QR Decomposition of Block Low-Rank Matrices

Muhammad Ridwan Apriansyah[1] and Rio Yokota[2]

[1]School of Computing, Tokyo Institute of Technology. ridwan@rio.gsic.titech.ac.jp
[2]Global Scientific Information and Computing Center, Tokyo Institute of Technology. rioyokota@gsic.titech.ac.jp

Rio Yokota Lab

Abstract

QR decomposition is a fundamental operation in linear algebra, and is often used in scientific computations. The cost of $O(N^2)$ storage and $O(N^3)$ computations to perform QR decomposition could be reduced if the matrix is approximated using hierarchical low-rank approximation. This work extends the tiled QR decomposition algorithm of Gunter and Geijin to work on block low-rank matrices (BLR-matrices), which is a simplified variant of hierarchical matrices. By choosing the appropriate block size, the algorithm requires $O(N^{1.5})$ memory and $O(N^{2.5})$ computations. We evaluate the speed and accuracy of the algorithm and compare it to the existing methods. Furthermore, we also present a parallel algorithm for QR factorization of BLR-matrices on shared-memory system.

Introduction

Block Low-Rank Matrices

Block low-rank matrices allows us to express dense matrix as blocks of full rank and low-rank matrices. The low-rank block is represented as a product of the Singular Value Decomposition (SVD) of the corresponding dense matrix.

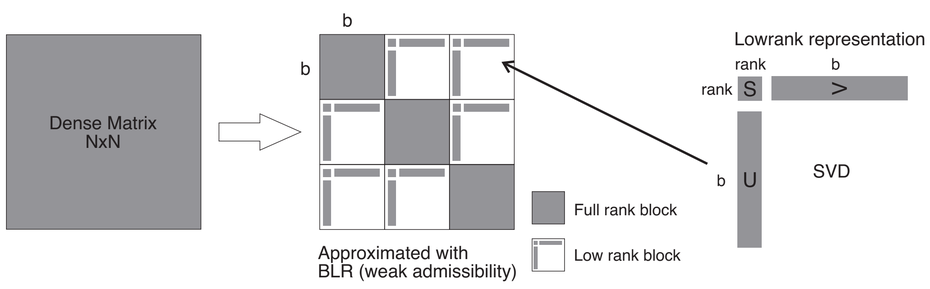

Figure 1: A dense matrix expressed as block low-rank matrix

Blockwise QR Decomposition

Algorithm 1: Blockwise QR decomposition
Input: BLR-matrix A of size $p \times q$
Output: BLR-matrix A and block-dense matrix T

```
for k = 1 to min(p,q) do
    GEQRT(A_kk, Y_kk, T_kk, T_kk);
    for j = k+1 to q do
        LARFB(A_kj, Y_kk, T_kk, R_kj);
    end
    for i = k+1 to p do
        TPQRT(R_kk, A_ik, Y_ik, T_ik);
        for j = k+1 to q do
            TPMQRT(R_kj, A_ij, Y_ik, T_ik);
        end
    end
end
```

Subroutines

Based on LAPACK [1] subroutines

- GEQRT$(A_{kk}, Y_{kk}, T_{kk}, R_{kk})$:
  $QR(A_{kk}) \rightarrow (Y_{kk}, T_{kk}, R_{kk})$
- LARFB$(A_{kj}, Y_{kk}, T_{kk}, R_{kj})$:
  $R_{kj} = (I - Y_{kk} T_{kk}^T Y_{kk}^T) A_{kj}$
- TPQRT$(R_{kk}, A_{ik}, Y_{ik}, T_{ik})$:
  $QR \begin{pmatrix} R_{kk} \\ A_{ik} \end{pmatrix} \rightarrow (Y_{ik}, T_{ik}, R_{kk})$
- TPMQRT$(R_{kj}, A_{ij}, Y_{ik}, T_{ik})$:
  $\begin{pmatrix} R_{kj} \\ A_{ij} \end{pmatrix} = \left(I - \begin{pmatrix} I \\ Y_{ik} \end{pmatrix} T_{ik}^T \begin{pmatrix} I & Y_{ik}^T \end{pmatrix} \right) \begin{pmatrix} R_{kj} \\ A_{ij} \end{pmatrix}$

