jh170011-DAJ

High-performance Randomized Matrix Computations for Big Data Analytics and Applications

Takahiro Katagiri (Nagoya University)

Abstract
In this project, we evaluate performance of randomized matrix algorithm for big data analysis with several kinds of supercomputers with providing JHPCN. Target of the algorithm is singular value decomposition (SVD) and related eigenproblem. In this year, we focus on evaluating of prototyping for SVD. The integrated SVD (iSVD), which is a new algorithm by members of this project, is evaluated with provided supercomputer resources.

1. Basic Information

(1) Collaborating JHPCN Centers

This research is planned to use multiple supercomputer resources in the centers of JHPCN. The consideration for division of joint research are thus covered. Necessity of multiple supercomputer in the proposal is summarized as follows.

- GPUs for TSUBAME2.5
  - Implementation of parallelism on large-scale cluster for SVD (singular value decomposition) and LS (Linear System)
  - 1-CPU-1-GPU; multiple-CPU-Multiple-GPU;
- Multiple-node: CPU cluster (MPI + OpenMP);
  - Advanced CPUs for the CX400, the FX10 and the FX100.
- Performance comparison with the FX10 and the FX100 with respect to hardware improvements, such as bandwidths for memory accesses.
- Supercomputer adaptation for SVD and LS.
- Performance evaluation and profiling with Fujitsu performance profiler.
- Evaluating auto-tuning effects.

The necessity for implementing the project as a JHPCN joint research project are two-folded.

First, the project aims at dealing with very large-scale numerical computations and data processing. Supercomputers offered by the JHPCN are critical and extremely necessary. On with the accesses of these supercomputers, we can (i) deploy the proposed random sketching algorithms in a large-scale parallel environments, (ii) develop and study auto-tuning schemes of the computer codes for these advanced supercomputers to achieve high computing capabilities, and (iii) handle the very large-scale data sets which are generated from important and timely scientific, engineering, social network applications.

Second, the project can be conducted only if researchers with various expertise work together. A strong international team has been formed to conduct this project. The team members include experts in high-performance computations, numerical linear algebra, statistical analysis and computations. The support from JHPCN is essential to establish the collaborations of these researches to achieve the goals of the project.

(2) Research Areas

- ☐ Very large-scale numerical computation

- ☑ Very large-scale data processing
- ☐ Very large capacity network technology
- ☐ Very large-scale information systems

(3) Roles of Project Members

Takahiro Katagiri researches large scale implementation, and adaptation of auto-tuning.

Weichung Wang researches parallel algorithm development, surrogate-assisted turning, and big data applications.

Su-Yun Huang researches random sketching algorithm development, mathematical and statistical analysis.

Kengo Nakajima provides knowledge of high performance implementation of iterative methods.

Osni Marques provides knowledge of parallel eigenvalue algorithms. He provides knowledge of SVD and implementation of numerical libraries.

Feng-Nan Hwang provides knowledge of parallel eigenvalue algorithms. He provides knowledge of SVD and its parallel implementation.

Toshio Endo provides knowledge of optimizations for hierarchical memory and adaptation of its auto-tuning. He provides knowledge of hierarchical memory optimization.

2. Purpose and Significance of the Research

This project aims at developing random sketching algorithms with high-performance implementations on supercomputers to compute SVD and LS solutions of very large-scale matrices. Consequently, the resulting algorithms and packages can be used to solve important and demanding problems arising in very large-scale numerical computations and data processing that remain challenges nowadays.

Matrix computation is one of the most important computational kernels in many applications of numerical computations and data processing. Because matrix representations are almost everywhere in scientific simulations, engineering innovations, data analysis, statistical inferences, and knowledge extractions, just to name a few, these timely applications lead to strong needs of efficient parallel numerical solvers based on matrices. However, few numerical solvers, especially randomized algorithms, are designed to tackle very large-scale matrix computations on the latest supercomputers. This project plans to fill the gap by focusing on algorithmic and software aspects.

