ranged fundamental research on numerical algorithms that can be deployed on practical quantum computers, which will be available in ten years from now in the field of quantum computer science. In particular, it is thought to be one of the missions to explore the possibility of numerical computation on post-silicon generation computers by making full use of the emulator on Fugaku and the quantum computer programming environment developed by related organizations.

4.4 Publications

4.4.1 Articles

[1] Y. Ali, N. Onodera, Y. Idomura, T. Ina, and T. Imamura: 'GPU Acceleration of Communication Avoiding Chebyshev Basis Conjugate Gradient Solver for Multiphase CFD Simulations,' Proc. 2019 10th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems (ScalA), Nov. 2019

[2] Daichi Mukunoki, Takeshi Ogita: 'Performance and Energy Consumption of Accurate and Mixed-precision Linear Algebra Kernels on GPUs,' Journal of Computational and Applied Mathematics, Volume 372, (Available online January 2020) July 2020.

[3] Yiyu Tan, Toshiyuki Imamura, and Daichi Mukunoki: 'Design of an FPGA-based Matrix Multiplier with Task Parallelism,' International Conference on Parallel Computing, Prague, Czech, September 2019, Advances in Parallel Computing, Volume 36: Parallel Computing: Technology Trends, pp.241-250, IO-Press, April 2020

[4] Toshiyuki Imamura, Masaaki Aoki, Mitsuo Yokokawa: 'Batched 3D-distributed FFT kernels towards practical DNS codes,' International Conference on Parallel Computing, Prague, Czech, September 2019, Advances in Parallel Computing, Volume 36: Parallel Computing: Technology Trends, pp.169-178, IO-Press, April 2020

[5] Susumu Yamada, Toshiyuki Imamura, Masahiko Machida: 'High performance eigenvalue solver for Hubbard model: Tuning strategies for LOBPCG method on CUDA GPU,' International Conference on Parallel Computing, Prague, Czech, September 2019, Advances in Parallel Computing, Volume 36: Parallel Computing: Technology Trends, pp.105-113, IO-Press, April 2020

[6] Daichi Mukunoki, Takeshi Ogita, Katsuhisa Ozaki: 'Reproducible BLAS Routines with Tunable Accuracy Using Ozaki Scheme for Many-core Architectures,' 13th International Conference on Parallel Processing and Applied Mathematics (PPAM2019), pp 516-527, LNCS, volume 12043, March 2020.

[7] Yiyu Tan, and Toshiyuki Imamura, 'A FPGA-based Accelerator for Sound Field Rendering,' Proc. in the 22nd International Conference on Digital Audio Effects, Birmingham, UK, September 2019.

4.4.2 Oral talks and Poster presentations

[8] Toshiyuki Imamura, and Inge Gutheil: 'Review on standard eigensolvers on a high-end GPU system,' Project talk on 'HPC libraries for solving dense symmetric eigenvalue problems,' 9th JLESC workshop at ICL, Knoxville, TN, US. April 2019.

[9] Toshiyuki Imamura: 'Research advancement in autotuning in libraries and applications,' BOS on Autotuning, 9th JLESC workshop at ICL, Knoxville, TN, US. April 2019.

[10] Daichi Mukunoki, Takeshi Ogita, Katsuhisa Ozaki: 'Accurate and Reproducible CG Method on GPUs,' European Numerical Mathematics and Advanced Applications Conference 2019 (ENUMATH2019), Egmond aan Zee, Oct. 1, 2019.

[11] S. Kudo, and T. Imamura: 'A level-3 BLAS like kernel of the Jacobi rotations for the Jacobi's eigenvalue algorithms,' ParNum2019, Dubrovnik, Oct. 2019.

[12] Toshiyuki Imamura: 'Numerical software on Fugaku', Joint US-Japan Workshop on PostK-ECP Collaboration and JIFT Exascale Computing Collaboration, October 28-29, 2019, at R-CCS, Riken, Kobe, Japan

[13] Toshiyuki Imamura, Daichi Mukunoki, Fabienne Jézéquel, Stef Graillat, Roman Iakymchuk: 'Numerical Reproducibility based on Minimal-Precision Validation,' Computational Reproducibility at Exascale Workshop (CRE2019), in cooperation with SC19, 2019 (extended abstract).

[14] Daichi Mukunoki, Takeshi Ogita: 'High-performance Implementations of Accurate Linear Algebra Kernels on GPUs,' 3rd International Conference on Modern Mathematical Methods and High Performance Computing in Science & Technology (M3HPCST), Jan. 9-11, 2020.

[15] Daichi Mukunoki: 'Minimal-Precision Computing for High-Performance, Energy-Efficient, and Reliable Computations,' SIAM Conference on Parallel Processing for Scientific Computing (PP20), Seattle, Feb. 2020

[16] Toshiyuki Imamura and Yiyu Tan: 'Precision Tuning of the Arithmetic Units in Matrix Multiplication on FPGA,' SIAM Conference on Parallel Processing for Scientific Computing (PP20), Seattle, Feb. 2020

[17] Yiyu Tan, and Toshiyuki Imamura, 'Design of an FPGA-based Matrix Multiplier with Task Parallelism,' Poster presentation at the ISC High Performance Conference, Frankfurt, Germany, June 2019.

