Large-scale Parallel Numerical Computing Technology Research Team

1. Team members

Toshiyuki Imamura (Team Leader)

Takeshi Fukaya (Post Doctoral Researcher)

Yusuke Hirota (Post Doctoral Researcher)

Yoshiharu Ohi (Post Doctoral Researcher)

Daichi Mukunoki (Post Doctoral Researcher)

Daisuke Takahashi (Senior Visiting Researcher)

Franz Franchetti (Visiting Researcher)

Yoshio Okamoto (Visiting Researcher)

Yukiko Akinaga (Assistant)

2. Research Activities

The Large-scale Parallel Numerical Computing Technology Research Team conducts research and development of large-scale, highly parallel and high-performance numerical software for K computer. 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. In order to take advantage of the full potential of K computer, we must select pertinent algorithms and develop a software package by assembling numerical libraries based on the significant concepts of high parallelism, high performance, high precision, resiliency, and scalability. Our primary mission is to develop and deploy highly parallelized and scalable numerical software on K computer, namely KMATHLIB. It comprises several components such as for solving

1) Systems of linear equations,
2) Eigenvalue problems,
3) Singular value decomposition,
4) Fast Fourier transforms, and
5) Nonlinear equations.

The K-specific topics and technical matters for emerging supercomputer systems are also our challenging works as follows;

a) Tofu interconnect,
b) Parallel I/O,
c) Fault detection (soft-error), and
d) Higher accuracy computing.

We are going to complete this project through a tight collaboration among computational science (simulation), computer science (hardware and software), and numerical mathematics. Our final goal is to establish fundamental techniques to develop numerical software libraries for next generation supercomputer systems based on strong cooperation within AICS.

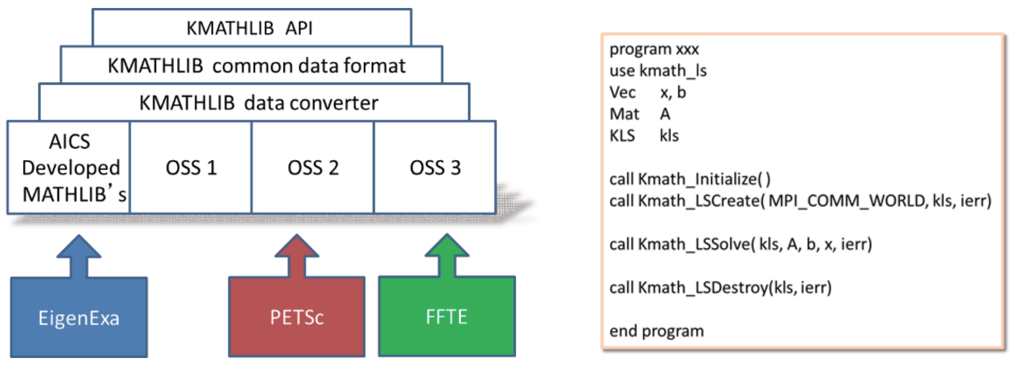
3. Research Results and Achievements

Following published annual reports 2012-13 and 2013-14, we focus on the update and the latest results of running projects, 1) development of KMATHLIB, 2) development of EigenExa, 3) investigation of the FDTD related method, and 4) other fundamental studies for optimization of BLAS kernels by automatic parameter tuning. The plans and the publication list are also presented in the last section.

3.1. KMATHLIB Project

1) Development of KMATHLIB for the integration of OSS packages

Since FY2012-2013, we have developed an integration framework named KMATHLIB for number of numerical libraries installed on K computer. KMATHLIB supports a broad range of spectral of a lot of numerical libraries, and KMATHLIB-API covers the resource usage from hundred to ten thousand nodes or up to the whole system.



```
program xxx
use kmath_ls
Vec     x, b
Mat     A
KLS     kls

call Kmath_Initialize( )
call Kmath_LSCreate( MPI_COMM_WORLD, kls, ierr)

call Kmath_LSSolve( kls, A, b, x, ierr)

call Kmath_LSDestroy(kls, ierr)

end program
```

Figure 1. The structure of the KMATHLIB package (left) and a code example (right)

The left picture in Figure 1 depicts the schematic of KMATHLIB. KMATHLIB-API is on the top layer and is accessed by users directly. Since we designed a flexible plugin mechanism and API’s, favorite OSS can be plugged in like the bottom highlighted part of Figure 1. KMATHLIB-API can conceal the differences of API’s and data structures. The right part of Figure 1 shows an example of KMATHLIB-API. As Figure 1 (right) illustrates, KMATHLIB-API adopts a modern API style of the standard numerical libraries, like PETSc and FFTW. Thus, we only have to use a unique procedure to use the numerical solver plugged in the KMATHLIB package. In this FY, we have updated the plugin mechanism, and it enables users to enhance the KMATHIB library according to their computational environment. Currently, the KMATHLIB library supports LAPACK, ScaLAPACK, EigenExa, PETSc, FFTW, FFTE, and SSL II as built-in libraries.

2) Enhancement of the KMATHLIB numerical libraries

Since FY2012-2013, we also have reviewed the parallel performance of the standard parallel OSS. We have concluded that most of OSS does not scale on a large number of processors (see the former annual reports in FY2012-2013 and FY2013-2014). Improvement of performance and enhancement of functionality is reported in this subsection.

