An Optimization of $\mathcal{H}$-matrix-vector Multiplication by Using Un-used Cores

Tetsuya Hoshino
The University of Tokyo
Kashiwa-shi, Chiba, Japan
hoshino@cc.u-tokyo.ac.jp

Toshihiro Hanawa
The University of Tokyo
Kashiwa-shi, Chiba, Japan
hanawa@cc.u-tokyo.ac.jp

Akihiro Ida
The University of Tokyo
Bunkyo-ku, Tokyo, Japan
ida@cc.u-tokyo.ac.jp

EXTENDED ABSTRACT

Over the last decade, processor performance has mainly been improved by increasing the number of cores, and the high-performance computing field is correspondingly shifting from multi- to many-core processors. On the other hand, the growth of memory performance is relatively slow. As a result, especially in memory-bounded applications, the remaining cores, which do not contribute to the performance or are detrimental, have been appeared. For example, Intel Knights Landing (KNL) consists 68 cores with 16GB fast memory (MCDRAM) and 96GB DDR4 memory. The memory throughput of KNL will be saturated by 60 cores even if MCDRAM is used, that is, 8 cores are un-used cores.

In this research, we consider optimization techniques to use these un-used cores effectively. As a case study, we optimize an $\mathcal{H}$-matrix-vector multiplication on KNL cluster. The Byte/Flop ratio of $\mathcal{H}$-matrix vector computation is $8/2 = 4$ while KNL's ratio is about 0.2, therefore the computation is memory-bounded.

$\mathcal{H}$-matrices are an approximation technique for dense matrices, such as the coefficient matrix of the boundary element method (BEM). An $\mathcal{H}$-matrix is expressed by a set of low-rank approximated and small dense sub-matrices, each of which has various ranks. Figure 1 shows the structure of low-rank matrices including $\mathcal{H}$-matrices. To simplify MPI communication, we use Lattice $\mathcal{H}$-matrix [1] instead of $\mathcal{H}$-matrix for distributed memory cluster.

To perform an $\mathcal{H}$-matrix-vector multiplication by using MPI + OpenMP hybrid parallel programming model, 2 reduction operations are required. Figure 2 shows the calculation patterns of $\mathcal{H}$-matrix-vector multiplication with MPI + OpenMP. Lattice blocks will be distributed to MPI processes and the computation of inner lattice block will be parallelized by OpenMP threads. As shown in Fig. 2, each OpenMP thread creates local result vector. To obtain the result vector b, we have to reduce those local result vectors in a MPI process. As a straight forward way, we can use atomic operation, which often causes performance degradation, to reduce the vectors. After that, we have to collect the local result vector of a MPI process.

We propose to use un-used cores for the reduce operations instead of atomic operation. By controlling the size of lattice, the reduce calculation can be executed on L1/L2 cache of un-used cores. In addition, we propose to leave MPI communications to un-used cores to overlap the calculation time and communication time.

In the poster presentation, we will show the detail of optimization and the experimental result of $\mathcal{H}$-matrix-vector multiplication on KNL cluster.

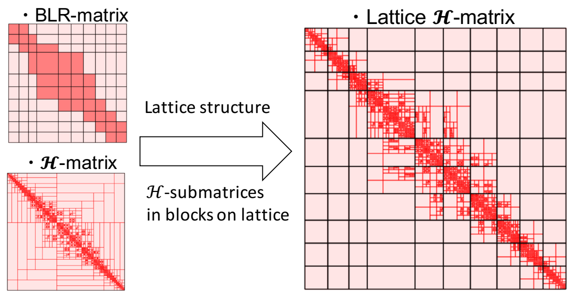
・BLR-matrix
・$\mathcal{H}$-matrix
Lattice structure
$\mathcal{H}$-submatrices in blocks on lattice
・Lattice $\mathcal{H}$-matrix

Figure 1: The structures of $\mathcal{H}$-matrix, BLR-matrix, and lattice $\mathcal{H}$-matrix. Blocks painted in deep red and light red show dense and low-rank sub-matrices, respectively. Lattice $\mathcal{H}$-matrix is a combination of BLR-matrix and $\mathcal{H}$-matrix.

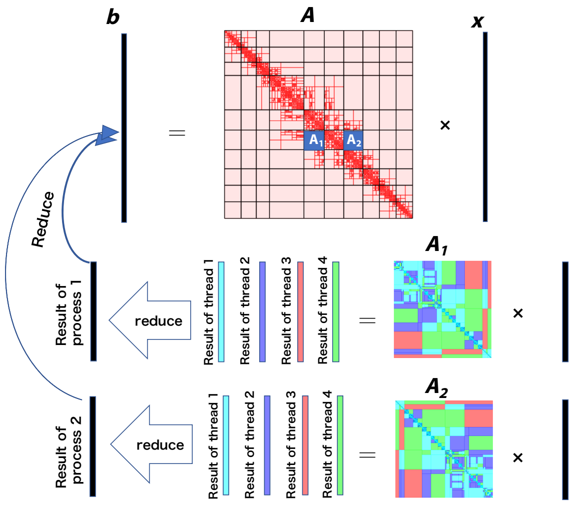

Figure 2: Calculation pattern of Lattice $\mathcal{H}$-matrix-vector multiplication ($b = Ax$). To calculate b, reduce type computations are required.

REFERENCES

[1] Akihiro Ida. Lattice H-matrices on distributed-memory systems (in press). In *2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2018.