階層的低ランク近似法の最新研究動向と応用例

東京工業大学
学術国際情報センター
横田理央

第22回AT研究会オープンアカデミックセッション（ATOS22）
東京大学情報基盤センター4F遠隔会議室，2019年5月13日

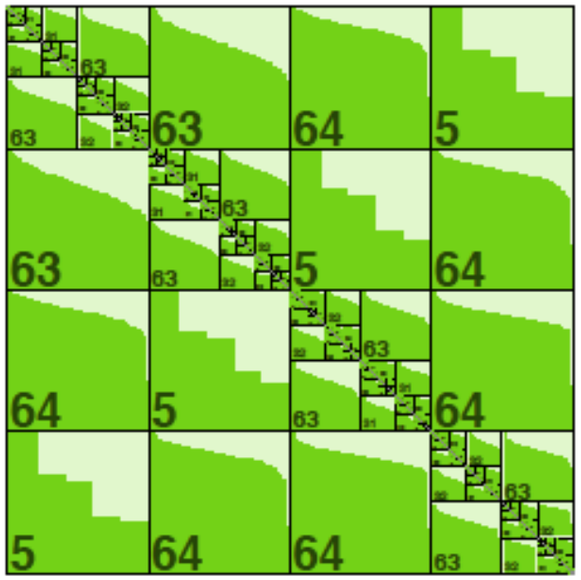
階層的低ランク近似法とは？

密行列の近似直接解法

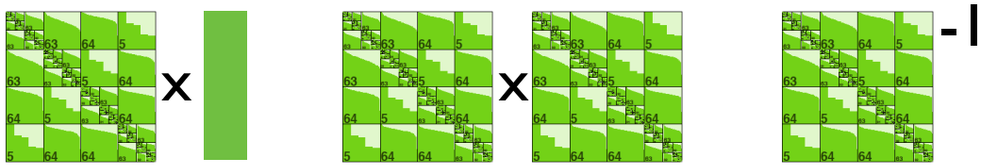
演算量：$\mathcal{O}(N^3) \longrightarrow \mathcal{O}(N \log^2 N)$

