

jh180027-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 and medical processing application for big data analysis with several kinds of supercomputers with providing JHPCN. Target of the algorithm is singular value decomposition (SVD), related eigenproblem, and registration problem for medical image processing. In this year, we focus on auto-tuning. We newly add application for medical image processing. Hence in this report, a medical image processing program is evaluated with a provided supercomputer resource with respect to tunable performance parameters.

## 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.

[Hidekata Hontani](#) provides knowledge of medical image processing for registration problem.

[Toshihiro Hanawa](#) provides knowledge of system software (I/O) for medical image processing.

## 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, and medical image processing for large scale data, like 3D medical body scanning data. 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.

In 2018 FY, we also focus on a medical image processing to show effectiveness of big data processing.

### 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.

In addition to the above, medical image processing algorithm is added for the third year. Medical image processing requires very large medical data, like several Tera Bytes for one sample. Hence it is very nice example to treat big data sets.

### 4. Outline of the Research Achievements up to FY2017

We have several results for eigenvalue program and sparse linear equations solvers in this project. We only show result of iSVD.

● **Parallel Implementation of iSVD**

In the Year FY2017, 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. 1 shows a typical parallel performance for iSVD with row-block distribution.

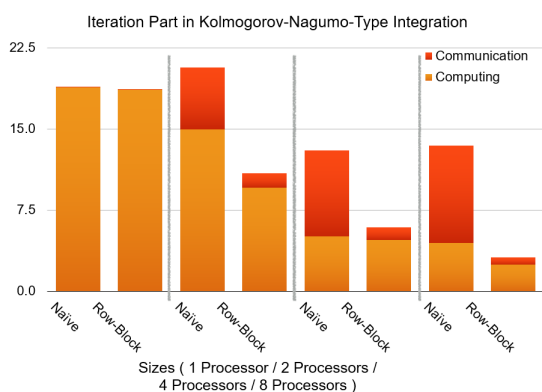


Fig. 1 A Typical Parallel Performance for iSVD with row-block distribution.

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. 2.

In Fig. 2, 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. 2 is shown in Fig. 3.

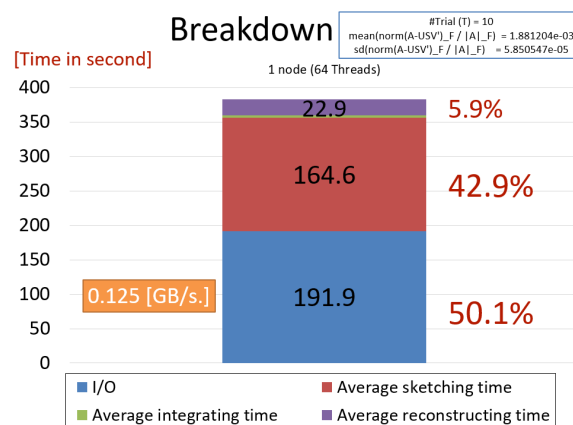


Fig.2 A Breakdown of Execution Time of iSVD in one node of the Oakforest-PACS.

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.

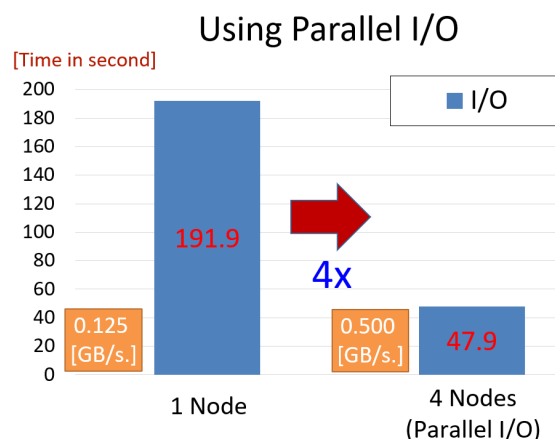


Fig.3 Parallel I/O Effect with 4 MPI processes.

According to Fig. 3, 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. 3 shows total execution time in seconds between MATLAB execution and iSVD execution.

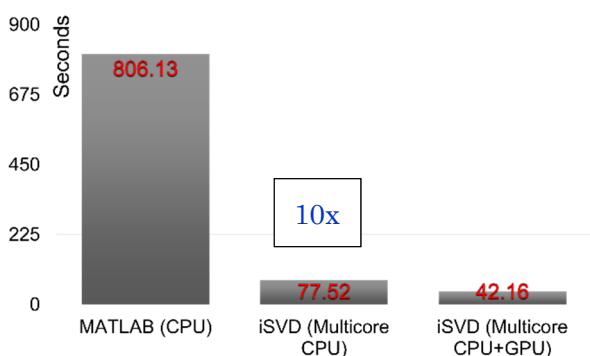


Fig. 3 Total Runtime of Facebook Data.  
 Time in seconds.