As a part of the KMATHLIB project, we have carried out research on the development of a solver for generalized eigenvalue problems of banded matrices (GEPBs) $A\boldsymbol{x} = \lambda B\boldsymbol{x}$. In our solver, a new divide-and-conquer algorithm for GEPBs that is based on Elsner's idea is employed. The method has two advantages: i) its FLOPS count is smaller than the conventional method if the half bandwidth of the matrices is 3 or less, and ii) the most of the operations can be parallelized efficiently.

In FY2014-2015, we implemented the solver by OpenMP and evaluated the single node performance of our solver and conventional solvers (DSBGVD and DSYGVD in Intel MKL, as an optimized LAPACK implementation). Experiments are carried out on a standard workstation installed with two

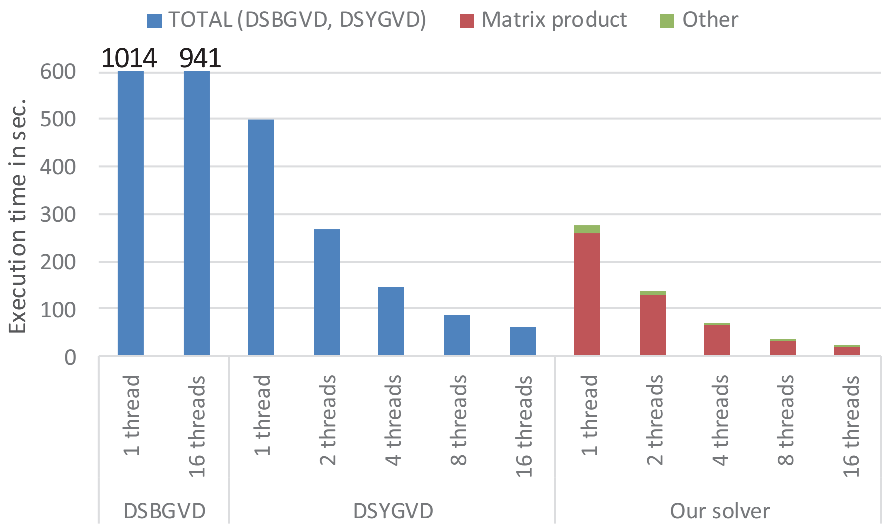


Figure 2. Comparison of the Execution time of solvers; DSBGVD and DSYGVD (from LAPACK), and our proposed solver.

Xeon E5-2660 CPUs (totally 16 cores). Figure 2 shows the execution time to compute all eigenpairs of a GEBP for pentadiagonal matrices (i.e. half bandwidth equals two). Our solver performs 3.2 times faster than the conventional solvers when using 16 threads. The speedup (defined by 'the execution time in the sequential mode' / 'the execution time in the thread-parallel mode') on our solver is higher than that of the conventional solvers. We also have developed a prototype implementation for Intel Xeon Phi and preliminary experiments have been done.

KMATH_RANDOM is a pseudo real random number generator for highly parallel computers such as K computer. KMATH_RANDOM is based on dSFMT, which is an open source code written by Prof. Saito and Prof. Matsumoto in Hiroshima University. KMATH_RANDOM manages random number generators running in each MPI process. On each MPI process, KMATH_RANDOM reads a file called 'jump file' and generates real random numbers independently. KMATH_RANDOM has been released on our team website since December 2014.
In FY2014, we studied the performance bottleneck of KMATH_RANDOM that we reported in last year. Figure 3 shows the execution time and the breakdown (initialization time and generation time of the random numbers) of each process to generate 1 billion random numbers in total with 1,000 MPI processes. Although the generation times are identical in appearance among processes, the initialization times in some nodes are much larger than the median time, and thus it causes the performance degradation of KMATH_RANDOM.

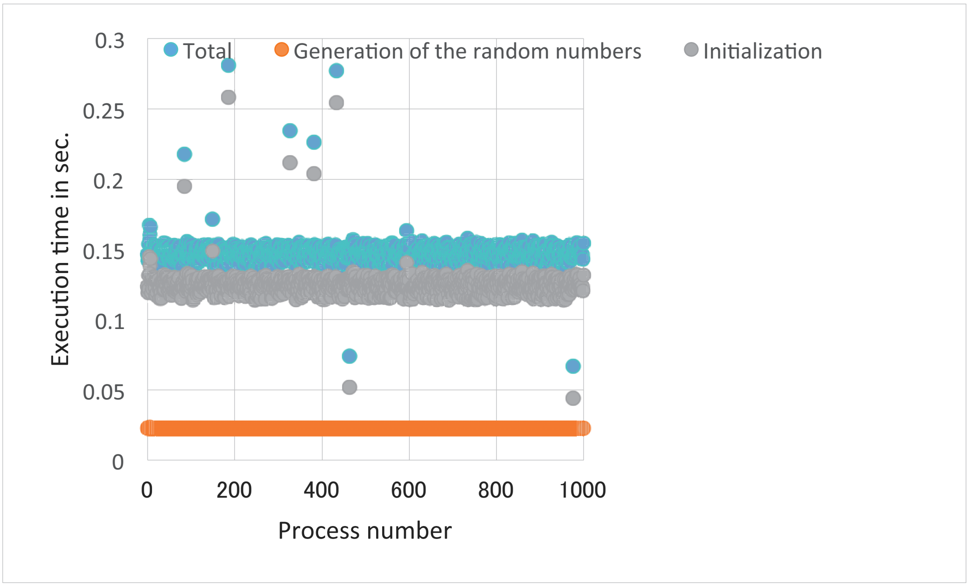


Figure 3. Execution time of each process and its breakdown.