メモリ：$\mathcal{O}(N^2) \longrightarrow \mathcal{O}(N)$

非対角ブロックのランクは小さいという仮定

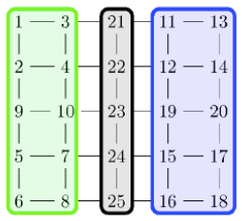


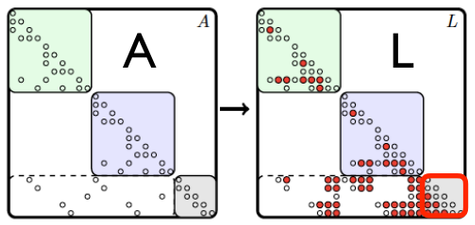
疎行列に使えるのか？



Fill-inさせないことの方が重要

Schur補元は密だが非対角ブロックのランクは小さい





Schur complement

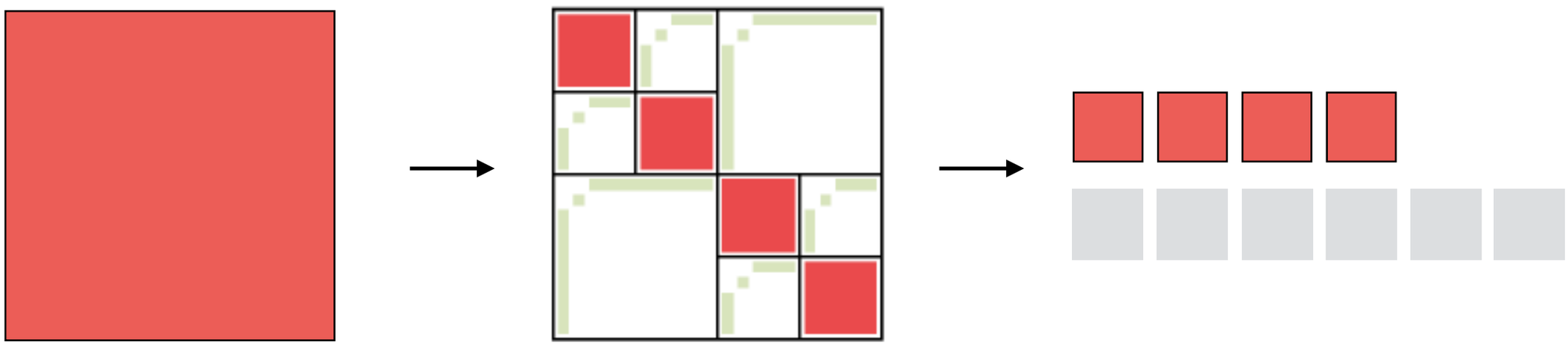
反復法では？

前処理として使えるがMultigridにはなかなか勝てない

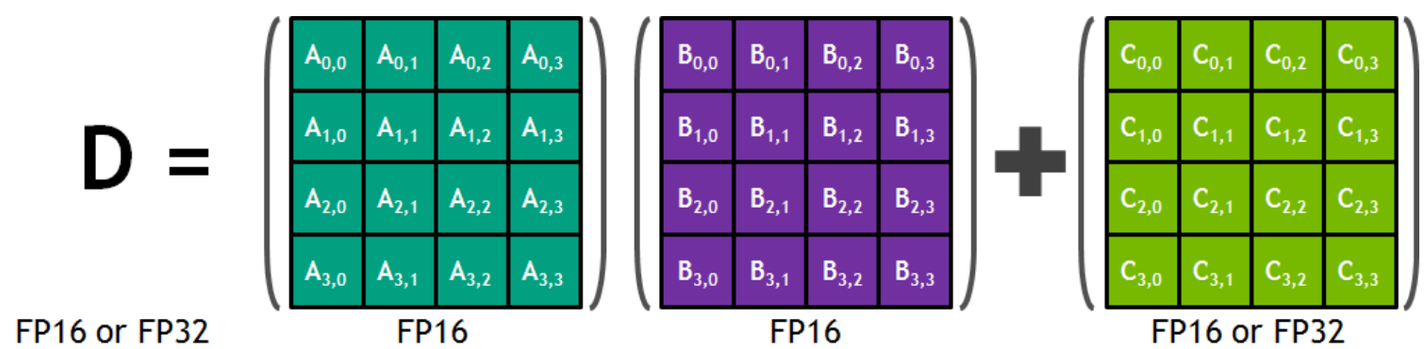
条件数が悪い問題でなら優位かも？

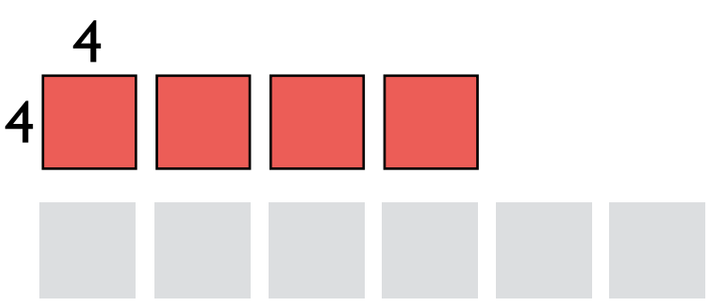
機械学習向けハードウェアと相性が良い

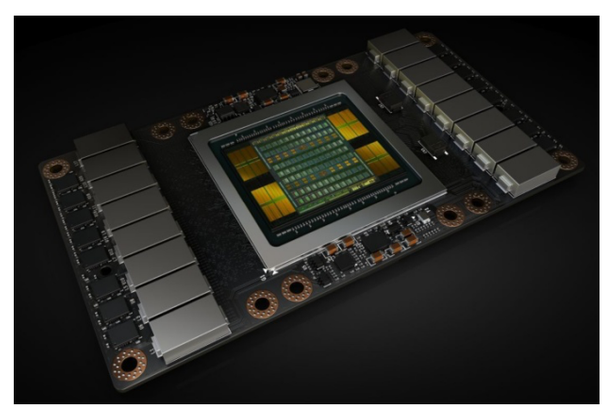
小さな密行列演算がたくさん生じる



低ランク近似なので低精度演算で十分

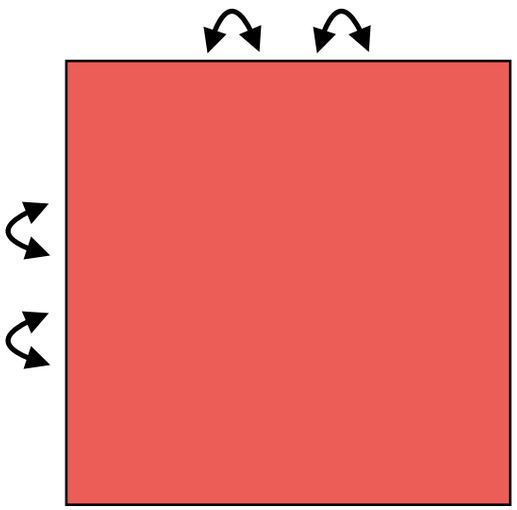






階層的低ランク近似法の3つのステップ

並べ替え



<u>最小化したいのは？</u>

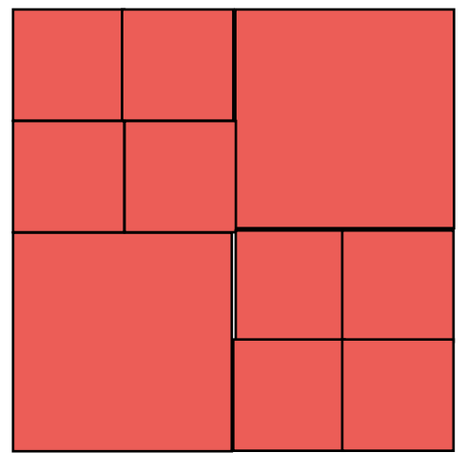
ランク (幾何学的距離)

通信 (データ局在性)

Fill-in (グラフの接続)

→普通は近いものと繋がっているのでFill-inとランクの最小化は両立する

階層化



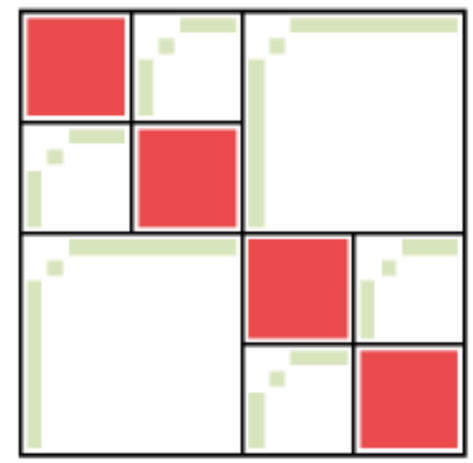
<u>どこまで分割するか？</u>

分割数を増やせばそれぞれのランクは小さくなる

ランクは固定で分割によって精度を制御することもできる

→SIMD friendly

低ランク近似



<u>速さか，安定性か？</u>

ACAは速いが不安定

RSVDは安定だが遅い

どのブロックを近似するかを選択するのにGram距離を使うこともできる

階層的低ランク近似の応用例

Green関数の境界・体積積分（Laplace, Helmholtz）

Laplace BEM GCA [Börm, Christophersen 2018]
Maxwell volume integrals [Ma, Jiao 2018]
Micromagnetics [Ida, Ataka, Takahashi, Mifune, Iwashita, Furuya 2018]
Helmholtz BEM preconditioner [Takahashi, Coulier, Darve 2017]
Variable coefficient Poisson [Pouransari, Coulier, Darve 2017]
Laplace double layer, 3-D complex geometry [Corona, Rahimiam, Zorin 2017]
All eigenvalues & eigenvectors [Vogel, Xia, Cauley, Balakrishnan 2016]

正則化回帰モデル（Gaussian）

Gaussian process [Litvinenko, Ying, Genton, Keyes 2019]
Kernel ridge regression distributed GOFMM [Yu, Reiz, Biros 2018]
Kernel ridge regression GOFMM [Yu, Reiz, Levitt, Biros 2017]
Kernel ridge regression [Yu, March, Biros 2017]
Gaussian process [Akbudak, Ltaief, Mikhalev, Keyes2017]
Kernel summation in high dimension parameter space [March, Xiao, Biros 2016]
Gaussian process [Minden, Damle, Ho, Ying 2016]
Kernel based scattered data interpolation [Iske, LeBorne, Wende 2016]
Gaussian process [Ambikasaran, Foreman-Mackey, Greengard, Hogg, O'Neil 2015]
Kalman filter [Li, Ambikasaran, Darve, Kitanidis 2014]