In the algorithmic side, we intend to develop efficient sketching schemes to compute approximate SVD and LS solutions of large-scale matrices. The main idea is to sketch the matrices by randomized algorithms to reduce the computational dimensions and then suitably integrate the sketches to improve the accuracy and to lower the computational costs. Such techniques will benefit many scenarios that low-rank SVD approximation or approximate LS solutions with quick sketches are sufficient. Our schemes will also benefit the applications that it is not possible to compute full SVD or highly-accurate LS solutions provided the matrices are very-large or the matrices are provided in the form of streaming.

In the software aspect, we intend to implement the proposed algorithms on supercomputers. Consequently, the software packages can be used to solve very large-scale real world problems and advance the frontier of science and technology innovations. One essential component of this project is to develop effective automatic software auto-tuning (AT) technologies, so that the package can fully take advantage of the computational capabilities of the target supercomputers that include CPU homogeneous and CPU-GPU heterogeneous parallel computers.

3. Significance as a JHPCN Joint Research Project

The significance is summarized as follows:

(1) Novel and efficient randomized sketching algorithms: We anticipate the international interdisciplinary joint efforts in this project will lead to highly efficient randomized sketching algorithms for computing very large-scale singular value decompositions and for solving large-scale linear systems. We have observed that the size of matrices increases rapidly nowadays. These large matrices arising in, for example, finer meshes in discrete differential equations, larger number of sensors, and collections of activities on the Internet. To tackle these large-scale problems, random sketching is one of the most powerful approach. Our integration type algorithms are expected to improve the accuracy and accelerate the computation on parallel computers.

(2) High-performance scalable software packages with auto-tuning mechanisms: The proposed algorithms are intended to be implemented on the JHPCN supercomputers. While the data size keeps increasing rapidly, fortunately, the number of computing nodes also keeps increasing and more and more recent parallel computers are equipped with many-core co-processors. It is thus essential to develop scalable algorithms to take advantage of such latest computer architectures for tackling very large-scale problems. On the other hand, thanks to the auto-tuning technologies to be studied and deployed within the software, we expect the software can fully take advantage of the computational capabilities of the target supercomputers.

(3) Implications of applications with large-scale data sets: We expect the resulting algorithms and software packages will impact several important big data applications. Singular value decomposition of matrices and solutions of linear systems are two essential components of big data analytics and various practical applications. These applications include unsupervised dimension reduction (e.g. matrix factorization, principal component analysis, spectral clustering) and supervised dimension reduction (e.g. linear discriminant analysis, canonical correlation analysis, inverse regression), machine learning, numerical simulations in geophysics, energy, human genetic variation, and many others.

4. Outline of the Research Achievements up to FY2016

- Randomized Algorithm (A Prototyping)

Fig. 1 shows the algorithm of Integrated SVD with multiple sketches (iSVD)

We have parallelized iSVD with MPI. The line 3 in Fig. 1 can be parallelized easily. We need to distribute $\Omega_{[i]}$ to each process in the line 1 by using MPI_Alltoall. We also need to gather distributed $Q_{[i]}$ by using MPI_gather. In this version, matrix A is not distributed.

