jh180027-DAJ

Takahiro Katagiri (Nagoya University)

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

● Background

- Developing random sketching algorithms with high-performance implementations on supercomputers to compute **singular value decomposition (SVD) and linear system (LS)** solutions of very large-scale matrices.
- Few numerical solvers, especially **randomized algorithms**, are designed to tackle very large-scale matrix computations on the latest supercomputers. .
- 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.
- We intend to implement the proposed algorithms on supercomputers. 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.

● Members

- **Takahiro Katagiri** (Nagoya U., Japan) : AT (ppOpen-AT), parallel eigenvalue algorithms, and supercomputer implementations.
- **Weichung Wang** (National Taiwan U., Taiwan): Numerical linear algebra, parallel computing, and AT (surrogate-assisted turning). and big data applications.
- **Su-Yun Huang** (Institute of Statistical Science, Academia Sinica, Taiwan) : Mathematical statistics and machine learning (random sketching algorithm).
- **Kengo Nakajima** (U. Tokyo, Japan) : Parallel algorithms in numerical iterative method (hybrid MPI/OpenMP execution).
- **Osni Marques** (LBNL, USA) : Eigenproblem and its implementation (LAPACK, SVD algorithms).
- **Feng-Nan Hwang** (National Central U., Taiwan) : Eigenproblem and its parallelization (SLEPc, SVD algorithms)
- **Toshio Endo** (TITECH, Japan) : System software (optimizations for hierarchical memory and adaptation of its AT)
- **Hidekata Hontani** (Nagoya Institute of Technology, Japan): Providing knowledge of Medical Image Processing

● iSVD Algorithm

Rank-k SVD

$$A \approx U_k \Sigma_k V_k^\top$$

U_k is an $m \times k$ orthonormal matrix that $k < m$, Σ_k is a $k \times k$ diagonal matrix, and V_k is an $n \times k$ orthonormal matrix. The columns of U_k and V_k are the leading left singular vectors and right singular vectors of A, respectively. The diagonal entries of Σ_k are the k largest singular values of A.

Algorithm 2 Integrated SVD with multiple sketches (iSVD).

Require: Input 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 $A \approx \widehat{U}_k \widehat{\Sigma}_k \widehat{V}_k^\top$

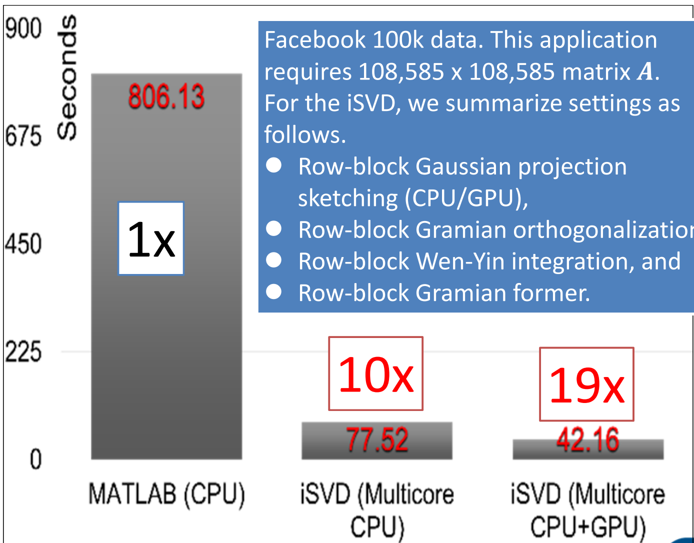
Generate $n \times \ell$ random matrices $\Omega_{[i]}$ for $i = 1, \ldots, N$
Assign $Y_{[i]} \leftarrow (AA^\top)^q A\Omega_{[i]}$ for $i = 1, ..., N$ with $\Omega_{[i]} = \Omega_{gp}$ or Ω_{cs} (in parallel)
Compute $Q_{[i]}$ whose columns are orthonormal basis of $Y_{[i]}$ (in parallel)
Integrate $\overline{Q} \leftarrow \{Q_{[i]}\}_{i=1}^N$ (by Algorithm 3 or Algorithm 4)
Compute SVD of $\overline{Q}^\top A = \widehat{W}_\ell \widehat{\Sigma}_\ell \widehat{V}_\ell^\top$
Assign $\widehat{U}_\ell \leftarrow \overline{Q}\widehat{W}_\ell$
Extract the largest k singular-pairs from $\widehat{U}_\ell$, $\widehat{\Sigma}_\ell$, $\widehat{V}_\ell$ to obtain $\widehat{U}_k$, $\widehat{\Sigma}_k$, $\widehat{V}_k$

● Research Plan

- 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.

[By Ting-Li Chen, Su-Yun Huang, Hung Chen, David Chang, Chen-Yao Lin, and Weichung Wang]

● Application Adaptation



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.

Main Results of FY2017

● Parallel Implementation of iSVD

Main contribution is to parallelize input matrix *A* with row-block distribution with parallel reduction for MPI to reduce communication time. The following figure shows a typical parallel performance for iSVD with row-block distribution.