According to Fig.3, 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.

**5. Details of FY2018 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 (FY2018):** Auto-tuning of large-scale codes and tests of applications.

The adaptation of auto-tuning **includes finding auto-tuning parameters** in each application. Also, in this year, an image processing application is included, hence **developing high performance implementation for the image processing application is also main aim** with respect to adaptation of auto-tuning.

**6. Progress of FY2018 and Future Prospects**

In this year, medical image processing is newly added for target program. We explain preliminary result of parallelization by using a JHPCN supercomputer.

● **Overview of the Target Program**

We use NIX code for medical image processing. The NIX code is a registration processing code for 3D body scanning data, which is developed by Hontani Group, Nagoya Institute of Technology, Japan. The algorithm for the registration is based on the large deformations diffeomorphic metric mapping (LDDMM). Hontani Group developed a new method, Partially Rigid Large Deformation Diffeomorphic Metric Mapping (PR-LDDMM). NIX code is using PR-LDDMM.

The following four STEPs are majority parts for computation in the NIX code.

1. **STEP 1 - Calculate Jacobians,**
2. **STEP 2 - Backward Integration,**
3. **STEP 3 - Update Velocities, and**

#### 4. STEP 4 - Update Paths & Landmarks.

We focus on the above parts from the STEP 1 to the STEP 3 for adaptation of parallelization. However the heaviest part is the STEP 3.

Overview of codes for the STEP 3 is shown in Fig. 4. In Fig. 4, there is a five-nested loop. We can parallelize the loops with any loops. To increase parallelisms, the most inside three-nested loop (colored red lines) can be adapted for loop collapse. This is one of code optimization techniques (Loop transformation).

In this research, performance of the loop collapse is evaluated.

```

for (unsigned int t=0; t<M; t++) {
#pragma omp parallel for
for(int n=0;n<(int)N;n++) {
float f1 = L[t][n].y[0] - L[M-1][n].x[0]; float f2 = L[t][n].y[1] - L[M-1][n].x[1];
float f3 = L[t][n].y[2] - L[M-1][n].x[2];
K[n][0] = f1*L[t][n].Dp[0][0] + f2*L[t][n].Dp[1][0] + f3*L[t][n].Dp[2][0];
K[n][1] = f1*L[t][n].Dp[0][1] + f2*L[t][n].Dp[1][1] + f3*L[t][n].Dp[2][1];
K[n][2] = f1*L[t][n].Dp[0][2] + f2*L[t][n].Dp[1][2] + f3*L[t][n].Dp[2][2];
}
#pragma omp parallel
for(i=0;i<x;i++) {
for(j=0;j<y;j++) {
for(k=0;k<z;k++) {
for(unsigned int n=0; n<N; n++) {
float f1 = L[t][n].x[0] - (float); float f2 = L[t][n].x[1] - (float); float f3 = L[t][n].x[2] - (float);
float f4 = (f1*f1 + f2*f2 + f3*f3)/Sigma;
if (f4>minExp) {
float f0 = fexp(f4); Kx += f0*K[n][0]; Ky += f0*K[n][1]; Kz += f0*K[n][2];
}
}
}
}
}
B[t].v[0][i][j][k] -= Epsilon*(2*B[t].v[0][i][j][k] - Rho*Kx);
B[t].v[1][i][j][k] -= Epsilon*(2*B[t].v[1][i][j][k] - Rho*Ky);
B[t].v[2][i][j][k] -= Epsilon*(2*B[t].v[2][i][j][k] - Rho*Kz);
}
}
}
}
}

```

Fig. 4 Overview of codes for the STEP 3 (Update Velocities) in NIX code.

#### • Parallelization Method

We apply the following methodology to the NIX code for 3D registration problem because the program requires such computational complexity rather than memory space complexity, we mentioned.

1. Allocate memory space as same as sequential program.
2. Separate computations into each MPI (Message Passing Interface) process by changing loop length.
3. Store partially computed results in 2 to sending buffer.

4. Gather the results in 3 by using **MPI\_Allgather**.

#### • Result

We use one node of the Fujitsu PRIMEHPC FX100 (Hereafter, FX100) at the Information Technology Center, Nagoya University. Each node has 32 cores. We use 32 nodes with Hybrid MPI/OpenMP execution. The number of OpenMP threads is fixed as 32 threads. Hence the number of MPI processes equals the number of nodes in this environment.

We have implemented for three different candidates for computational kernels, such as (i)Original loop, (ii) Loop collapse of two-nested loop, and (ii) Loop collapse of three-nested loop.

Fig. 5 shows speedup ratios to original loop execution.