[18] Daichi Mukunoki, Takeshi Ogita, and Katsuhisa Ozaki: 'Accurate and Reproducible Linear Algebra Operations for Many-core Architectures,' Poster presentation at Russian Supercomputing Days 2019 (RuSCDays 2019), Sep. 23 - 24, 2019.

[19] Yiyu Tan, Daichi Mukunoki, Toshiyuki Imamura, et al, 'Reduced and Extended-precision Computations on FPGAs and GPUs,' Poster presentation at the 11th Symposium on Discovery, Fusion, Creation of New Knowledge by Multidisciplinary Computational Sciences, Tsukuba, Oct. 2019.

[20] Daichi Mukunoki, Toshiyuki Imamura, Yiyu Tan, Atsushi Koshiba, Jens Huthmann, Kentaro Sano, Fabienne Jézéquel, Stef Graillat, Roman Iakymchuk, Norihisa Fujita, Taisuke Boku: 'Minimal-Precision Computing for High-Performance, Energy-Efficient, and Reliable Computations,' Poster presentation at France-Japan-Germany trilateral workshop: Convergence of HPC and Data Science for Future Extreme Scale Intelligent Applications, Nov. 7, 2019.

[21] Daichi Mukunoki, Toshiyuki Imamura, Yiyu Tan, Atsushi Koshiba, Jens Huthmann, Kentaro Sano, Fabienne Jézéquel, Stef Graillat, Roman Iakymchuk, Norihisa Fujita, Taisuke Boku: 'Minimal-Precision Computing for High-Performance, Energy-Efficient, and Reliable Computations,' SC19 research poster session, The International Conference for High Performance Computing, Networking, Storage, and Analysis, Denver, USA, November 2019.

[22] Tan Yiyu and Toshiyuki Imamura: 'High-order FDTD Method for Room Acoustic Simulation,' the 40th Symposium on Ultrasonic Electronics (USE 2019), Tokyo, November 2019.

[23] Roman Iakymchuk, Fabienne Jézéquel, Stef Graillat, Daichi Mukunoki, Toshiyuki Imamura, Yiyu Tan, et al.: 'Optimizing Precision for High-performance, Robust, and Energy-efficient Computations,' Poster presentation at International Conference on High Performance Computing in Asia-pacific Region (HPC Asia), Fukuoka, January. 2020.

[24] Yiyu Tan and Toshiyuki Imamura: 'Sound Rendering and its Acceleration Using FPGA,' Poster presentation at International Conference on High Performance Computing in Asia-pacific Region (HPC Asia), Fukuoka, January 2020.

[25] Daichi Mukunoki, Katsuhisa Ozaki, Takeshi Ogita, and Toshiyuki Imamura: 'Accurate DGEMM using Tensor Cores,' Poster presentation at International Conference on High Performance Computing in Asia-pacific Region (HPC Asia), Fukuoka, January 2020.

[26] Toshiyuki Imamura, Daichi Mukunoki, Yiyu Tan, Atsushi Koshiba, Jens Huthmann, Kentaro Sano, Fabienne Jézéquel, Stef Graillat, Roman Iakymchuk, Norihisa Fujita and Taisuke Boku: 'Minimal-Precision

Computing for High-Performance, Energy-Efficient, and Reliable Computations,' Poster presentation at the 2nd R-CCS International Symposium, Feb. 2020

[27] Yiyu Tan, Toshiyuki Imamura, and Daichi Mukunoki: 'An FPGA-based Matrix Multiplier with Task Parallelism,' Poster presentation at the 2nd R-CCS International Symposium, Feb. 2020

[28] Yasuhiro Idomura, Takuya Ina, Yussuf Ali, and Toshiyuki Imamura: 'Optimization of Fusion Plasma Turbulence Code GT5D on FUGAKU and SUMMIT,' Poster presentation at the 2nd R-CCS International Symposium, Feb. 2020

[29] Kento Sato, Akiyoshi Kuroda, Kazuo Minami, Jens Domke, Aleksandr Drozd, Mohamed Wahib, Shuhei Kudo, Toshiyuki Imamura, Kiyoshi Kumahata, Keigo Nitadori, Kazuo Ando, and Satoshi Matsuoka: 'DL4Fugaku: Deep learning for Fugaku — Scalability Performance Extrapolation —,' Poster presentation at the 2nd R-CCS International Symposium, Feb. 2020

[30] Toshiyuki Imamura, Yusuke Hirota, and Takuya Ina: 'Re-design of parallel divide and conquer algorithm for a symmetric band matrix,' Poster presentation at the 2nd R-CCS International Symposium, Feb. 2020

[31] 椋木大地: 尾崎スキームに基づく高精度かつ再現性のあるBLASルーチンの実装と自動チューニングの適用, 第22回AT研究会オープンアカデミックセッション（ATOS22）, 東京大学情報基盤センター, 東京都, 2019年5月13日

[32] 工藤周平, 今村俊幸, "オンデマンドな行列計算カーネルの生成機構の構想," 第24回計算工学講演会, 大宮市, 2019年5月.