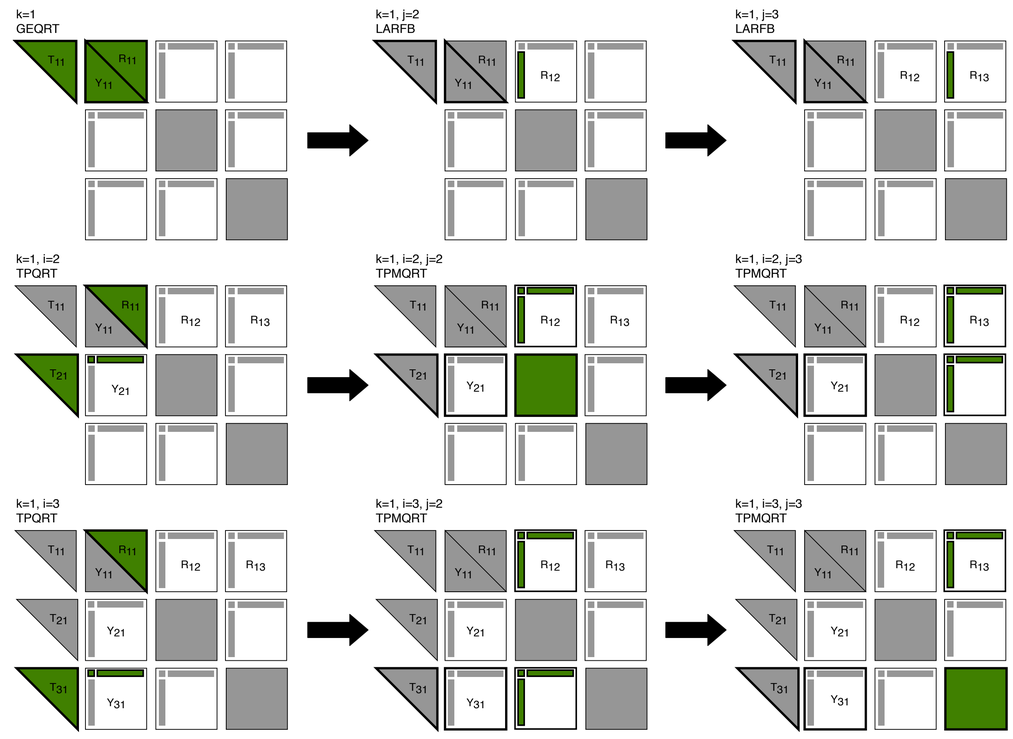

Figure 2: Graphical representation of Algorithm 1 on a BLR matrix with $p = q = 3$. A thick border shows the blocks that are being read and a green fill shows the blocks that are being written at each step.

Parallel Blockwise QR Decomposition

Some of the operations in Algorithm 1 are independent of each other and can be done in parallel. By formulating each function call as a task and considering the dependencies between them, we can generate the dependency graph and execute the tasks in parallel based on the graph.

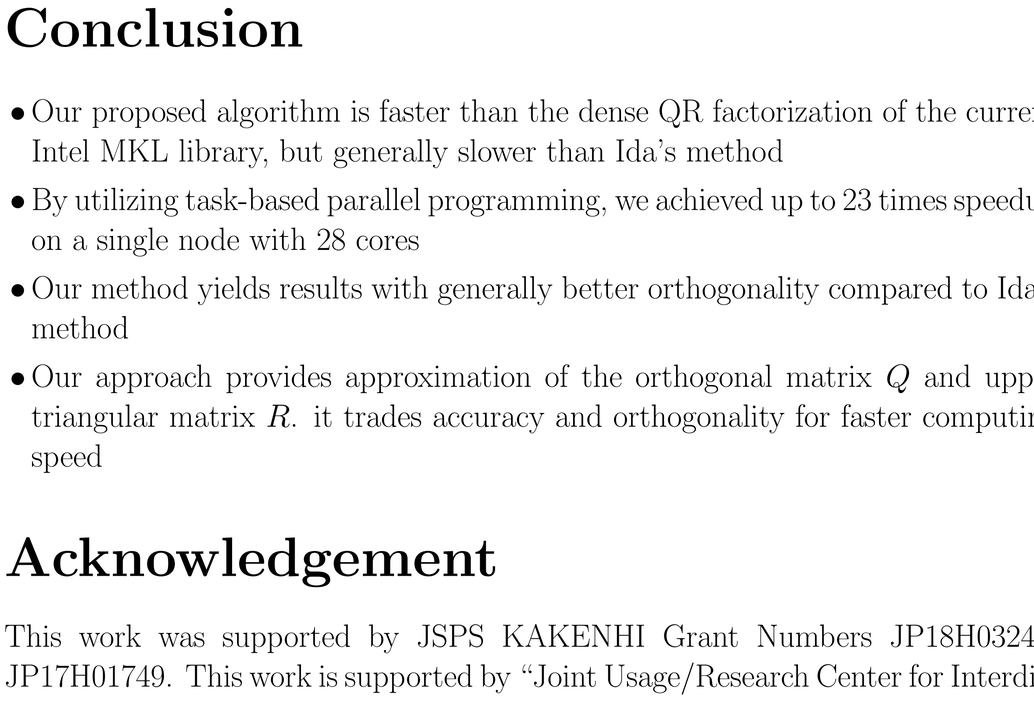

Figure 3: Dependency graph of operations in Algorithm 1 on 3x3 BLR-matrix

Results

- Experiments were conducted on TSUBAME 3.0 (up to 28 cores on single node)
- OpenMP 4.5 *tasks* is used for parallelization
- Configured with $block\_size = \sqrt{Nr}$, fixed $rank = 16$ for low-rank blocks
- Compared with "geqrt2" routine of Intel MKL and existing BLR QR decomposition method of Ida et al. [2]

(a) Execution time

(b) Parallel scalability

(c) Residual

$\frac{\|A - QR\|}{\|A\|}$

(d) Orthogonality

$\frac{\|I - Q^T Q\|}{\sqrt{N}}$

Conclusion

- Our proposed algorithm is faster than the dense QR factorization of the current Intel MKL library, but generally slower than Ida's method
- By utilizing task-based parallel programming, we achieved up to 23 times speedup on a single node with 28 cores
- Our method yields results with generally better orthogonality compared to Ida's method
- Our approach provides approximation of the orthogonal matrix Q and upper triangular matrix R. it trades accuracy and orthogonality for faster computing speed

Acknowledgement

This work was supported by JSPS KAKENHI Grant Numbers JP18H03248, JP17H01749. This work is supported by "Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures" in Japan (Project ID: jh190043-NAHI).

References

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

[2] Akihiro Ida, Hiroshi Nakashima, Tasuku Hiraishi, Ichitaro Yamazaki, Rio Yokota, and Takeshi Iwashita. Qr factorization of block low-rank matrices with weak admissibility condition. *Journal of Information Processing*, 27:831–839, 2019.