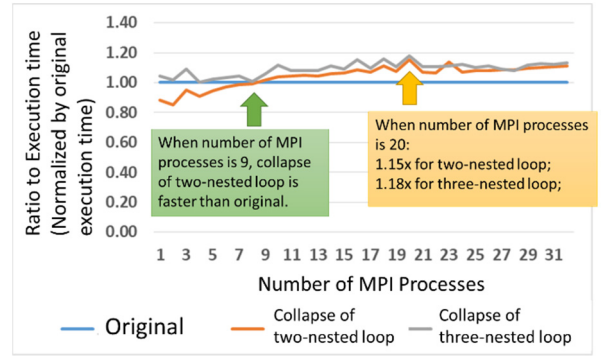


Fig. 5 Speedup ratios for hybrid

MPI/OpenMP executions. The ratios are

based on the original execution.

According to Fig. 5, when number of MPI processes is 9, collapse of two-nested loop is faster than original. On the other hand, when number of MPI processes is 24, two-nested loop is the fastest. Hence the fastest implementation changes according to the number of MPI processes. Hence we find tuning parameter of implementations in this application.

Other way to tune the application is changing scheduling for OpenMP. The tunable parameters are chunk size and scheduling method, such as static or dynamic. The whole

execution time by changing the performance parameters is shown in Fig. 6.

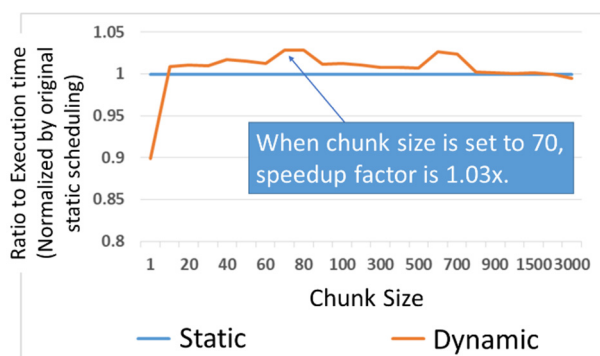


Fig. 6 Speedup ratios for changing scheduling method in OpenMP. The ratios are based on static scheduling in each size of chunk.

Fig.6 shows that the fastest parameter is 70 for chunk size and dynamic scheduling. These are also tunable parameters for auto-tuning.

As a conclusion of the numerical experiments, we found tunable parameters for auto-tuning for NIX code, such as loop transformation (loop collapses), and OpenMP scheduling (chunk size, static or dynamic). This is result for auto-tuning in this project.

## 7. List of Publications and Presentations

### (1) Journal Papers : None

### (2) Conference Papers:

1. 中島大地, 田村友輝, 物部峻太郎, 片桐孝洋, 本谷秀堅, 永井亨, 荻野正雄, “医用画像処理における LDDMM の並列化とコード最適化”, 研究報告ハイパフォーマンスコンピューティング(HPC), Vol. 2018-HPC-167, No. 2, pp.1-7, 2018.

### (3) Oral Presentations

2. W. Wang, “Artificial Intelligences for Medical Images Analysis and Applications”, Ministry of Science and Technology, Taipei, Taiwan, April, 2018.

3. W. Wang, “Artificial, Augmented, and Human Intelligence in the Medical Imaging Chain”, The Annual Meeting for Taiwan Medical

Association, Taipei, Taiwan, November, 2018

4. Furen Xiao, Cheyu Hsu, Chiatze J. Wang, Pochun Wang, Tinghui Wu, Hueihsiuan Yen, Weichung Wang, “Three Dimensional Deep Neural Network Based Multi-Modality and Multi-Organ Automatic Segmentation for Brain Radiosurgery”, RSNA Annual Meeting, Chicago, USA, November, 2018.

5. 中島大地, 田村友輝, 物部峻太郎, 片桐孝洋, 本谷秀堅, 永井亨, “医用画像処理に用いられる大変形微分同相写像の MPI 並列化とコード最適化”, 日本応用数理学会 2018 年度年会, 2018.

6. Su-Yun Huang, Ting-Li Chen, Hung Chen, Dawei D. Chang, Mu Yang, Chen-Yao Lin, Weichung Wang, “A Scalable Randomized SVD with Multiple Sketches for Big Data Analytics”, SIAM CSE19, Spokane, Washington, USA, Feb. 2019.

### (4) Others

[ Poster Presentation]

7. Chiatse J. Wang , Yuhsiang M. Tsai , Chengen Lee, Yuehchou Lee, Anthony Costa, Cheyu Hsu, Eric Oermann, Weichung Wang, “Brain tumor segmentation with capsule networks versus fully convolutional neural networks”, MICCAI, Granada, Spain, September, 2018.