実装一覧

コード名	手法	開発者	url
AHMED	H-matrix	M. Bebendorf	https://github.com/xantares/ahmed
ASKIT	FMM	C. D. Yu	http://padas.ices.utexas.edu/libaskit
DMHM	H-matrix	J. Poulson	https://bitbucket.org/poulson/dmhm/src/default/
GOFMM	H^2-matrix	C. D. Yu	https://github.com/ChenhanYu/hmlp
H2Lib	H^2-matrix	S. Börm	https://github.com/H2Lib/H2Lib
H2Tools	H^2-matrix	A. Mikhalev	https://bitbucket.org/muxas/h2tools
HACApK	H-matrix	A. Ida	https://github.com/HLRA-JHPCN/HACApK-MAGMA
HiCMA	H-matrix	H. Ltaief	https://github.com/ecrc/hicma
HLib	H-matrix	L. Grasydyck	http://www.hlib.org
HLibPro	H-matrix	R. Kriemann	http://www.hlibpro.com
hmglib	H-matrix	P. Zaspel	https://github.com/zaspel/hmglib
HODLR	HODLR	A. Aminfar	https://github.com/amiraa127/Dense_HODLR
HSS	HSS	J. Xia	http://www.math.purdue.edu/~xiaj/
LoRaSp	H^2-matrix	H. Pouransari	https://bitbucket.org/hadip/lorasp
MUMPS-BLR	BLR	P. R. Amestoy	http://mumps.enseeiht.fr
STURMPACK	HSS	P. Ghysels	http://portal.nersc.gov/project/sparse/strumpack

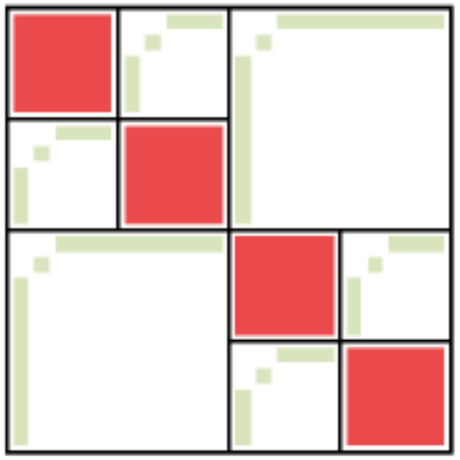
https://github.com/gchavez2/awesome_hierarchical_matrices

手法間の違い

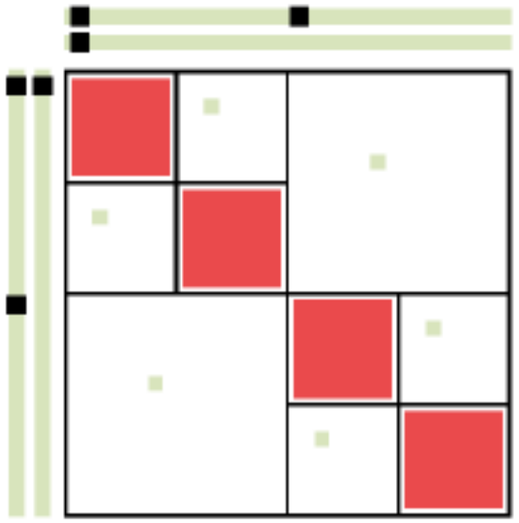
	Nested Basis	Admissibility
H-matrix	No	Strong
H^2-matrix	Yes	Strong
HODLR	No	Weak
HSS	Yes	Weak
BLR	No	non-hierarchical

Nested Basis

Non-nested

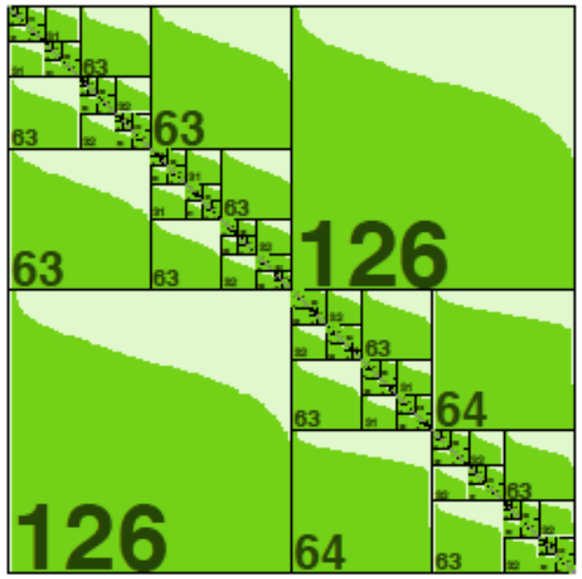


Nested

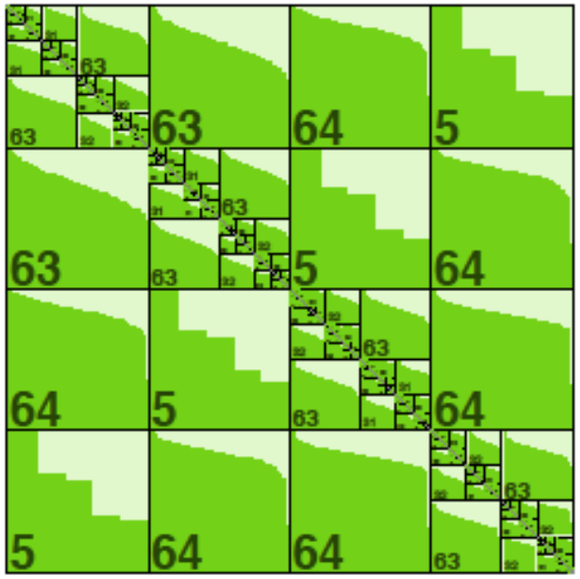


Admissibility

Weak



Strong



勢力図

Germany

- Shared memory H-LU
  Kriemann (2014)
- Nested cross approximation
  Börm & Christophersen (2014)
- H^2-matrix for eigenvalues
  Berner et al. (2015)
- OmpSs H-LU
  Aliaga et al. (2017)
- GCA H^2-matrix
  Börm et al. (2018)

Berkeley

- HSS2D
  Xia (2014)
- HSS selected inversion
  Xia et al. (2015)
- Superfast DC eigenvalue
  Vogel, et al. (2016)
- Shared memory HSS MF
  Ghysels et al. (2016)
- Distributed HSS MF
  Rouet et al. (2016)