Require: Input $\boldsymbol{A}$ (real $m \times n$ matrix), k (desired rank of approximate SVD), p (oversampling parameter), $\ell = k + p$ (dimension of the sketched column space), q (power of projection), N (number of random sketches)
Ensure: Approximate rank-k SVD of $\boldsymbol{A} \approx \widehat{\boldsymbol{U}}_k \widehat{\boldsymbol{\Sigma}}_k \widehat{\boldsymbol{V}}_k^\top$
1: Generate $n \times \ell$ random matrices $\boldsymbol{\Omega}_{[i]}$ for $i = 1, \ldots, N$
2: Assign $\boldsymbol{Y}_{[i]} \leftarrow (\boldsymbol{A}\boldsymbol{A}^\top)^q \boldsymbol{A}\boldsymbol{\Omega}_{[i]}$ for $i = 1, ..., N$ with $\boldsymbol{\Omega}_{[i]} = \boldsymbol{\Omega}_{gp}$ or $\boldsymbol{\Omega}_{cs}$ (in parallel)
3: Compute $\boldsymbol{Q}_{[i]}$ whose columns are orthonormal basis of $\boldsymbol{Y}_{[i]}$ (in parallel)
4: Integrate $\overline{\boldsymbol{Q}} \leftarrow \{\boldsymbol{Q}_{[i]}\}_{i=1}^N$ (by Algorithm 3 or Algorithm 4)
5: Compute SVD of $\overline{\boldsymbol{Q}}^\top \boldsymbol{A} = \widehat{\boldsymbol{W}}_\ell \widehat{\boldsymbol{\Sigma}}_\ell \widehat{\boldsymbol{V}}_\ell^\top$
6: Assign $\widehat{\boldsymbol{U}}_\ell \leftarrow \overline{\boldsymbol{Q}}\widehat{\boldsymbol{W}}_\ell$
7: Extract the largest k singular-pairs from $\widehat{\boldsymbol{U}}_\ell$, $\widehat{\boldsymbol{\Sigma}}_\ell$, $\widehat{\boldsymbol{V}}_\ell$ to obtain $\widehat{\boldsymbol{U}}_k$, $\widehat{\boldsymbol{\Sigma}}_k$, $\widehat{\boldsymbol{V}}_k$

Fig.1 Algorithm of Integrated SVD with multiple sketches (iSVD)

- **Accuracy evaluation of iSVD**

In interim report, we reported parallel performance of iSVD. In this report, we evaluate accuracy of iSVD in viewpoint of low rank approximation to show ability of performance for processing big data.

Our computer environment in this experience is as follows:

- The CX400 (ITC, Nagoya U.)
  - Node: 2 sockets of Xeon E5-2697v3 (Haswell) (2.6GHz).
  - Interconnection: Infiniti Band, full connection.
  - Intel composer_xe_2015.5.223, MKL with same version is used.
  - Intel compiler: version 15.0.4.
  - Intel MPI version 5.0.3.049.

The test matrices used in here are as follows:

- **Matrix 1: A Randomized Matrix**

In this matrix 1, matrices U and V are made by using normal random generator [0,1]. A is made by using the generated U and V with A = U S V^T, where, singular value (true answer) is set to $i = 1 \sim k$: 1.0 / i, $i = k + 1 \sim m$: 1e-2 / (i - k). Hence, true answer of U and V is the U and V.

We can check the answer with:

- Compute mean value of diag(S'), (U U'^T) = U"S" V"^T, where, A = U'S' V'^T.
- Error(max, mean, min): sqrt (diag(S") - mean).

- **Matrix 2: H-Matrix (Low Rank Approximation)**

This matrix 2 is an example to show ability of low rank approximation by using a well-approximated problem. The projection with randomized algorithm, hence, can be established in theoretically.

We use a problem from static electric field analysis. After permutation and partition by H-matrices, which is one of other low rank approximations for matrices, we obtain a target sub-matrix. We use the sub-matrix with N= 300 x 300 for the test matrix.

Error of the sub-matrix can be evaluated with the following three viewpoints:

- Frobenius Norm (FB):

$$\left\|A-U\Lambda V^T\right\|_F / \left\|A\right\|_F , \tag{1}$$

- Mean:

$$\sum_{i=1}^{T} \left(\left\|A-U\Lambda V^T\right\|_F / \left\|A\right\|_F\right)_i / T , \tag{2}$$

- Square root (Sqrt):

$$\sqrt{\sum_{i=1}^{T} \left(\left\|A-U\Lambda V^T\right\|_F / \left\|A\right\|_F - mean\right)_i^2} , \tag{3}$$

where T is the number of test trials in iSVD.

