CSIAM Trans. Appl. Math.
doi: 10.4208/csiam-am.2020-0206

Vol. 2, No. 3, pp. 431-459
September 2021

Distributed-Memory $\mathcal{H}$-Matrix Algebra I: Data Distribution and Matrix-Vector Multiplication

Yingzhou Li[1,*], Jack Poulson[2] and Lexing Ying[3]

[1] *School of Mathematical Science, Fudan University, Shanghai, China.*
[2] *Hodge Star, Toronto, Canada.*
[3] *Department of Mathematics and ICME, Stanford University, Stanford, CA 94305, USA.*

Received 29 August 2020; Accepted 13 February 2021

Abstract. We introduce a data distribution scheme for $\mathcal{H}$-matrices and a distributed-memory algorithm for $\mathcal{H}$-matrix-vector multiplication. Our data distribution scheme avoids an expensive $\Omega(P^2)$ scheduling procedure used in previous work, where P is the number of processes, while data balancing is well-preserved. Based on the data distribution, our distributed-memory algorithm evenly distributes all computations among P processes and adopts a novel tree-communication algorithm to reduce the latency cost. The overall complexity of our algorithm is $\mathcal{O}\big(\frac{N\log N}{P}+\alpha\log P+\beta\log^2 P\big)$ for $\mathcal{H}$-matrices under weak admissibility condition, where N is the matrix size, α denotes the latency, and β denotes the inverse bandwidth. Numerically, our algorithm is applied to address both two- and three-dimensional problems of various sizes among various numbers of processes. On thousands of processes, good parallel efficiency is still observed.

AMS subject classifications: 65F99, 65Y05

Key words: Parallel fast algorithm, $\mathcal{H}$-matrix, distributed-memory, parallel computing.

1 Introduction

For linear elliptic partial differential equations, the blocks of both forward and backward operators, when restricted to non-overlapping domains, are numerically low-rank [7]. Hence both operators can be represented in a data sparse form. Many fast algorithms benefit from this low-rank property and apply these operators in quasi-linear scaling. Such fast algorithms include but not limit to tree-code [4, 41], fast multipole method

*Corresponding author. *Email addresses:* yingzhouli@fudan.edu.cn (Y. Li), jack@hodgestar.com (J. Poulson), lexing@stanford.edu (L. Ying)

 ©2021 Global-Science Press

(FMM) [3, 11, 18–20, 39, 42, 50], panel clustering method [21], etc. The low-rank structures in these fast algorithms are revealed via various interpolation techniques such as: pole expansion, Chebyshev interpolation, equivalent interaction, etc [15, 19, 39, 50].

In contrast to approximating the application of operators, another group of research focuses on approximating operators directly in compressed matrix forms. As one of the earliest members in this group, $\mathcal{H}$-matrix [5–7, 17, 21, 22, 26–28] hierarchically compresses operators restricted to far-range interactions by low-rank matrices. The memory cost and matrix-vector multiplication complexity are quasi-linear with respect to the degrees of freedom (DOFs) in the problem. Shortly after introducing $\mathcal{H}$-matrix, Hackbusch et al. [23] again introduced $\mathcal{H}^2$-matrix, which uses nested low-rank bases to further reduce the memory cost and multiplication complexity down to linear. Related to the fast algorithms above, $\mathcal{H}$-matrix and $\mathcal{H}^2$-matrix can be viewed as algebraic versions of tree code and FMM respectively. But they are more flexible in choosing different admissibility conditions and low-rank compression techniques, which are related to general advantages of algebraic representations.

Developments in the $\mathcal{H}$-matrix group and extensions beyond the group are explored in the past decade. Hierarchical off-diagonal low-rank matrix (HOLDER) [2] and hierarchical semi-separable matrix (HSS) [48] are two popular hierarchical matrices with the simplest admissibility condition, i.e., weak admissibility condition. Different from hierarchical matrices, recursive skeletonization factorization (RS) [37] and hierarchical interpolative factorization (HIF) [24, 25] introduce separators in the domain partition and compress the operator as products of sparse matrices. The partition and factorization in RS and HIF are in the similar spirit as that in multifrontal method [1, 12] and superLU method [30], while extra low-rank approximations are introduced to compress the interactions within frontals. Other algebraic representations include block low-rank approximation [49], block basis factorization [45], etc. The benefits of algebraic representations over analytical fast algorithms come in two folds: 1) numerical low-rank approximation is more effective than interpolation; 2) matrix factorization and inversion become feasible. We emphases that these algebraic representations are not only valid for linear elliptic operators, but also valid for operators associated with low-to-medium frequency Helmholtz equations and radial basis function kernel matrices. When operators admit high-frequency property, the low-rank structure appears in a very different way comparing to that in all aforementioned fast algorithms, and are also well-studied by the community [8, 9, 13, 14, 31–34, 38].

Many of these fast algorithms and algebraic representations have been parallelized on either shared-memory or distributed-memory setting to be applicable to practical problems of interest [10, 16, 18, 35, 40–44, 46, 47, 50]. Here we focus on the parallelization of $\mathcal{H}$-matrix. Kriemann [28, 29] implemented a shared-memory parallel $\mathcal{H}$-matrix using a block-wise distribution, i.e., each block is assigned to a single process. Processes assigned to blocks near root level are responsible for computations of complexity linear in N, where N is the total DOFs. Hence the speedup of such a parallelization scheme is theoretically upper bounded by $\mathcal{O}(\log N)$ and limited in practice up to 16 processes.