Japan

- Distributed H-matrix
  Ida et al. (2015)
- Distributed GPU H-matrix
  Yamazaki et al. (2018)
- Lattice H-matrix
  Ida (2018)
- GPU load-balance ACA
  Hoshino et al. (2018)
- GPU autotuning
  Ohshima et al. (2018)

EPFL

- HODLR QR
  Kressner et al. (2018)

Minnesota

- Multilevel Low-Rank
  Li & Saad (2013)
- DD Low-Rank
  Li & Saad (2014)
- Schur Low-Rank
  Li & Saad (2015)
- Multilevel Schur Low-Rank
  Xi et al. (2016)
- SMASH
  Cai et al. (2018)

Stanford(Ying)

- O(N) RS 2-D
  Corona (2015)
- HIF for PDEs
  Ho & Ying (2016)
- Distributed memory HIF
  Li & Ying (2016)
- RS for maximum likelihood
  Minden et al. (2016)
- RS with strong admissibility
  Minden et al. (2017)
- Quantized Tensor Train
  Corona et al. (2017)

Stanford(Darve)

- HODLR multifrontal
  Aminfar et al. (2016)
- IFMM precond. Stokes
  Coulier et al. (2017)
- IFMM precond. Helmholtz
  Takahashi et al. (2017)
- Extended sparsification IFMM
  Pouransari et al. (2017)
- Non-extensive sparsification
  Sushnikova et al. (2017)
- Sparsified Nested Dissection
  Cambier et al. (2019)

INRIA

- BLR multifrontal
  Amestoy et al. (2015)
- BLR multicore
  Amestoy et al. (2017)
- Multilevel BLR
  Amestoy et al. (2017)

KAUST

- BLR Cholesky
  Akbudak et al. (2017)
- Batched QR, SVD
  Boukaram et al. (2018)
- GPU MatVec
  Boukaram et al. (2019)

Texas (Biros)

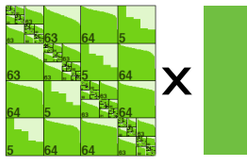
- inv-ASKIT
  Yu et al. (2016)
- Distributed inv-ASKIT
  Yu et al. (2017)
- GOFMM
  Yu et al. (2017)
- Distributed GOFMM
  Yu et al. (2018)

最近の論文の紹介

著者・年	行列	演算	構造	近似法	OpenMP	MPI	GPU
2016Vogel	密	$Q\Lambda Q^T$	HSS	?	×	×	×
2017Akbudak	密	Cholesky	BLR	RSVD	○	×	×
2017Fernando	密	LU	HSS	ID	×	○	○
2017Ghysels	疎	LU	HSS	RSVD	○	○	×
2017Li	疎	LU	HSS	RRQR	×	○	×
2017Minden	疎	LU	H2	RRQR	×	×	×
2018Amestoy	疎	LU	MBLR	?	×	×	×
2018Börm	密	O(N)圧縮	H2	GCA	×	×	×
2018Cai	密	O(N)圧縮	HSS	RRQR	×	×	×
2018Kressner	密	QR	HODLR	?	×	×	×
2018Yu	密	O(N)圧縮	H	ID	○	○	×
2019Amestoy	疎	LU	BLR	?	○	○	×
2019Boukaram	密	MV	H2	RSVD	×	×	○
2019Cambier	疎	LU	H2	RRQR	×	×	×
2019Zaspel	密	MV	H	ACA	×	×	○

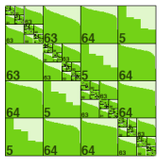
並列化

R. Kriemann (2005), Parallel {H}-Matrix Arithmetics on Shared Memory Systems
R. Kriemann (2015), H-LU factorization on many-core systems