- **Results**

Matrix 1

In this chapter, we evaluate errors of low rank approximation by iSVD. Fig.2 shows errors by Matrix 1 with respect to parallel execution.

Fig.2 indicates the accuracy improves according to the number of cores. Hence the iSVD has scalability of accuracy. Note that the execution time is almost same, but the number of samplings for low rank approximations are increasing to the number of cores.

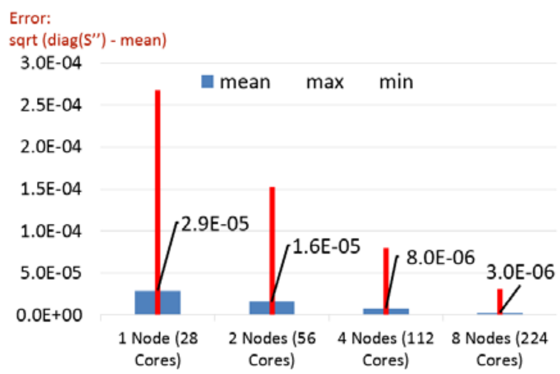


Fig.3 Accuracy of the Matrix 1.

Matrix 2:

Fig. 3 shows errors of low rank approximation by equation (1)-(3). The number of trials T is set to 100. The k is dimension number of low rank approximation.

Fig. 3 indicates that accuracy depends dimensions of row rank approximation. According to Fig. 3 (a), if we accept accuracy around 2e-02, we can reduce matrix dimension from 300x300 to 3x3. This is 1/100 scale. Hence this means $1/100^3$ reduction of computation complexity, since computation complexity of dense eigenproblem is $O(n^3)$.

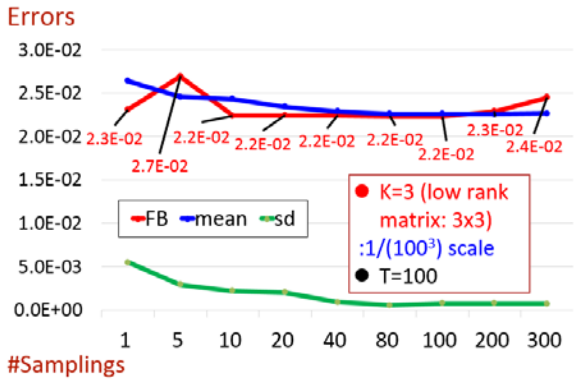
As summary of this experiment, we conclude that iSVD provides reasonable accuracy in viewpoint of low rank approximation. This means that iSVD is one of candidates to process big data analysis for the eigenvalue problem.

Several publications for the other topics are now under preparation.

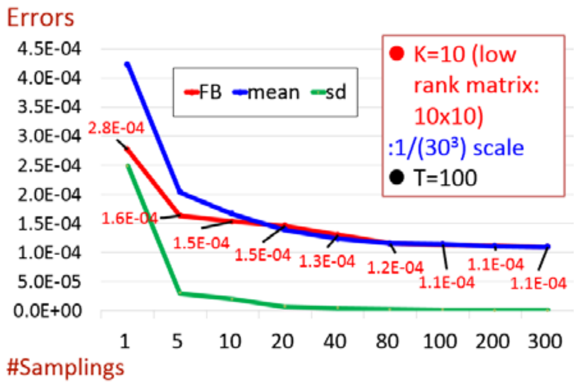
5. Details of FY2017 Research Achievements

The original proposal is planned with the following three years.