3.2 EigenExa Project

The EigenExa Project is basically a collaborative work between Prof. Yusaku Yamamoto (formerly Kobe University, presently the University of Electro-Communications) granted from 'Development of System Software Technologies for post-Peta Scale High Performance Computing' by JST CREST. We have conducted mainly a part of the development of a high performance parallel dense eigenvalue solver, namely EigenExa. The EigenExa library was released in August 2013. In this FY 2014-2015, we mainly promoted the library and evaluated it on some supercomputer systems. Besides, we have investigated fundamental algorithms in numerical linear algebra such as the QR factorization (or orthonormalization).

1) Detailed performance evaluation of the EigenExa eigensolver

In this FY, we have evaluated the performance of our EigenExa eigensolver in detail, mainly with the focus on the difference between the *eigen_s* routine, which is based on the traditional tridiagonalization, and the *eigen_sx* routine, which employs a novel approach via pentadiagonalization. The latter has advantages in the step of transforming the input matrix into a penta/tridiagonal matrix. However, the latter requires more cost for computing the eigenvalues and eigenvectors of the pentadiagonal matrix by the divide-and-conquer method than the former. In order to clarify the potential of our new approach, it is important to evaluate the performance of both routines and to compare them in detail.

Figure 4 shows the breakdown of the execution time of eigen_s and eigen_sx on the Oakleaf-FX system (at The University of Tokyo). We used its 4,800 nodes and computed all eigenvalues and eigenvectors of a matrix with N=230,000. The graph clearly illustrates that two steps, namely tri/pentadiagonalization and divide-and-conquer steps, are dominant. It also indicates the advantages of eigen_sx (i.e. less flop and communication costs in the pentadiagonalization step) and the disadvantage (i.e. the additional cost in the divide-and-conquer step). By combining these detailed performance results and the expected specifications of post-petascale systems, we could predict the performance of these two routines on upcoming systems, which would strengthen the effectiveness of our new approach.

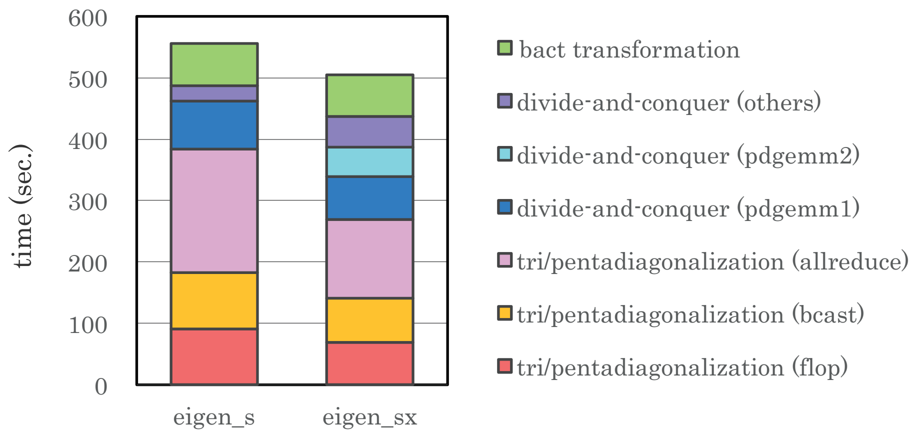


Figure 4. Breakdown of the execution time of eigen_s and eigen_sx. (N=230,000, and using 4,800 nodes of Oakleaf-FX)

2) Studies on the Cholesky QR factorization for tall-skinny QR factorizations

Computing the QR factorization of tall and skinny matrices (tall-skinny QR factorization) is required in blocked subspace projection methods for solving large and sparse eigenvalue problems. The Cholesky QR factorization is an algorithm that computes the QR factorization through the Cholesky factorization of the Gram matrix of the input matrix: $A^\top A \rightarrow R^\top R$. This algorithm consists almost entirely of Level-3 BLAS operations (e.g. matrix-matrix multiplication) and requires only one global reduction in parallel computation. These features mean that Cholesky QR is very suitable for recent large-scale parallel systems such as K computer. However, it is well-known that Cholesky QR has a serious numerical instability: the loss of orthogonality of the computed Q factor grows rapidly as the condition number of the input matrix increases. Therefore, it has been a consensus that Cholesky QR is fast but not practical.

In this FY, through collaboration with researchers in the field of applied mathematics, we have studied an algorithm that simply repeats the Cholesky QR algorithm twice, which we call the CholeskyQR2 algorithm. We have pointed out that CholeskyQR2 is as stable as conventional stable algorithms (e.g. the Householder QR factorization) until the condition number of the input matrix is less than about 10^8 (see Figure 5). In our evaluation on K computer, CholeskyQR2 is still faster than the TSQR algorithm (see Figure 6), which is a stable and communication-avoiding algorithm. These results totally indicate that CholeskyQR2 has a remarkable potential to be used in practical computations.