batched MAGMA

LU
QR



StarPU / OmpSs

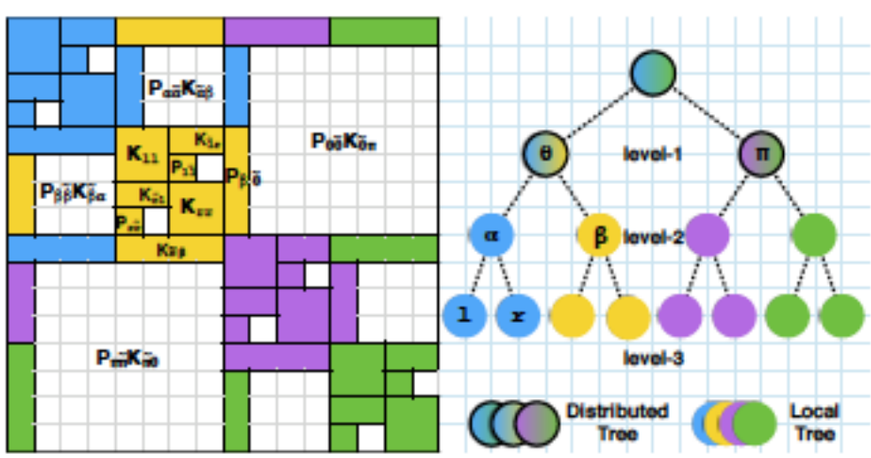
JHPCN

A. Ida
I. Yamazaki
S. Oshima
T. Hiraishi
T. Iwashita
K. Nakajima
T. Aoki
J. Dongarra

共有メモリ

分散メモリ

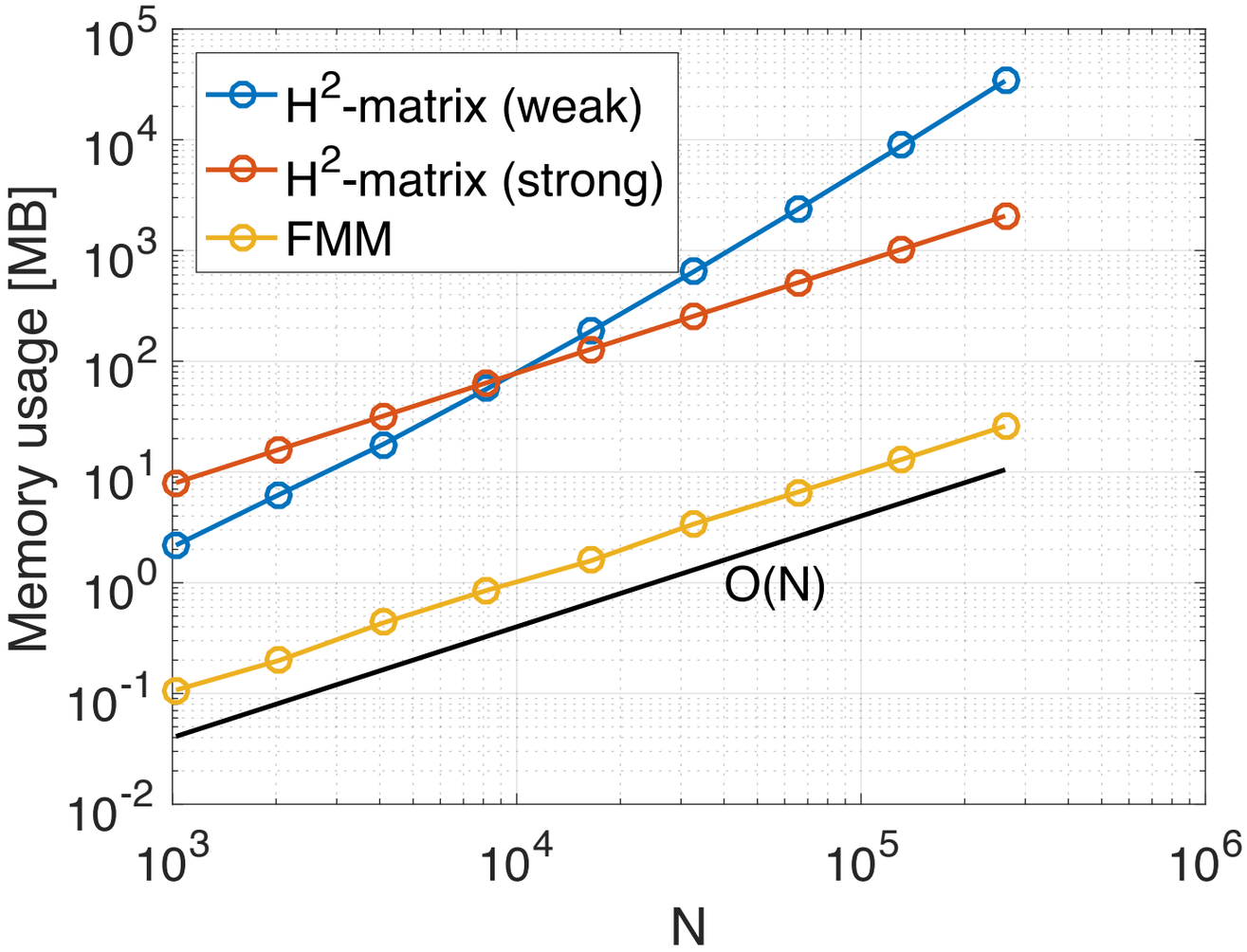
M. Izadi (2012), Hierarchical Matrix Techniques on Massively Parallel Computers
S. Wang (2013), Efficient Scalable Algorithms for Solving Dense Linear Systems with HSS
Y. Li (2016), Distributed-memory Hierarchical Interpolative Factorization
C. D. Yu (2016), INV-ASKIT: A Parallel Fast Direct Solver for Kernel Matrices



	Complexity	Concurrency
BLR	$\mathcal{O}(N^{4/3})$	High
H²(HSS)	$\mathcal{O}(N \log^2 N)$	Low