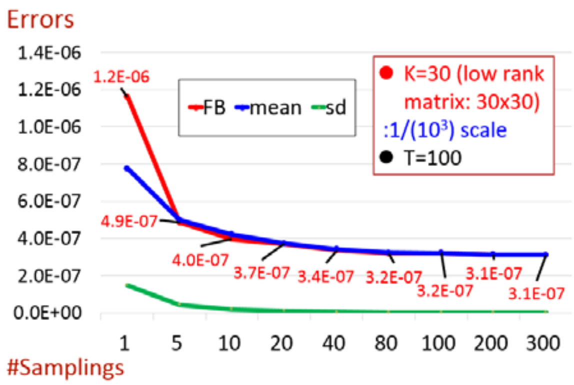
- Year 1 (FY2016): Algorithm development and testing environments deployment. (A prototyping)
- Year 2 (FY2017): Large-scale implementation and software integrations.
- Year 3: Auto-tuning of large-scale codes and tests of applications.



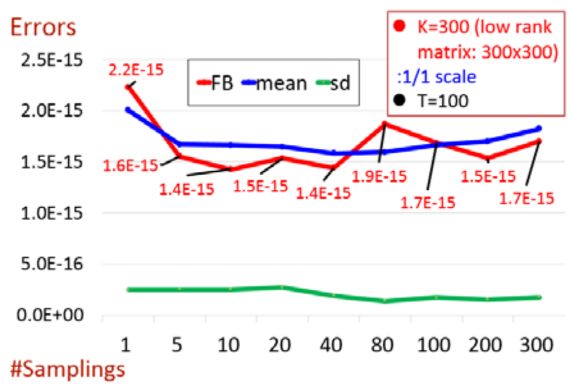
(a) Case of k = 3x3



(b) Case of k = 10x10



(c) Case of k = 30x30



(d) Case of k = 300x300

Fig.3 Accuracy of the Matrix 2.

Mathematical and statistical algorithms:

In the Year 1, we have evaluated several contents as follows: (1) SVD and LS based on random projection and random sampling; (2) Algorithms integrating multiple randomized results based on optimization and numerical linear algebra techniques; (3) Algorithm analysis from viewpoints of geometry, statistics, and complexity; (4) Preliminary numerical experiments for the purpose of proof-of-concept.

Parallelism on large-scale cluster:

In the Year 2 (this year), we focus on the followings: (1) Algorithm parallelism including task separation, task distribution, data structure, data communication; (2) Parallel implementations: (2-a) Single-node: 1-CPU-multi-cores (OpenMP); Multiple-CPU (OpenMP); 1-CPU-1-GPU; multiple-CPU-Multiple-GPU; CPU with Xeon Phi; (2-b) Multiple-node: CPU cluster (MPI + OpenMP); CPU-GPU cluster (MPI + OpenMP);

Auto-tuning and applications:

In the end of the Year 2 and the Year 3, we focus on the followings: (1) Adaptation of general performance model for AT, in particular, surrogate models; (2) Development of auto-tuning methodologies for computation kernels and MPI communications of SVD and LS, in particular, hierarchical memory optimizations; (3) Extension of AT functions for ppOpen-AT with respect to nature of processes for SVD and LS. (4) Human Genetic Variation; Underground water; Air pollution; Images analysis;

6. Progress of FY2017 and Future Prospects

We have several results for eigenvalue program and sparse linear equations solvers in this project. Due to page limitation, we only show result of iSVD.

- Parallel Implementation of iSVD

In this year, a prototyping for parallel implementation has been finished for Prof. Wang's team to establish large scale implementation for iSVD. Main contribution is to parallelize input matrix A with row-block distribution with parallel reduction for MPI to reduce communication time. Fig. 4 shows a typical parallel performance for iSVD with row-block distribution.

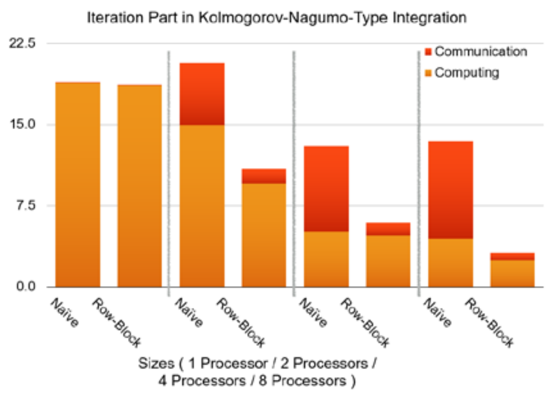


Fig. 4 A Typical Parallel Performance for iSVD with row-block distribution.

