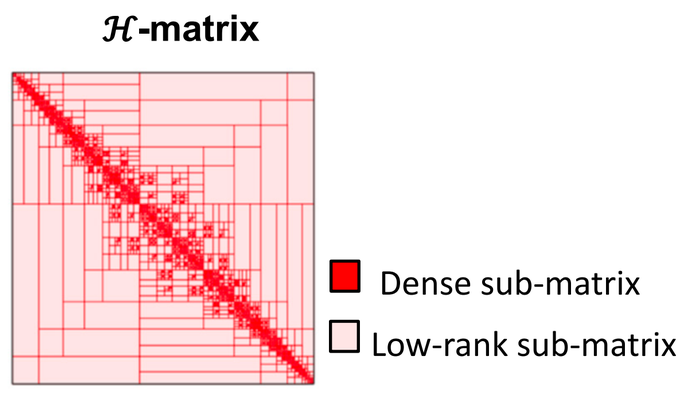
Optimizations of $\mathcal{H}$-matrix-vector Multiplication for Modern Multi-core Processors

Tetsuya Hoshino (Information Technology Center, Nagoya univ., E-mail: hoshino@cc.nagoya-u.ac.jp)

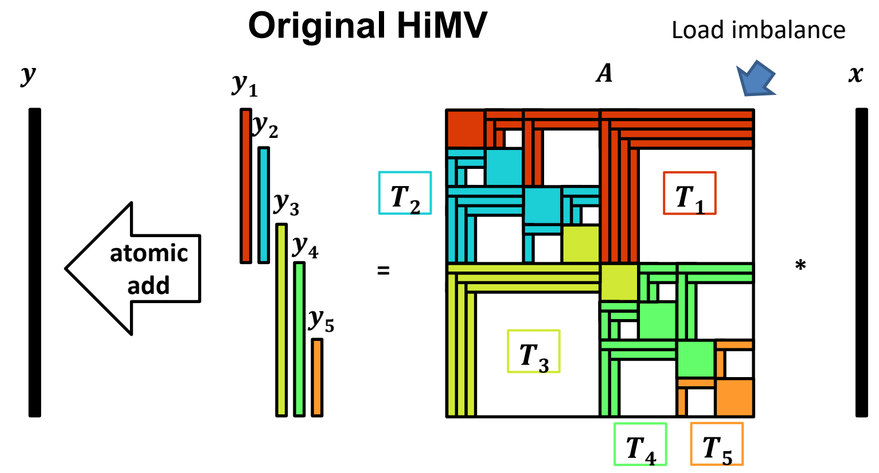
Hierarchical matrices ($\mathcal{H}$-matrices)

- Approximation method for dense matrices such as the coefficient matrix of the boundary element method (BEM)
- Expressed by a set of low-rank approximated and small dense sub-matrices
- Reduce memory $O(N^2)$ -> $O(N \log N)$
- $\mathcal{H}$-matrix-vector multiplication (HiMV) is important kernel
  - ✓ For example, earthquake simulations require >10,000 HiMV executions



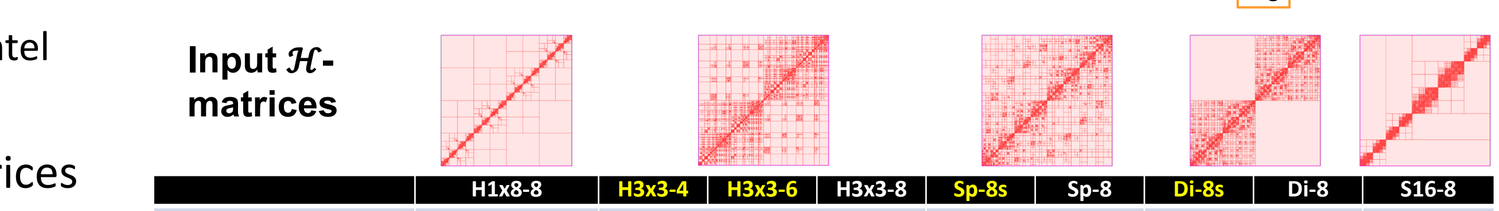
Optimizations of HiMV

- To optimize the HiMV kernel for modern multi-core CPUs, the following optimizations were applied
  1. **firsttouch**: First touch
  2. **contiguous**: Contiguous memory placement by changing from a tree structure to a one-dimensional array
  3. **avoidAtomic**: A reduce algorithm that avoids atomic operations
  4. **balancing**: Inter-thread load balancing
  5. **padding**: Zero padding for aligned memory access
  6. **blocking**: $\mathcal{H}$-matrix Storage for cache blocking
  7. **sort**: Submatrix sorting for efficient cache usage



Experimental results

- Evaluated on modern multi-core processor
  - ✓ A64FX with HBM2, AMD EPYC Rome, Intel Xeon CascadeLake
- Evaluated by valuable types of $\mathcal{H}$-matrices appearing in electrostatic field analysis
  - ✓ **H1x8**: Humanoid objects lined up in one dimension
  - ✓ **H3x3**: Humanoid objects aligned in two dimension
  - ✓ **Sp**: Single spherical object (a three-dimensional example)
  - ✓ **Di**: Two spherical object, such as the Earth and Moon
  - ✓ **S16**: A tetrahedron with 16 objects (four near each vertex)



	H1x8-8	H3x3-4	H3x3-6	H3x3-8	Sp-8s	Sp-8	Di-8s	Di-8	S16-8
Shape	Humanoid 1x8	Humanoid 3x3			Single Sphere		Dipole		Spheres 16
$\mathcal{H}$-matrix Accuracy	10^{-8}	10^{-4}	10^{-6}	10^{-8}	10^{-8}		10^{-8}		10^{-8}
# unknowns	157,312	176,976			20,000	200,000	21,600	160,000	216,000
# low rank sub-mats	129,678	139,588			16,640	182,176	17,002	149,426	195,178
# dense sub-mats	243,328	288,876			29,480	242,976	33,096	248,864	322,800
(MAX, AVE) rank	(70,21)	(22,8)	(44,14)	(75,22)	(34,16)	(35,14)	(53,17)	(52,16)	(53,16)
Memory [GB]	4.243	3.293	6.045	9.695	0.460	6.791	0.453	5.148	6.491
Memory vs. dense [%]	2.143	1.314	2.412	3.869	14.38	2.122	12.14	2.514	1.739

Evaluation results on single socket A64FX, Rome, and CascadeLake

Summary

- By applying the series of optimizations, we obtained the following performance
  - ✓ **A64FX:** 57.9 ~ 84.8% of dgemv
  - ✓ **EPYC Rome:** 93.2 ~ 100.7% of dgemv
  - ✓ **CascadeLake:** 97.6 ~ 98.7% of dgemv
- We found that for A64FX with HBM2, optimization to increase the efficiency of cache utilization is important

For more information, please refer to the following paper: T. Hoshino, A. Ida and T. Hanawa, "Optimizations of H-matrix-vector Multiplication for Modern Multi-core Processors," (CLUSTER 2022)

Information Technology Center, Nagoya University
http://www.icts.nagoya-u.ac.jp/en/center/