complexity-concurrency tradeoff

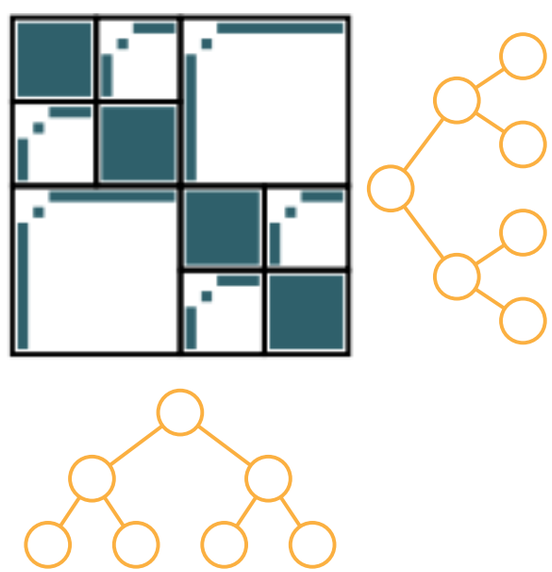
FMMとH²-matrixの関係



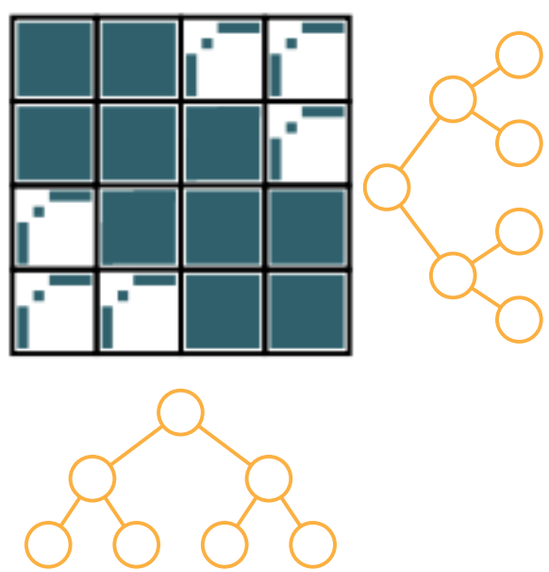
FMMはmatrix-freeのH²-matrix

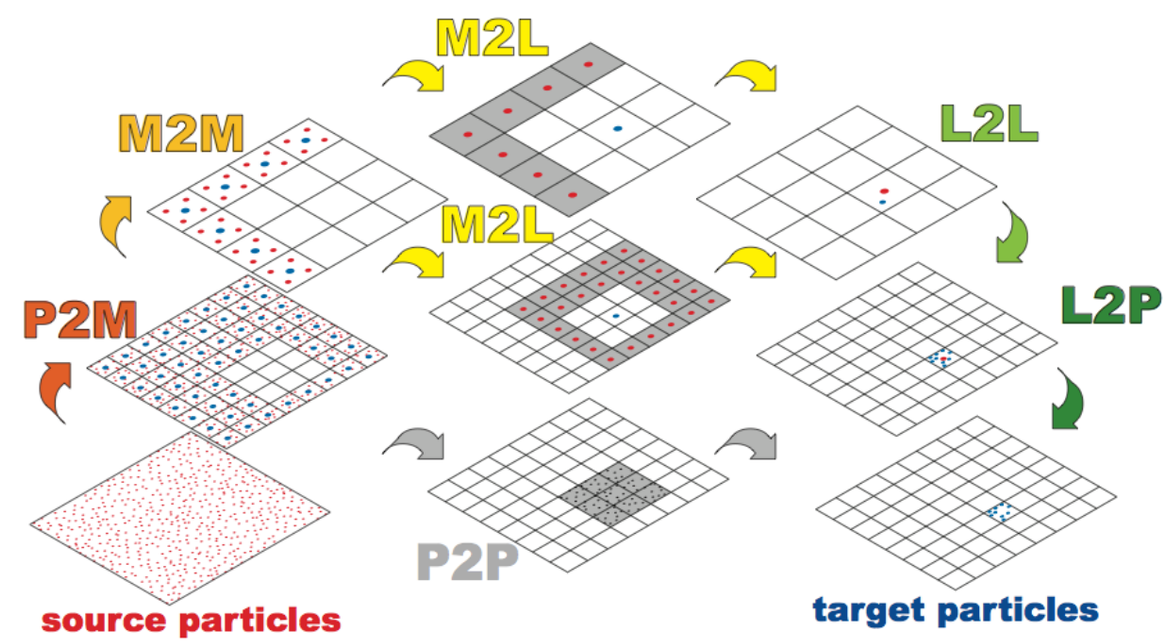
FMMは動径基底関数しか扱えない

weak admissibility



strong admissibility





My new C++ code

C++ Class

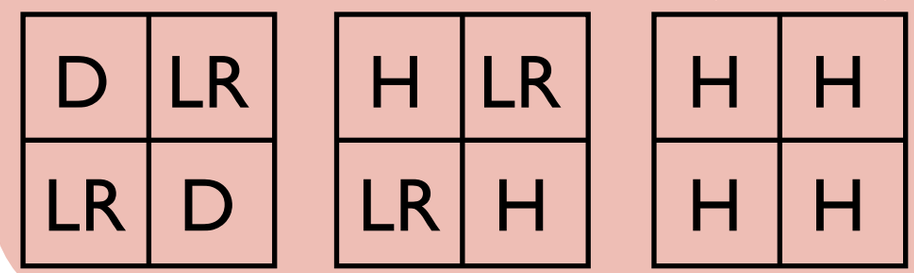
dense | low-rank | hierarchical

Operator overload

```
D=D+D
L=D*L
H=H+H
```

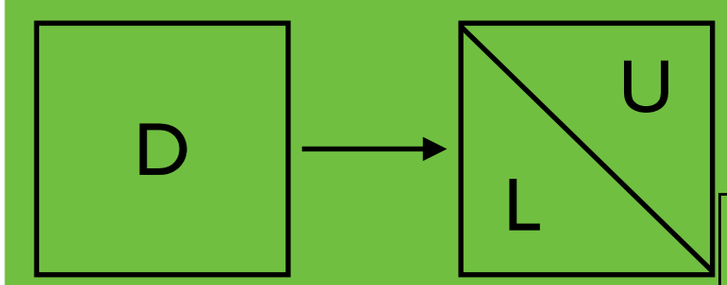
constructor destructor

```
Dense D(N,N)
LowRank LR(D,rank)
Hierarchical H(2,2)
```

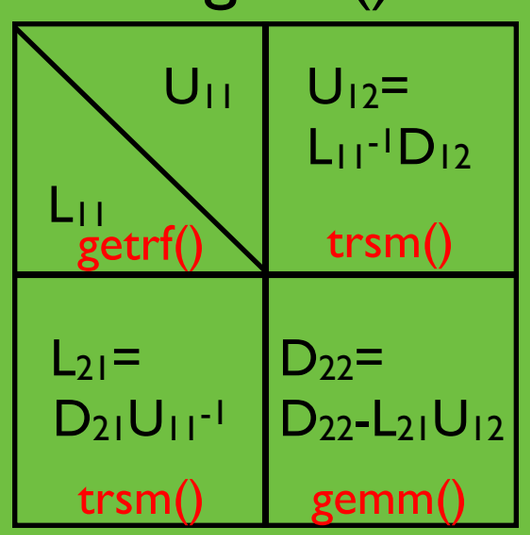


LU decomposition

D.getrf()



H.getrf()



D.trsm(L)

$L_{11}^{-1}D_{12}$

H_D.trsm(H_L)

$$\begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}^{-1} \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} = \begin{bmatrix} L_{11}^{-1}D_{11} & L_{22}^{-1}D_{12} \\ -L_{22}^{-1}L_{21}L_{11}^{-1}D_{11} + L_{22}^{-1}D_{21} & -L_{22}^{-1}L_{21}L_{11}^{-1}D_{12} + L_{22}^{-1}D_{22} \end{bmatrix}$$

D.gemm(L,U)

D_{22}-$L_{21}U_{12}$

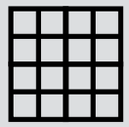
H_D.gemm(H_L,H_U)

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} -= \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}$$

Any H structure

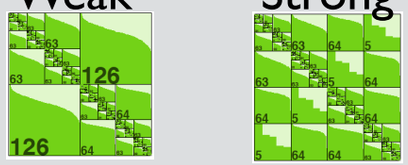
```
Hierarchical H(4,4)
```

BLR too

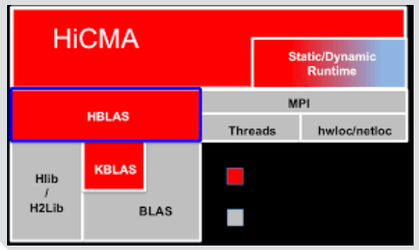


admissibility

Weak | Strong



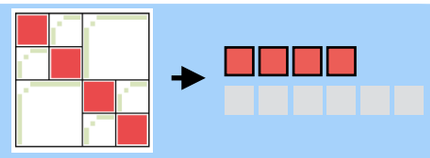

```
Hierarchical A(laplace1d, randx, N, N,
rank, nleaf, admis, nblocks, nblocks);
```