According to Fig. 4, communication time for Row-Block distribution can be dramatically reduced according to number of processors, while communication time in conventional (naive) implementation increases. This is because, parallel reduction operation based on row-block distribution is crucial to conventional implementation.

- I/O evaluation

To treat with big data, I/O time cannot be ignored in general. Even in iSVD, I/O time inside processes will be a serious problem if we do not implement efficient I/O. To show evidence of this, breakdown among each procedure and I/O time for previous version of iSVD is shown in Fig. 5.

In Fig. 5, 50.1% to total time is occupied with I/O execution in this environment. It should be parallelized for the I/O process. To show effect of the parallelization, a test for parallel I/O in the example in Fig. 5 is shown in Fig. 6.

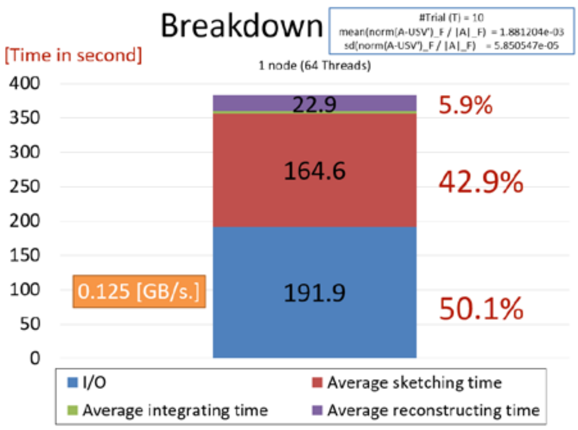


Fig.5 A Breakdown of Execution Time of iSVD in one node of the Oakforest-PACS.
The size of I/O is 24 GB. The size of input matrix is 5,000 x 200,000 , and size of sampled matrix is 500 x 500.

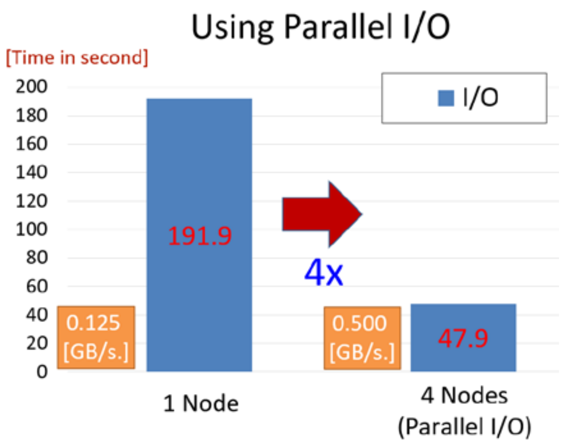


Fig.6 Parallel I/O Effect with 4 MPI processes.

According to Fig. 6, we can establish 4x speedup of I/O with 4 MPI processes. This is perfect scaling. Please note that the result is not using fast I/O technologies, such as burst buffer. Hence, we have enough room to speed up the I/O implementation.

- Application Adaptation

We have adapted several applications, but due to page limit, we show an adaptation example of iSVD. We apply iSVD algorithm to Facebook 100k data. This application requires 108,585 x 108,585 matrix ***A***. For the iSVD, we summarize settings as follows.

- Row-block Gaussian projection sketching (CPU/GPU),
- Row-block Gramian orthogonalization,
- Row-block Wen-Yin integration, and
- Row-block Gramian former.

We use C++ for iSVD. Target machine is the Reedbush-H at ITC, the University of Tokyo. Using number of nodes is 4 nodes, 144 cores, including 8 GPUs (Pascal). Numerical parameters of iSVD are set to N = 256, k = 20, p = 12, P = 8.