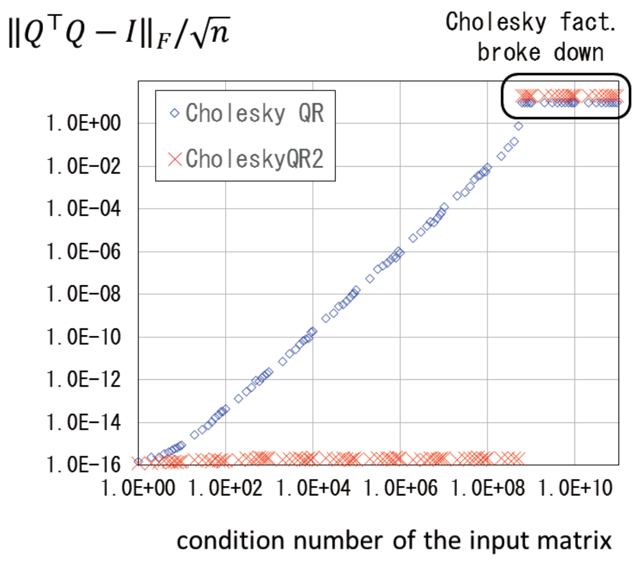


Figure 5. Numerical stability of CholeskyQR2.

(m=2,097,152, n=64, and #processes=8,192 on Oakleaf-FX system)

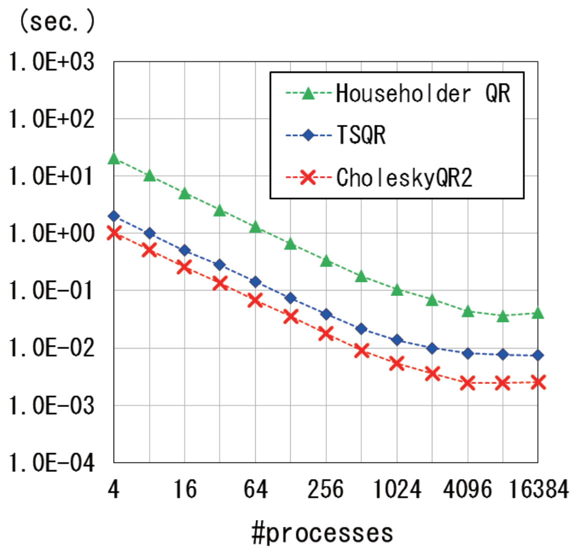


Figure 6. Parallel performance of CholeskyQR2 on the K computer.

(m=4,194,304, and n=64)

3.3 Investigation of the FDTD Related Methods

1) FDTDM (finite-difference time-domain method)

Recently, an electric power problem is one of the important issues for industrial and environmental engineering communities. A large-scale numerical simulation reveals various unknown factors. It is widely known that the finite-difference time-domain method (FDTDM) is applied for numerical simulations of electromagnetic wave propagation phenomena. Since most of the preexistent FDTDM software is commercial (not free software), it is quite hard to modify the simulation code to analyze a phenomenon by new approaches. Thus, we recognized that it is one of the key issues to develop open source software based on FDTDM for K computer. In FY2014-2015, we surveyed papers in computational electromagnetics.

2) MTDM (meshless time-domain method)

In FDTDM, nodes of electric and magnetic fields are arranged based on an orthogonal mesh like a staggered mesh. Hence, it has restrictions in practice when we treat a complex shaped domain. On the other hand, the radial point interpolation method (RPIM), which is one of the meshless methods, can be theoretically applied to an arbitrarily shaped domain. The meshless time-domain method (MTDM) based on RPIM has useful features inherited from FDTMD and RPIM. However, we have open issues about the relation between two vital functions, the weight function and the shape function, which appear in MTDM. For example, the most suitable weight function to generate the shape function and appropriate arrangement of the nodes of electric and magnetic field must be clarified. In FY2014-2015, we have evaluated the transmission loss of electromagnetic wave propagation in a straight waveguide, numerically. In the simulation, we varied the weight function with a multi-quadratic, a reciprocal of the multi-quadratic, a quadratic spline, and an exponential function. As a result, we obtained that the quadratic spline function is the most suitable for MTDM. Also, we improved the acceleration method (we proposed originally in FY2013-2014) for the generation of shape functions. Two things are essential for the improvement, i) enlargement of the computational domain in which shape functions are reused, ii) the reduction of the number of times of shape function generations. The modified method performs excellently on a multicore processor, and we confirmed up to 100x speedup compared with the conventional method.

3.4 A study for development of multifunctional linear algebra libraries on future architectures

Traditional linear algebra libraries such as Basic Linear Algebra Subprograms (BLAS) and Linear Algebra PACKage (LAPACK) are still essential building blocks for computational science. However, nowadays they are required not only to achieve high performance with scalability, but also to support accurate, fault-tolerant, and energy-efficient computations toward the Exa-scale computing. We are conducting a study for the development of multifunctional linear algebra libraries dealing with these issues by utilizing some technologies such as auto-tuning and mixed precision calculation (including higher-precision compared to the hardware supported precision).

1) Implementation of fast GEMV routines with online auto-tuning on GPUs

We assume that auto-tuning is one of the most prominent technologies for the efficient development of high performance multifunctional numerical libraries on complex architectures such as many-core processors. We proposed a sophisticated implementation of GEMV, a BLAS routine for computing dense matrix-vector multiplication, for NVIDIA GPUs. Some existing implementations have performance fluctuations depending on the input matrix size. On the other hand, our implementation achieved both higher throughput and better performance stability with respect to entry size of the matrix on multiple GPU architectures (Fermi, Kepler, and Maxwell architectures) by introducing an auto-tuning method that determines the optimal thread block size for the GEMV kernel. Our auto-tuning is "online" auto-tuning, which does not need trial executions to determine the tuning parameters by using a theoretical performance model. In the coming year, we will extend the experiences in this work on GPUs to other many/multi-core architectures such as Intel Xeon Phi and K computer.

