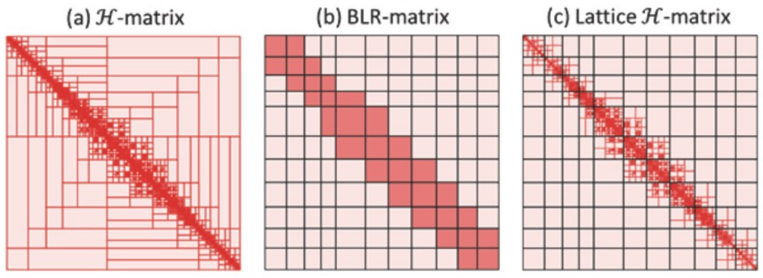
A Communication Overlapping Method for Lattice $\mathcal{H}$-Matrix-Vector Multiplication on GPU Clusters

Naoki Momotake[1], Tetsuya Hoshino[2], Akihiro Ida[3], Masatoshi Kawai[4], So Ozawa[5], Ryosuke Ando[6], Takahiro Katagiri[2]
1. Graduate School of Informatics, Nagoya University 2. Information Technology Center, Nagoya University
3. Research Institute for Value-Added-Information Generation (VAiG), Japan Agency for Marine-Earth Science and Technology (JAMSTEC)
4. Cyberscience Center, Tohoku University 5. Earthquake Research Institute University of Tokyo 6. Graduate School of Science, University of Tokyo

1. Introduction and Motivation

Lattice $\mathcal{H}$-Matrix

- The Boundary Element Method requires dense coefficient matrices, making low-rank approximation essential for large-scale simulations.
- **Lattice $\mathcal{H}$-Matrix**[1] is a low-rank approximation technique designed for large-scale parallel environments, combining the advantages of :
  - $\mathcal{H}$-matrix – Memory usage of $O(N \log N)$.
  - BLR matrix – Regular lattice structure.
- Near-diagonal blocks typically do not admit low-rank approximation as they represent interactions between geometrically close elements.

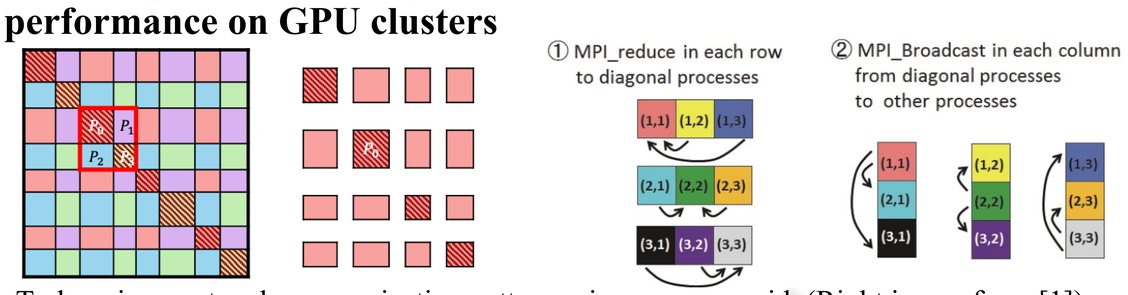


Examples of matrix structures.
Dark red : dense submatrices. Light red :low-rank submatrices. (From [1])

Conventional Method and Challenge

- Task assignments by 2D process grids for iterative solvers face a trade-off between load balance and communication costs.
  - Square grid : Efficient communication pattern, but poor load balance as diagonal blocks are assigned exclusively to diagonal processes.
  - Rectangular grid : Better load balance, but higher communication cost as efficient square grid patterns are inapplicable.
- Communication overlap was not implemented.
- Evaluations were limited to pure-MPI implementations on CPU clusters.

- **Objective : Propose a communication overlapping method for iterative lattice $\mathcal{H}$-matrix-vector multiplication (LHMVM), enabled by a new task assignment that improves load balance, and evaluate its performance on GPU clusters**



Task assignment and communication pattern using a square grid. (Right image from [1])