For MATLAB execution, we use a PC cluster (WLab Cluster) with 1 Node, 24 cores. Numerical parameters of N = 256, k = 20, p = 12.

Fig. 7 shows total execution time in seconds between MATLAB execution and iSVD execution.

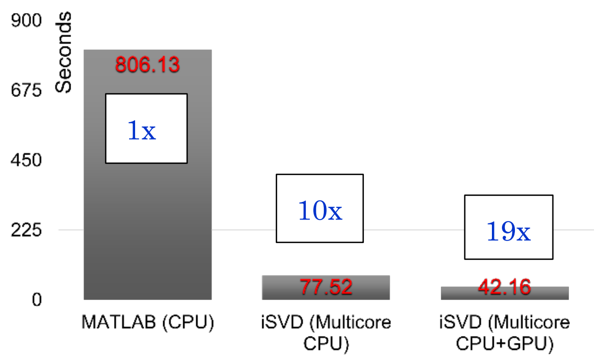


Fig. 7 Total Runtime of Facebook Data. Time in seconds.

According to Fig.7, our developed parallel iSVD has a merit from 10x to 19x. Hence, we can conclude that our developed iSVD code is very useful for actual application.

- Future Prospects

There are several topics to progress the research in this financial year. The research topics are summarized as follows.

- Detailed evaluation on several supercomputers, and improvement of implementations for MPI.
- Adaptation of auto-tuning for selection of implementations that are inside processes in iSVD. However, this is main research topic for the Year 3.
- Additional adaptations and evaluations to many application software.

- Related Workshop

In this year, we held in an international conference related to the project, named Second International Workshop on Deepening Performance Models for Automatic Tuning (DPMAT) in September 2017 at Nagoya University. Many members in this project presented their work in DPMAT2017. See the attached list of publications.

7. List of Publications and Presentations

(1) Journal Papers

None.

(2) Conference Papers

None.

(3) Oral Presentations

[1] Weichung Wang: "Parallel Singular Value Decomposition for Large Matrices by Multiple Random Sketches", DPMAT2017, (Sep. 2017)

[2] Takahiro Katagiri: "Auto-tuning to Scientific Applications –Traditional Approach by Code Transformations and New Approach by AI –", DPMAT2017, (Sep. 2017)

[3] Feng-Nan Hwang, "Nonlinear Preconditioner for Full-space Lagrange-Newton-Krylov Algorithms with applications in Large-scale PDE-constrained Optimization Problems", DPMAT2017, (Sep. 2017)

[4] Osni Marques, "Unconstrained Functionals for Efficient Parallel Scaling of Conjugate Gradient Eigensolvers", DPMAT2017, (Sep. 2017)

[5] Takahiro Katagiri, Mu Yang, Weichung Wang, Akihiro Ida, "Performance Evaluation of Integration of Multiple Randomized Low-Rank Singular Value Decompositions", 2017年並列／分散／協調処理に関する『秋田』サマー・ワークショップ (SWoPP2017), 日本応用数理学会 「行列・固有値問題の解法とその応用」研究部会 (MEPA) (Jul. 2017)

[6] Weichung Wang, "Parallel Singular Value Decomposition via Multiple Random Sketches for Large Matrices", International Conference on Econometrics and Statistics (EcoSta 2017) (June 2017)

[7] Weichung Wang, "Parallel SVD by Random Sketches and Big Data Intelligent Analytics Forum on Artificial Intelligence for Data Science", National Sun Yat-sen University, Kaohsiung, Taiwan (Dec. 2017)

[8] Weichung Wang, "Efficient Matrix Decompositions for Photonic Device Simulations & Big Data Analytics", Second International Symposium on Research and Education of Computational Science (RECS), The Computational Science Alliance of the University of Tokyo (Dec. 2017)

(4) Others

None