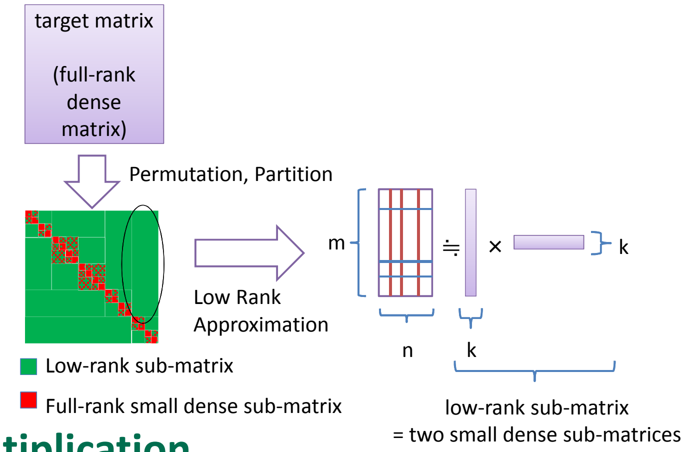
Numerous small dense-matrix-vector multiplications on GPU toward to approximate matrix method

Satoshi Ohshima (Information Technology Center, Nagoya University, E-mail: ohshima@cc.nagoya-u.ac.jp)

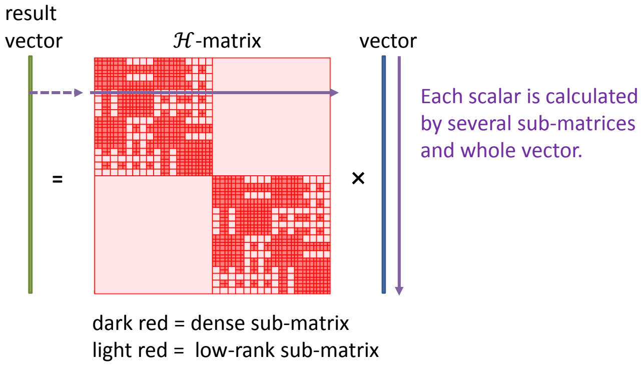
Background

- The demand for dense matrix computation in large scale and complex simulations is increasing. However, memory capacities of current computer systems are very limited. One of the solutions is approximation techniques.
- H-matrices is one of the approximation matrix method. Target matrix is approximated by many low-rank sub-matrices and full-rank small dense sub-matrices.
- We want to use H-matrix in BiCGSTAB method. Matrix-vector multiplication is a dominant part of this method. Thus, we focus on H-matrix - vector multiplication.



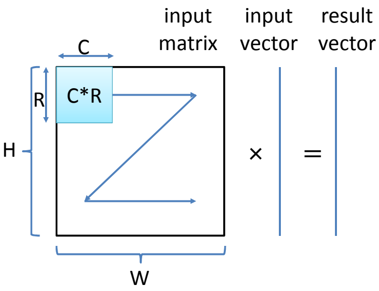
Numerous small dense-matrix-vector multiplication

- H-matrix vector multiplication consists of numerous small dense-matrix-vector multiplications. To calculate result vector, all low-rank sub-matrix - vector multiplications and small dense sub-matrix - vector multiplications are calculated and merged.
- To obtain high calculation performance of GPU, high parallelism is required. However, each sub-matrix - vector multiplication is too small to fill the computation cores of GPU.
- Solution: calculate many sub-matrix - vector multiplications at once
- To obtain high performance, how to assign multiplications to GPU?
- While BATCHED BLAS libraries provide this calculation, there are room for improvement for specific applications.


result vector
$\mathcal{H}$-matrix
vector
=
×
Each scalar is calculated by several sub-matrices and whole vector.
dark red = dense sub-matrix
light red = low-rank sub-matrix

Optimization parameters

- **T**: #WAPS = 32*T threads / 1Block
  - gpukernel<<<blk, T*32>>>(……);
- **C**: continuous C threads calculate one line
- **R**: R-row * C-column cores calculate 1 matvec
- (T*32) / (C*R) matvec are calculated concurrently
- variations of C, R, T = 1, 2, 4, 8, 16, 32
- total 216 cases (6*6*6), which is the best parameter?



Performance evaluation

- Measured the performance of dense sub-matrices of H-matrix - vector multiplications on Tesla P100.
- Target matrices and optimal parameters:

matrix name	number of dense sub-matrices	MByte	Optimal parameters
10ts	14,860	136	C=8, R=4, T=2
216h	33,096	295	C=4, R=8, T=4

human_1x1	30,416	298	C=2, R=16, T=1
			C=8, R=4, T=2

100ts	132,740	2,050	C=8, R=4, T=2

  - All matrices are generated from electric field analysis problems. Only the dense sub-matrices are calculated.
- No combinations obtained the best performance at all target matrices.
- We proposed combined kernel which uses multiple combinations at one GPU kernel.

Publication

- Satoshi Ohshima, Ichitaro Yamazaki, Akihiro Ida, Rio Yokota, “Optimization of Numerous Small Dense-Matrix-Vector Multiplications in H-matrix Arithmetic on GPU”, 2019 IEEE 13th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC), pp.9-16, 2019.
  - Presentation slide is available on online: http://mcsoc-forum.org/m2019/ieee-mcsoc-2019-presentation-slides/
  - Key implementation is available on online: https://github.com/exthnet/hacapk_hmvm

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