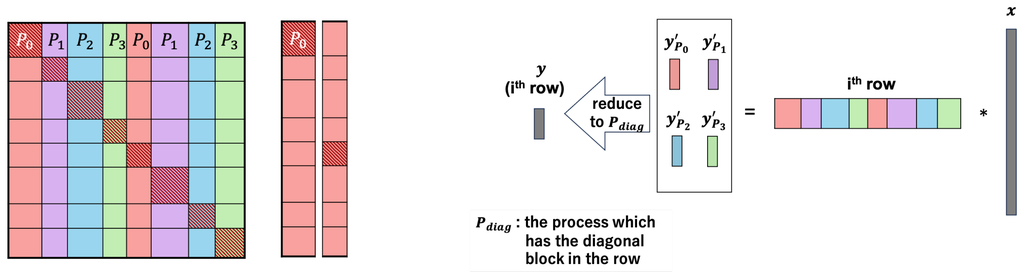
2. Proposed Method

Communication Overlap

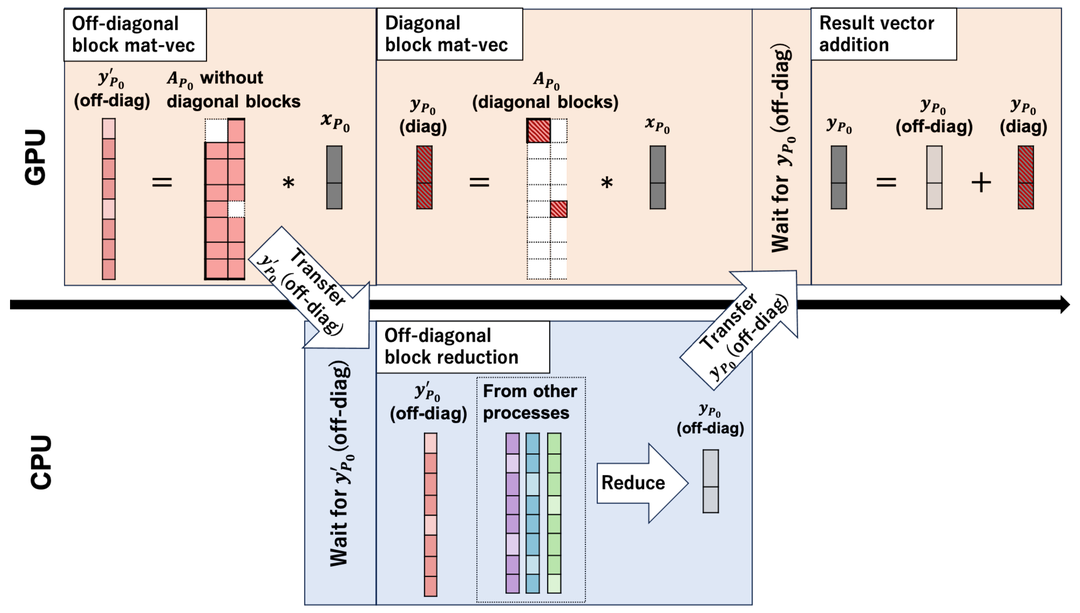
- **Method : Overlap MPI communication for off-diagonal blocks with GPU computation of diagonal blocks**
- **Simplified Workflow**
  1. GPU: Compute off-diagonal blocks (y'(off−diag)).
  2. CPU: Communication (Reduce off-diagonal results and get y(off−diag)).
     GPU: Compute diagonal blocks (y(diag)).
  3. GPU: Sum results ($y = y$(off−diag) + y(diag)).
- **Advantage over Naïve Row-by-Row Overlap**
  - Kernel launches and synchronizations after computations are reduced from the number of lattice rows to just 2.

Task Assignment

- **Method : 1D Column-Cyclic Task Assignment**
  - Row-wise communication cost increases as the number of communicating processes grows due to the assignment change.
- **Communication Pattern : Row-wise reduction rooted at the process owning the diagonal block**
  - This communication pattern eliminates the need for a subsequent broadcast, as the receiving process is the direct consumer of the data.
- **Merits**
  - Diagonal blocks are evenly distributed across all processes, improving load balance and ensuring the effectiveness of the proposed overlap method.
  - Eliminates column-wise communication.



The proposed task assignment and communication pattern.



Workflow of the proposed MPI communication overlapping method with the new task assignment.

3. Evaluation

Evaluation Setup

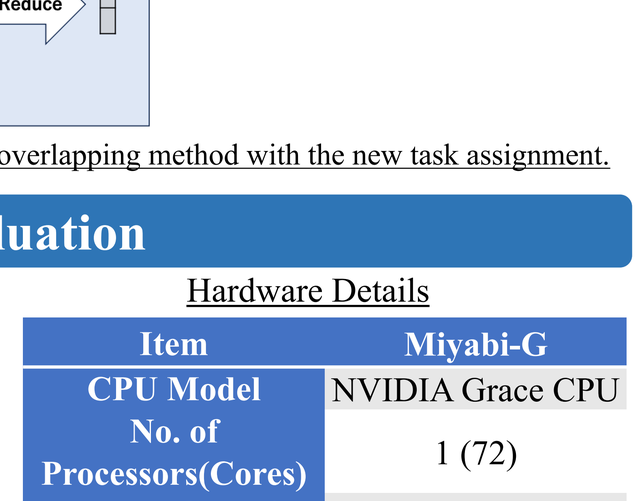
- **System :** Miyabi-G (NVIDIA GH200 Grace Hopper Superchip)
  - 1GPU / MPI process
- **Application :** Earthquake simulation (HBI[2])
- **Problem :** 1M, 2.25M, 4M elements
  - 100 time steps

