論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Linear Time Dense Direct Solver Without Trailing Sub-matrix Dependencies
著者(和文)	MAQianxiang
Author(English)	Qianxiang Ma
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第12880号, 授与年月日:2024年9月20日, 学位の種別:課程博士, 審査員:横田 理央,吉瀬 謙二,宮﨑 純,DEFAGO XAVIER,小野 峻佑
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第12880号, Conferred date:2024/9/20, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Category(English)	Doctoral Thesis
種別(和文)	論文要旨
Type(English)	Summary

（博士課程）
Doctoral Program

論 文 要 旨

THESIS SUMMARY

系・コース： Department of, Graduate major in	Computer Science Computer Science 系 コース	申請学位 (専攻分野)： Academic Degree Requested	博士 Doctor of (Engineering)
学生氏名： Student's Name	Qianxiang Ma	審査員主査： Chief Examiner	Rio Yokota

要旨（英文 800 語程度）
Thesis Summary (approx.800 English Words)

Obtaining the dense matrix factorization and direct solutions is used in various fields in scientific applications, such as the frontal matrix in multi-frontal sparse solvers and preconditioners of the integral equation solvers based on the boundary element methods. Driven by the application needs in machine learning, the recent hardware development trends have focused more and more on the adoption of low-precision arithmetic units and SIMD parallelism. Attracted by their outstanding performance, the low-precision and low-rank computations in scientific applications also gained their attention and emerging popularity.

Dense matrices usually arise as a form to represent an all-to-all interaction, such as the amount of electrostatic force placed on the particles, where both the sources and the targets are the complete set of the particles in a domain. The low-rank approximation comes into play in this problem setting, as the electrostatic force has characteristics that its magnitude decays rapidly for the increasing distance between the particles, where its magnitude is small enough to be approximated, yet cannot be completely ignored. The fast multipole method (FMM) is an analytical method that uses multipole expansions to approximate such long-range forces, that for an electrostatic N-body problem, it computes the potential on all bodies in just O(N) time.

FMM has its limits such as the multipole expansion requires knowledge of the gradient functions, and the complexity of the algorithm is critically dependent on the polynomial of the expansion order. In contrast, algebraic means for creating a low-rank approximation of a matrix come in more efficient in both the performance and rank representation. Additionally, the algebraic low-rank approximation does not require a differentiable kernel function, which gives it a broader range of use cases. The algebraic interpretations of the FMM can be viewed as an H^2-matrix, a member in the family of structured low-rank approximated matrices, where they differ in the following aspects the number of levels, the admissibility condition, and the use of shared or independent bases. The three aspects influence the implementation complexity, the algorithmic complexity, the different algorithms to adopt, and the ranks needed to meet the desired accuracy. Viewing from the algebraic perspective, the structured low-rank approximated matrices have capabilities of not only computing matrix-vector multiplication in linear time but also other arithmetics. This thesis explores the hierarchical low-rank approximation of dense matrices, which can reduce the complexity of the arithmetic from O(N^3) to close to O(N), including matrix-vector multiplication, matrix-matrix multiplication, factorization, and forward and backward substitution.

Existing approaches have extremely parallel implementations in the matrix-vector and matrix-matrix multiplications. Many researchers have devoted their time to searching for efficient algorithms and parallelization tools,

but only a few members of the structured low-rank matrices had their factorization implemented in parallel and running on large-scale supercomputers. One of the hardest challenges to overcome in obtaining a direct factorization for structured low-rank matrices is the chain of diagonal dependencies existing in the Cholesky/LU factorization. In the dense factorization and single-level BLR factorization, automated runtime systems introduce a significant boost in performance as the off-diagonal computation is heavy and the dominating factor, but they have little impact on the hierarchically structured low-rank matrices. For BLR^2 and HSS matrices, a possibility has been found to remove the data dependency amongst the diagonals via a ULV factorization. However, the remaining challenges existed primarily for the use of stronger admissibility conditions, which is a must-have feature for solving problems that have geometry in high-dimensional space. Only using the weak admissibility configurations poses great limits on the number of dimensions in the problem geometry, primarily due to the growing ranks of off-diagonal blocks in 3-D problems, and the method is no longer made O(N).

To address this problem, this thesis discusses the extension to the ULV factorization and substitution, in particular, made for strongly admissible H^2-matrices. The contribution discusses the algorithm, which removes the need to wait for the basis updates via a pre-compression phase, that brings the re-compression computations happening inside the ULV factorization before all factorization operations. By having this means of integrating the pre-compressed results into the shared basis, we also made the data dependencies in the diagonal trailing blocks disappear. The work presented featured the first inherently parallel factorization and substitution on distributed memory for a strongly admissible hierarchical low-rank matrix in O(N) complexity, without any assistance or the overhead from the parallel runtime tools. From the algorithm improvement alone, we achieved a maximum speed up of 4,700x for a 3-D problem with complex geometry, compared with a block low-rank factorization code LORAPO. In the many-GPU implementation of the same algorithm, we utilize the batched interfaces of BLAS and LAPACK and unify the different block sizes. The batching of the kernels compensates for the low arithmetic intensity in the low-rank computations, and we eventually extracted over 800 TFLOPS of performance on 512 NVIDIA V100 GPUs. Our implementation is also the first implementation for obtaining the direct solution of an H^2-matrix running on multiple GPUs.