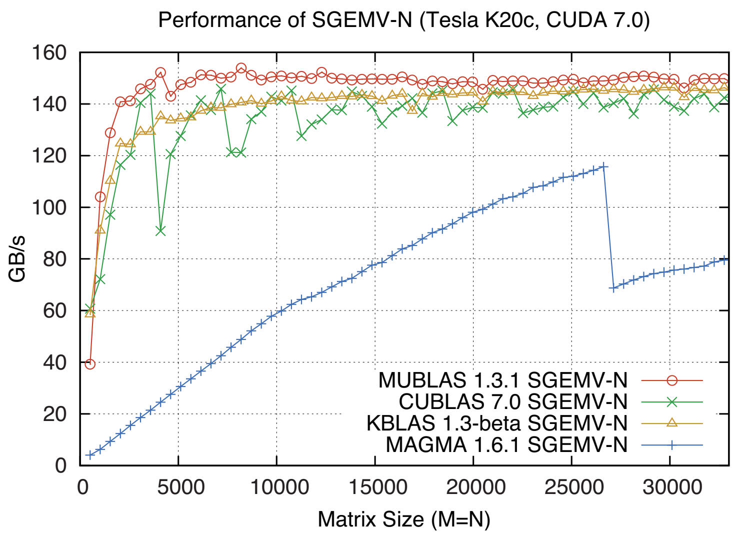


Figure 7. Performance comparison of SGEMV-N (single-precision GEMV for non-transposed matrix) on an NVIDIA Tesla K20c (Kepler architecture GPU). MUBLAS 1.3.1 (red-line) is our implementation.

2) *Acceleration of DGEMM with software-emulated double-precision floating-point arithmetic on Maxwell architecture GPUs*

In 1971, Dekker proposed double length floating-point arithmetic to compute a floating-point arithmetic with 2N length significand using a floating-point arithmetic with N length significand. The method is often used for accurate computation and is also known as double-double arithmetic by Bailey to compute quadruple-precision arithmetic using double-precision arithmetic. We used this method to accelerate DGEMM, a BLAS routine for computing double-precision dense matrix multiplication, on the NVIDIA Maxwell architecture GPU, the double-precision performance of which is 1/32 of single-precision. As a result, we showed that the DGEMM with the software-emulated double-precision arithmetic outperforms the vendor implementation computed in hardware FPU on the GPU.

3) Development of parallel quadruple-precision BLAS

We are developing some quadruple-precision numerical libraries using double-double arithmetic for accurate computation on K computer. P-QPBLAS is the parallel version of QPBLAS, a quadruple-precision BLAS implementation using double-double arithmetic by Japan Atomic Energy Agency (JAEA). The prototype version of the P-QPBLAS has been implemented in the past year, and we are going to continue the development to release it in the future. However, there remain some issues in terms of the performance, and we recognize that some refactoring tools should be introduced to reduce the development cost.

4. Schedule and Future Plan

Our three main projects, KMATHLIB, EigenExa, and FDTD are still on-going works.

1) KMATHLIB project

To promote the KMATHLIB package, we recognize to support much useful plugged-in OSS. We plan to extend plugin solvers for, such as sparse linear equations, GEBPs, SVD for tensors. Urgent work related to this report is about GEBPs solver on distributed highly parallel computers such as the K computer, and we will soon release it as a part of KMATHLIB in FY2015-FY016. Furthermore, we also plan to release the solver for many core accelerators such as Intel Xeon Phi in FY2015 or later. Improvement of the performance of the generalized eigenvalue solver for dense matrix and random number generator, which are already released, must be our principal task in FY2015-2016.

2) EigenExa project

After the first release of the EigenExa library, several application codes adopt to use EigenExa. We got another important mission to maintain EigenExa continually. Furthermore, we recognize that it is also significant to improve the performance and scalability. Introducing the Communication Avoiding, Communication Hiding, and Synchronous Avoiding are prominent technologies. Also, a lightweight or flexible implementation is also an important issue.

3) FDTD related method

Our MUST topics for the FDTD related project are to support a precise 3D simulation by using MTDM. In fact, to simulate side-effects of an MRI device to a human body have a significant impact on life science and medical engineering. We need to promote the FDTD related works, especially MTDM, which has been originally studied in our project. It shall enhance the research field in AICS.

4) Other issues

As we described from the startup of this project team, “Fault tolerance” or “Resilience” is one of the key words for the peta-scale computing. Since K computer is very stable and not often halt due to the system failure, technical review, and feasibility studies are proceeding on other systems than K computer. From the viewpoint of the numerical library, we will establish a new algorithmic fault detection mechanism and its framework.