Hardware Details

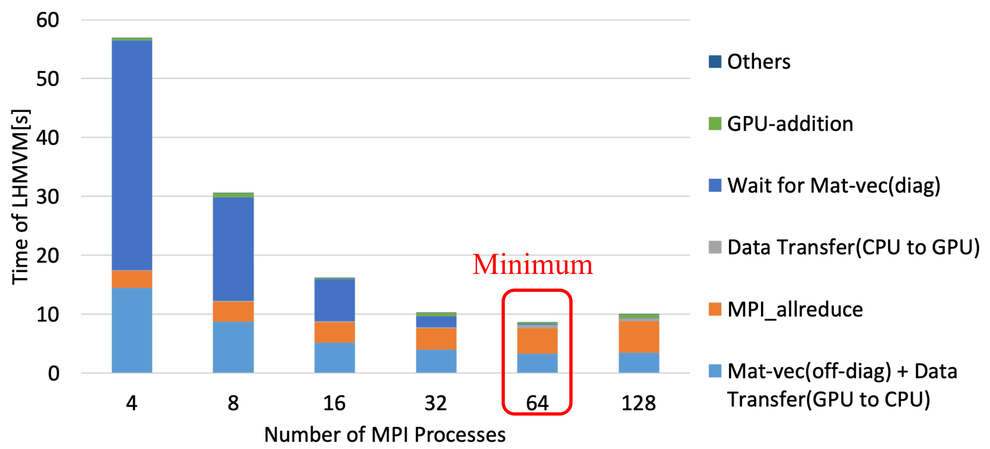
Item	Miyabi-G
CPU Model	NVIDIA Grace CPU
No. of Processors(Cores)	1 (72)
Memory Capacity	120GB
Memory Bandwidth	512GB/s
Inter-node interconnect	InfiniBand NDR (200Gbps)
GPU Model	NVIDIA Hopper H100 GPU
GPU Memory Capacity	96GB
GPU Memory Bandwidth	4.02TB/s

Result

- **Load Balance**
  - Efficiency Metric E : the ratio of average process memory usage to maximum process memory usage (where 1.0 is ideal)
  - The proposed assignment improved E from 0.4 (square grid) to 0.9.
- **Performance Comparison**
  - Achieved up to **3.08x speedup** with 1D column-cyclic + diag overlap over the Baseline.
    - MPI_Allreduce is used here as it outperformed MPI_Reduce.
  - The naïve row-by-row overlap performed poorly due to high overhead from frequent kernel launches and synchronizations.
- **Scaling analysis**
  - Achieved effective overlap of communication costs at 64 processes (Wait time ≈ 0).
  - At 128 processes, communication costs began to dominate the diagonal computation time.



Comparison of LHMVM execution time across different methods.
(16 MPI processes, 2,25M elements)



Scalability of the proposed overlap method with the new task assignment. (4M elements)

4. Conclusion and Future Work

Conclusion

- Proposed a communication overlapping method for LHMVM on GPU clusters, enabled by 1D column-cyclic task assignment.
- Demonstrated a maximum of 3.08x speedup over the baseline.

Future Work

- Investigate the limitations of scalability.
- Evaluate under various conditions like different problem sizes and matrix structures.

References

[1] A. Ida, "Lattice H-Matrices on Distributed-Memory Systems," 2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS), Vancouver, BC, Canada, 2018, pp. 389-398, doi: 10.1109/IPDPS.2018.00049.

[2] So Ozawa, Akihiro Ida, Tetsuya Hoshino, Ryosuke Ando, Large-scale earthquake sequence simulations on 3-D non-planar faults using the boundary element method accelerated by lattice H-matrices, Geophysical Journal International, Volume 232, Issue 3, March 2023, Pages 1471–1481, https://doi.org/10.1093/gji/ggac386.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Numbers JP24K02949 and JP25K00141. It also used the computational resources of the "Miyabi" supercomputer provided by the University of Tokyo through the HPCI System Research Project (Project ID: hp220105, hp250126).