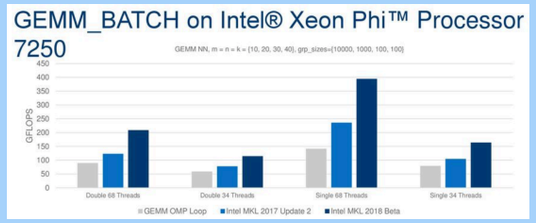
nested basis

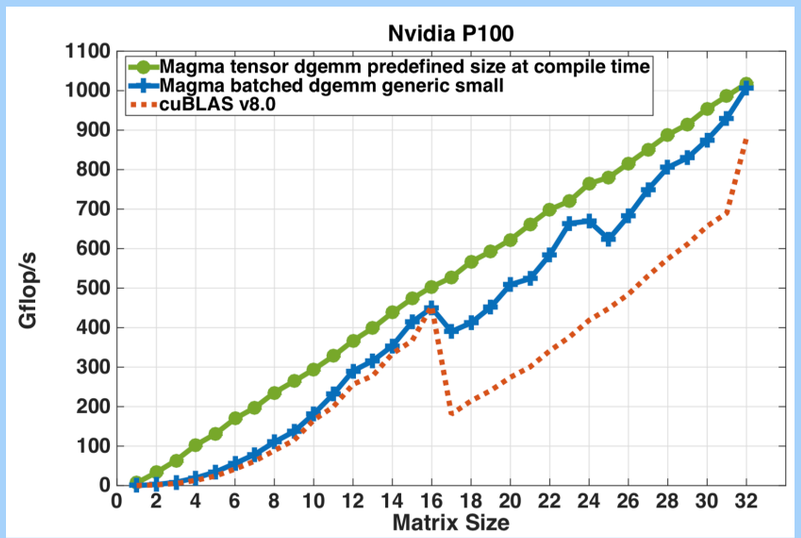
GPU implementation

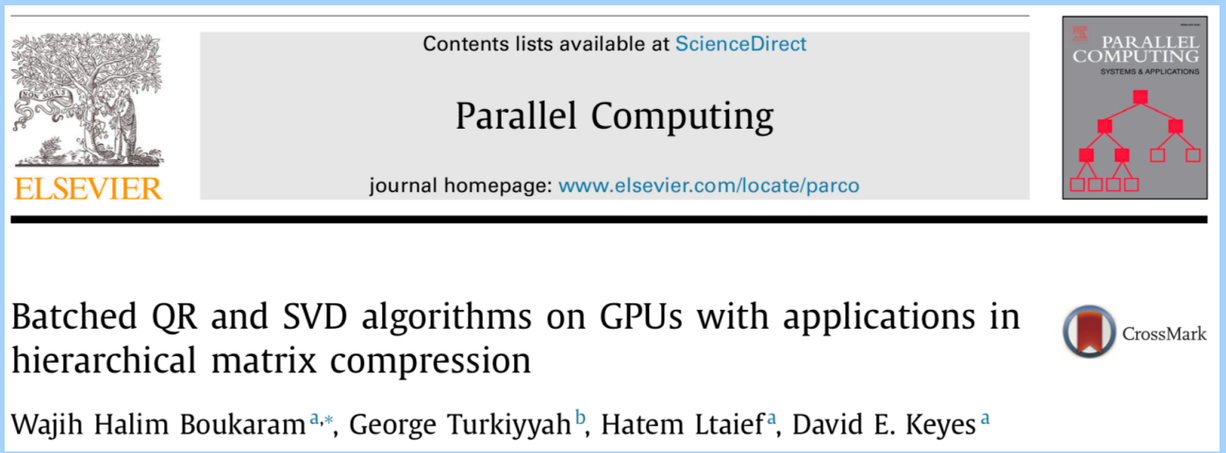
Batch KBLAS



batch GEMM
batch GEMV






Contents lists available at ScienceDirect

Parallel Computing

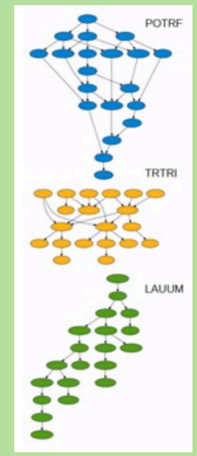
journal homepage: www.elsevier.com/locate/parco

Batched QR and SVD algorithms on GPUs with applications in hierarchical matrix compression

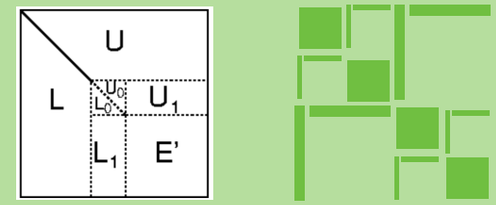
Wajih Halim Boukaram[a,*], George Turkiyyah[b], Hatem Ltaief[a], David E. Keyes[a]

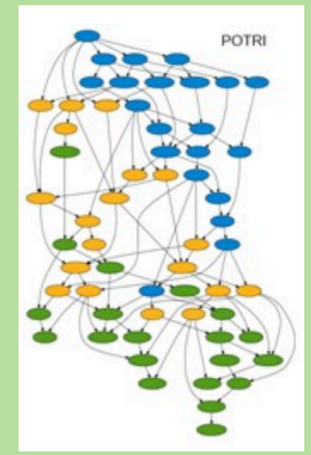
batch QR
batch SVD
batch RSVD
batch ACA (variable M,N,K)