5. Publication, Presentation, and Deliverables

(1) Journal Papers

1. Yasuhiro Idomura, Motoki Nakata, Susumu Yamada, Masahiko Machida, Toshiyuki Imamura, Tomohiko Watanabe, Masanori Nunami, Hikaru Inoue, Shigenobu Tsutsumi, Ikuo Miyoshi, and Naoyuki Shida, "Communication-overlap techniques for improved strong scaling of gyrokinetic Eulerian code beyond 100k cores on the K-computer", International Journal of High Performance Computing Applications, Vol. 28, No. 1, pp. 73-86, 2014
2. Takemasa Miyoshi, Keiichi Kondo, and Toshiyuki Imamura, "The 10,240-member ensemble Kalman filtering with an intermediate AGCM", Geophysical Research Letters, Vol. 41, July 28 2014
3. Teruo Tanaka, Ryo Otsuka, Akihiro Fujii, Takahiro Katagiri, and Toshiyuki Imamura, "Implementation of d-Spline-based incremental performance parameter estimation method with ppOpen-AT", Scientific Programming, Vol. 22, No. 4, pp. 299-307, 2014

(2) Conference Papers

(Reviewed)

1. Chongke Bi, Kenji Ono, Kwan-Liu Ma, Haiyuan Wu, and Toshiyuki Imamura, "A Study of Parallel Data Compression Using Proper Orthogonal Decomposition on the K Computer", Proc. Eurographics Symposium on Parallel Graphics and Visualization (EGPGV2014), Swansea, UK, June 9-10, 2014.
2. Takeshi Fukaya, Toshiyuki Imamura, and Yusaku Yamamoto, "Performance Analysis of the Householder-type Parallel Tall-Skinny QR Factorizations toward Automatic Algorithm Selection", Proc. the Ninth International Workshop on Automatic Performance Tuning (iWAPT2014), pp.1-8, Eugene, USA, July 1, 2014
3. Takeshi Fukaya, Yuji Nakatsukasa, Yuka Yanagisawa, and Yusaku Yamamoto, "CholeskyQR2: A Simple and Communication-Avoiding Algorithm for Computing a Tall-Skinny QR Factorization on a Large-Scale Parallel System", Proc. the 5th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems (ScalA), pp.31-38, New Orleans, USA, November 17, 2014
4. Daichi Mukunoki, Toshiyuki Imamura, and Daisuke Takahashi, "Fast Implementation of

General Matrix-Vector Multiplication (GEMV) on Kepler GPUs", Proc. the 23rd Euromicro International Conference on Parallel, Distributed and Network- based Processing (PDP 2015), pp. 642-650, Turku, Finland, March 4, 2015

(Non-reviewed)

1. Takahiro Katagir, Koichi Takayama, Takashi Yonemura, Hiroki Kumahora, Mitsuyoshi Igai, Junichi Kitagami, Yoshiyuki Eguchi, Takeshi Fukaya, Yusaku Yamamoto, Junichi Iwata, Kazuyuki Uchida, Satoshi Oshima, and Kengo Nakajima, "Application of the communication-avoiding algorithm CAQR to the orthogonalization process in RSDFT and its evaluation", IPSJ SIG Tech. Rep. [High Performance Computing], Vol.2014-HPC-144, No.3, pp.1-6, May 19, 2014 (in Japanese)
2. Takeshi Fukaya and Toshiyuki Imamura, "A study on auto-tuning for the structured QR factorization appearing in the TSQR algorithm", the 19th Annual Meeting of Japan Society for Computational Engineering and Science, Proc. JSCES, Vol.19, 2014 (in Japanese)
3. Toshiyuki Imamura, Yusuke Hirota, Takeshi Fukaya, Susumu Yamada, and Masahiko Machida, "Communication Avoiding (CA) and Communication Hiding (CH) in a dense-matrix eigenvalue problem", the 19th Annual Meeting of Japan Society for Computational Engineering and Science, Proc. JSCES, Vol.19, 2014 (in Japanese).
4. Yoshiharu Ohi, Taku Itoh and Soichiro Ikuno, "Speedup for generation of shape function in Meshless Time Domain Method", the 19th Annual Meeting of Japan Society for Computational Engineering and Science, Proc. JSCES, Vol.19, 2014 (in Japanese).
5. Toshiyuki Imamura, Daichi Mukunoki, Susumu Yamada, and Masahiko Machida, "Implementation and evaluation of CUDA-xSYMV", IPSJ SIG Tech. Rep. [High Performance Computing], Vol. 2014-HPC-146, No. 14, pp.1-12, September 25, 2014 (in Japanese)
6. Daichi Mukunoki, and Toshiyuki Imamura, "Implementation and Evaluation of DGEMM using Double-Float Arithmetic on Maxwell Architecture GPUs", IPSJ SIG Tech. Rep. [High Performance Computing], Vol. 2014-HPC-147, No. 26 pp. 1-6, December 2014 (in Japanese)
7. Toshiyuki Imamura, Daichi Mukunoki, Susumu Yamada, and Masahiko Machida, "Performance evaluation of the fastest GPU-based eigensalue solver by selecting CUDA-BLAS's", IPSJ SIG Tech. Rep. [High Performance Computing], Vol. 2015-HPC-148, No.4 pp.1-9, February 23, 2015 (in Japanese)

(3) Invited Talks

1. Toshiyuki Imamura, "The EigenExa Library – High Performance & Scalable Direct

Eigensolver for Large-Scale Computational Science", International Supercomputing Conference (ISC14), HPC in Asia session, June 26, 2014

2. Takeshi Fukaya, "The Cholesky QR factorization in high-performance computing", the 12th Computational Mathematics Conference, Dec. 28, 2014 (in Japanese)