[33] Toshiyuki Imamura: 'High Precision Floating and Integer Arithmetic on Supercomputing Environment,' Workshop on Largescale Parallel Numerical Computing Technology (LSPANC 2019 June), RIKEN R-CCS, Kobe, Jun. 7, 2019.

[34] Daichi Mukunoki: 'High-Performance Implementations of Accurate and Reproducible BLAS Routines on GPUs,' Workshop on Largescale Parallel Numerical Computing Technology (LSPANC 2019 June), RIKEN R-CCS, Kobe, Jun. 7, 2019.

[35] 工藤周平, 今村俊幸, "ヤコビ回転カーネルを用いたヤコビ固有値計算手法の性能評価", 研究報告HPC, 2019-HPC-170, vol.35, pp.1-8 (2019).

[36] 今村 俊幸: 「「京」ならびに「富岳」に向けた並列固有値計算ソルバについて」, 第4回 High Performance Computing Physics (HPC-Phys) 勉強会, R-CCS, 2019年8月26日

[37] 椋木大地, 荻田武史, 尾崎克久, 今村俊幸: 尾崎スキームによる高精度かつ再現性のあるBLAS実装, 日本応用数理学会2019年年会講演予稿集, pp. 402-403, 2019 (extended abstract).

[38] 椋木大地, 荻田武史, 尾崎克久: 尾崎スキームによる高精度BLAS実装「OzBLAS」とその応用, 第3回 精度保証付き数値計算の実問題への応用研究集会 (NVR 2019), 高松市, 2019年12月1日

[39] 今村俊幸: エクサ時代の非同期タスクを応用した高性能高次元数値線形代数の研究, 第11回 自動チューニング技術の現状と応用に関するシンポジウム(ATTA2019), 東京大学, 2019年12月23日

[40] Daichi Mukunoki: 'Minimal-Precision Computing for High-Performance, Energy-Efficient, and Reliable Computations,' Sapporo Winter HPC Seminar 2020, Information Initiative Center, Hokkaido University, Jan. 24, 2020.

[41] Toshiyuki Imamura: 'Overview of minimal-precision computing and (weak)-numerical reproducibility,' Workshop on Largescale Parallel Numerical Computing Technology (LSPANC 2020 January), RIKEN R-CCS, Kobe, Jan. 30, 2020.

[42] Daichi Mukunoki: 'Accurate BLAS implementations: OzBLAS and BLAS-DOT2,' Workshop on Largescale Parallel Numerical Computing Technology (LSPANC 2020 January), RIKEN R-CCS, Kobe, Jan. 30, 2020.

[43] Shuhei Kudo: 'How (not) to cheat in HPL-AI,' Workshop on Largescale Parallel Numerical Computing Technology (LSPANC 2020 January), RIKEN R-CCS, Kobe, Jan. 30, 2020.

[44] Yiyu Tan: 'Precision Tuning of the Arithmetic Units in Matrix Multiplication on FPGA,' Workshop on Largescale Parallel Numerical Computing Technology (LSPANC 2020 January), RIKEN R-CCS, Kobe, Jan. 30, 2020.

4.4.3 Award

[45] Daichi Mukunoki, Takeshi Ogita, Katsuhisa Ozaki: Best Research Poster Award, Russian Supercomputing Days 2019 (RuSCDays 2019), Sep. 2019 (Accurate and Reproducible Linear Algebra Operations for Many-core Architectures)

4.4.4 Other publication

[46] 「固有値計算と特異値計算, 計算力学レ クチャーコース」一般社団法人 日本計算工学会 編, 長谷川秀彦 今村俊幸 山田進 櫻井鉄也 荻田武史 相島健助 木村欣司 中村佳正 著, 丸善出版, 2019

4.4.5 Software (released as of April 2020)

[47] EigenExa, http://www.r-ccs.riken.jp/labs/lpnctrt/projects/eigenexa/.
[48] KMATH_EIGEN_GEV, http://www.r-ccs.riken.jp/labs/lpnctrt/projects/kmath-eigen-gev/.
[49] KMATH_RANDOM, http://www.r-ccs.riken.jp/labs/lpnctrt/projects/kmath-random/.
[50] KMATHLIB_API, http://www.r-ccs.riken.jp/labs/lpnctrt/projects/kmathlib-api.
[51] KMATH_FFT3D, http://www.r-ccs.riken.jp/labs/lpnctrt/projects/kmath-fft3d/.
[52] ASPEN.K2, http://www.r-ccs.riken.jp/labs/lpnctrt/projects/aspen-k2/.
[53] MUBLAS-GEMV, http://www.r-ccs.riken.jp/labs/lpnctrt/projects/mublas/.