(4) Technical reports, Posters and Presentations

(Technical Reports)

1. Takeshi Fukaya and Toshiyuki Imamura, "Performance evaluation of the EigenExa eigensolver by using the 4800 nodes of the FX10 system", Super Computing News, The University of Tokyo, Vol. 16, No. 3, pp. 20-27, May 2014 (in Japanese)
2. Takeshi Fukaya, "Performance evaluation of a communication-avoiding QR factorization by using the 4800 nodes of the FX10 system", Super Computing News, The University of Tokyo, Vol. 16, No. 4, pp. 11-20, July 2014 (in Japanese)

(Posters)

1. Yoshiharu Ohi, Taku Itoh, and Soichiro Ikuno, "Numerical Investigations of 3D Electromagnetic Wave Propagation Phenomena Using Meshless Time Domain Method", the 16th Biennial IEEE Conference on Electromagnetic Field Computation (CEFC2014), May 27, 2014.
2. Daichi Mukunoki, Toshiyuki Imamura, and Daisuke Takahashi, "Fast Implementation of SGEMV on Kepler Architecture GPUs", GTC Japan 2014, Tokyo, Japan, July 16, 2014 (in Japanese)
3. Takeshi Fukaya, "QR factorizations based on the Cholesky factorization", the 8th Kobe University Cooperative Division Symposium, Kobe Japan, September 11, 2014 (in Japanese)
4. Yusuke Hirota, "On Decomposition of Perturbation Matrices in Divide and Conquer Method for Generalized Eigenvalue Problems of Banded Matrices", the 8th Kobe University Cooperative Division Symposium, Kobe, Japan, September 11, 2014 (in Japanese)
5. Yoshiharu Ohi, and Soichiro Ikuno, "Numerical Investigation of Electromagnetic Wave Propagation Phenomena using 3D Meshless Time Domain Method", the 24th International Toki Conference (ITC-24), Toki, Japan, November 6, 2014
6. Takeshi Fukaya, "Modeling the performance of parallel dense eigensolvers on peta/post-petascale systems", JST/CREST International Symposium on Post Petascale System Software (ISP2S2), Kobe, Japan, December 2, 2014
7. Takeshi Fukaya, and Toshiyuki Imamura, "Performance evaluation of the EigenExa dense eigensolver on the K computer", the 5th AICS International Symposium, Kobe, Japan, December 8-9, 2014

8. Yusuke Hirota, and Toshiyuki Imamura, “Development of High Performance Parallel Real Random Number Generator KMATH_RANDOM”, the 5th AICS International Symposium, Kobe, Japan, December 8-9, 2014
9. Toshiyuki Imamura, Daichi Mukunoki, Narimasa Sasa, Susumu Yamada, and Masahiko Machida, “Pseudo-Quadruple precision extended mathematics Package, QP-Pack”, Annual Meeting on Advanced Computing System and Infrastructure (ACSI 2015), Tsukuba, Japan, January 26-28, 2015 (in Japanese)
10. Daichi Mukunoki, Toshiyuki Imamura, and Daisuke Takahashi, “Fast Implementation of SGEMV with Matrix Size Independent Performance on Kepler and Maxwell Architecture GPUs”, Annual Meeting on Advanced Computing System and Infrastructure (ACSI 2015), Tsukuba, Japan, January 26-28, 2015 (in Japanese)
11. Shin’ichi Sasaki, Akihiro Fujii, Teruo Tanaka, Daichi Mukunoki, and Toshiyuki Imamura, “Acceleration of Quadruple precision arithmetic on K supercomputer system”, Annual Meeting on Advanced Computing System and Infrastructure (ACSI 2015), Tsukuba, Japan, January 26-28, 2015 (in Japanese)
12. Yoshiharu Ohi, Yoshihisa Fujita, Taku Itoh, and Soichiro Ikuno, “Numerical Investigation of Influence of Node Alignment on Stable Calculation for Meshless Time Domain Method”, SIAM Conference on Computational Science and Engineering (CSE15), Salt Lake City, USA, March 14-18, 2015.
13. Daichi Mukunoki, ToshiyukiImamura, and DaisukeTakahashi, “High-Performance GEMV and SYMV with Auto-Tuning for Performance Stabilization on Multiple GPU Generations”, GPU Technology Conference (GTC 2015), San Jose, USA, March 17, 2015
14. Toshiyuki Imamura, “The EigenExa library: dense eigenvalue solver for post-petascale computing”, SSExa2015, Greifswald, Germany, March 22-25, 2015

(Oral presentations)

1. Takeshi Fukaya, “Communication-avoiding QR factorizations and related auto-tuning”, ATOS9, Tokyo, Japan, May 5, 2014 (in Japanese)
2. Toshiyuki Imamura, Yusuke Hirota, Takeshi Fukaya, Susumu Yamada, and Masahiko Machida, “EigenExa: high performance dense eigensolver, present and future”, the 8th International Workshop on Parallel Matrix Algorithms and Applications (PMAA2014), Lugano, Switerland, July 2-4, 2014
3. Yusuke Hirota and Toshiyuki Imamura, “Acceleration of Divide and Conquer Method for Generalized Eigenvalue Problems of Banded Matrices on Manycore Architectures”, the 8th International Workshop on Parallel Matrix Algorithms and Applications (PMAA2014), Lugano, Switzerland, July 2-4, 2014.

4. Takeshi Fukaya, Yusaku Yamamoto and Toshiyuki Imamura, “An study on blocking techniques of Householder transformations and related communication-avoiding”, Summer united Workshop on Parallel, distributed and cooperative Processing (SWoPP2014), Niigata, Japan, July 28, 2014 (in Japanese)
5. Takeshi Fukaya, Yuji Nakatsukasa, Yuka Yanagisawa and Yusaku Yamamoto, “Performance evaluation of the Cholesky QR factorization with reorthogonalization on large-scale parallel systems”, 2014 JSIAM Annual meeting, Tokyo, Japan, September 3, 2014 (in Japanese)
6. Yoshiharu Ohi and Soichiro Ikuno, “Analysis of node arrangement on computational accuracy and numerical stability using Meshless Time Domain Method”, 2014 JSIAM Annual meeting, Tokyo, Japan, September 3, 2014. (in Japanese)
7. Yuka Yanagisawa, Yuji Nakatsukasa, Takeshi Fukaya, Yusaku Yamamoto, Shinichi Oishi and Kannan Ramaseshan, “Shifted Cholesky QR factorization”, 2014 JSIAM Annual meeting, Tokyo Japan, September 4, 2014 (in Japanese)
8. Takuma Kawamura, Yasuhiro Idomura, Hiroko Miyamura, Hiroshi Takemiya, and Toshiyuki Imamura, “Remote visualization of large-scale simulation results on K-computer using particle-based volume rendering”, Annual meeting on Atomic Energy Society of Japan, Kyoto, Japan, September 8, 2014 (in Japanese)
9. Yoshiharu Ohi, and Soichiro Ikuno, “Influence of Weight Function to Numerical Precision in 3D Meshless Time Domain Method”, the 33rd JSST Annual Conference (JSST2014), Kita-kyushu, Japan, October 29, 2014.
10. Toshiyuki Imamura, Yusuke Hirota, Takeshi Fukaya, Susumu Yamada, and Masahiko Machida, “Post Peta-scale dense eigenvalue solver”, JST/CEST International Symposium on Post Petascale System Software, Kobe, Japan, December 3-4, 2014
11. Yoshiharu Ohi and Soichiro Ikuno, “Stability analysis of Meshless Time-Domain Method using Arbitrary node arrangement”, the 23th MAGDA conference (MAGDA2014), Takamatsu, Japan, December 4, 2014. (in Japanese)
12. Toshiyuki Imamura, “Review on large-scale dense eigenvalue computation”, 18th Symposium on Setouchi-rim JSIAM Local Research Group, Kurashiki, Japan, December 6, 2014 (in Japanese)
13. Yoshiharu Ohi, Taku Itoh, and Soichiro Ikuno, “Evaluation of Node Arrangement Based on Finite Element in Meshless Time-Domain Method”, Nonlinearity/Visualization-Section Meeting in Workshop on developments of new simulation methods for plasma-wall interaction 2014, Toki, Japan, January 26, 2015. (in Japanese)
14. Takeshi Fukaya and Toshiyuki Imamura, “Performance evaluation of the EigenExa eigensolver on the Oakleaf-FX supercomputing system”, Annual Meeting on Advanced

Computing System and Infrastructure (ACSI 2015), Tsukuba, Japan, January 27, 2015 (in Japanese, reviewed)

15. Takeshi Fukaya, "CholeskyQR2: an algorithm of the Cholesky QR factorization with reorthogonalization", 2015 Conference on Advanced Topics and Auto Tuning in High-Performance and Scientific Computing (2015 ATAT in HPSC), Taipei, Taiwan, February 27-28, 2015
16. Toshiyuki Imamura, "Automatic-tuning for CUDA-BLAS kernels parameter by multi-stage d-Spline", 2015 Conference on Advanced Topics and Auto Tuning in High-Performance and Scientific Computing (2015 ATAT in HPSC), Taipei, Taiwan, February 27-28, 2015
17. Takeshi Fukaya and Toshiyuki Imamura, "Performance Evaluation of EigenExa Dense Eigensolver on the Oakleaf-Fx Supercomputer System", SIAM Conference on Computational Science and Engineering (CSE15), Salt Lake City, USA, March 14, 2015
18. Toshiyuki Imamura, Takeshi Fukaya, Yusuke Hirota, Susumu Yamada and Masahiko Machida, "Numerical Eigenvalue Engine towards Extreme-scale Computing Era", SIAM Conference on Computational Science and Engineering (CSE15), Salt Lake City, USA, March 18, 2015
19. Yusuke Hirota, "Present State of Eigenvalue Solvers and Prospects for Post Peta-scale Systems", Cyber HPC Symposium, Suita, Japan, March 20, 2015 (in Japanese)

(5) Patents and Deliverables

Currently, we have released five OSS from our site as follows.

1. EigenExa, http://www.aics.riken.jp/labs/lpnctrt/EigenExa_e.html
2. KMATH_EIGEN_GEV,
   http://www.aics.riken.jp/labs/lpnctrt/KMATH_EIGEN_GEV_e.html
3. KMATH_RANDOM, http://www.aics.riken.jp/labs/lpnctrt/KMATH_RANDOM_e.html
4. ASPEN.K2, http://www.aics.riken.jp/labs/lpnctrt/ASPENK2_e.html
5. MUBLAS-GEMV, http://www.aics.riken.jp/labs/lpnctrt/ASPENK2_e.html