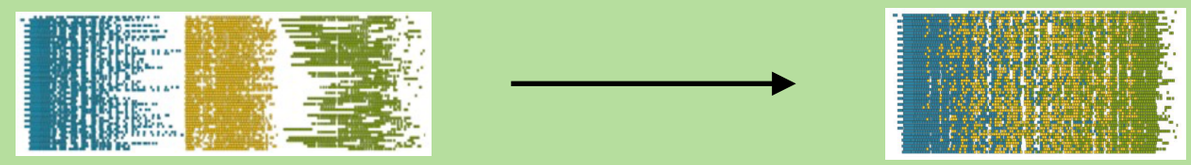
Runtime for LU



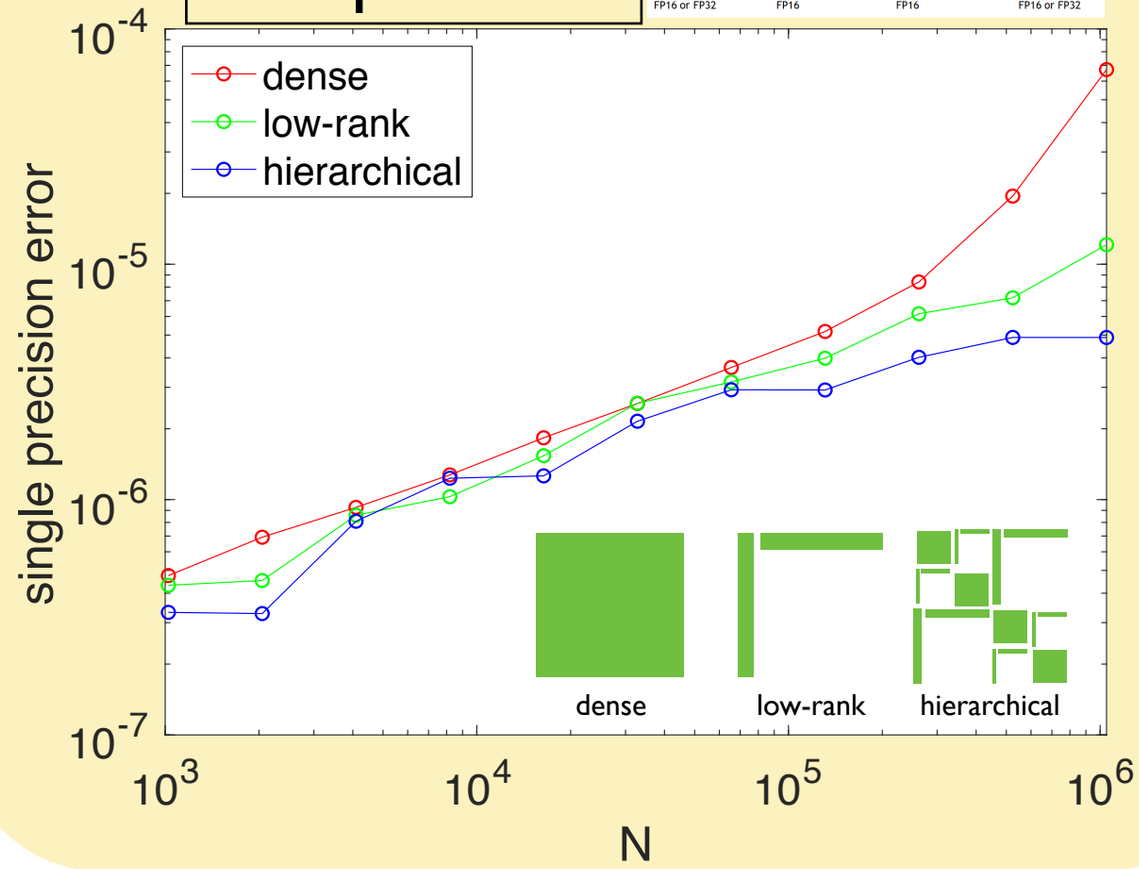
starPU
OmpSs







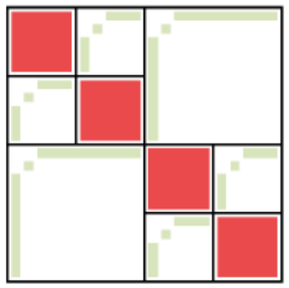
Low precision

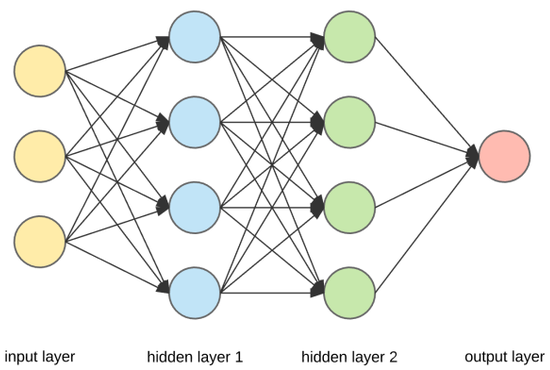


H行列は深層学習に適用できるか？

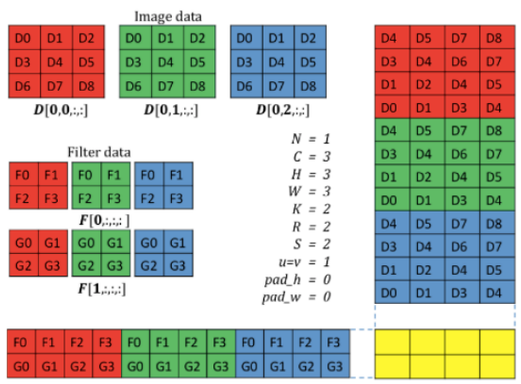
深層ニューラルネットは
密行列演算になる

H行列は密行列の
高速近似解法





畳み込みNNは
密行列積になる



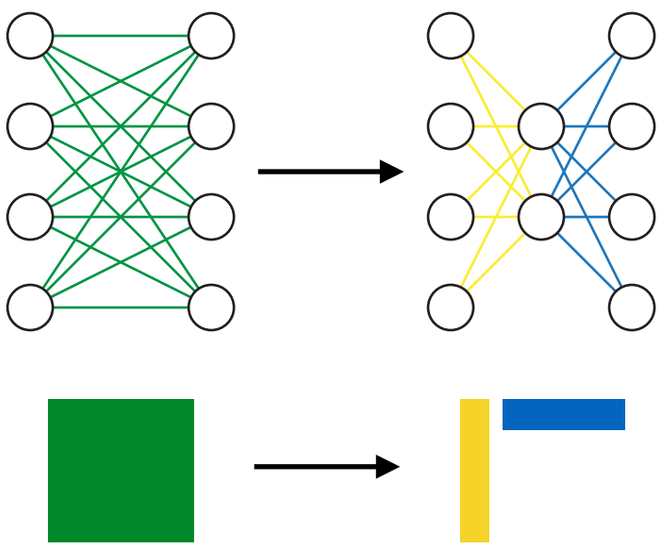
https://arxiv.org/abs/1410.0759

行列が非常に長細い

ブロックのランクが高い

圧縮した行列が１回しか使われない

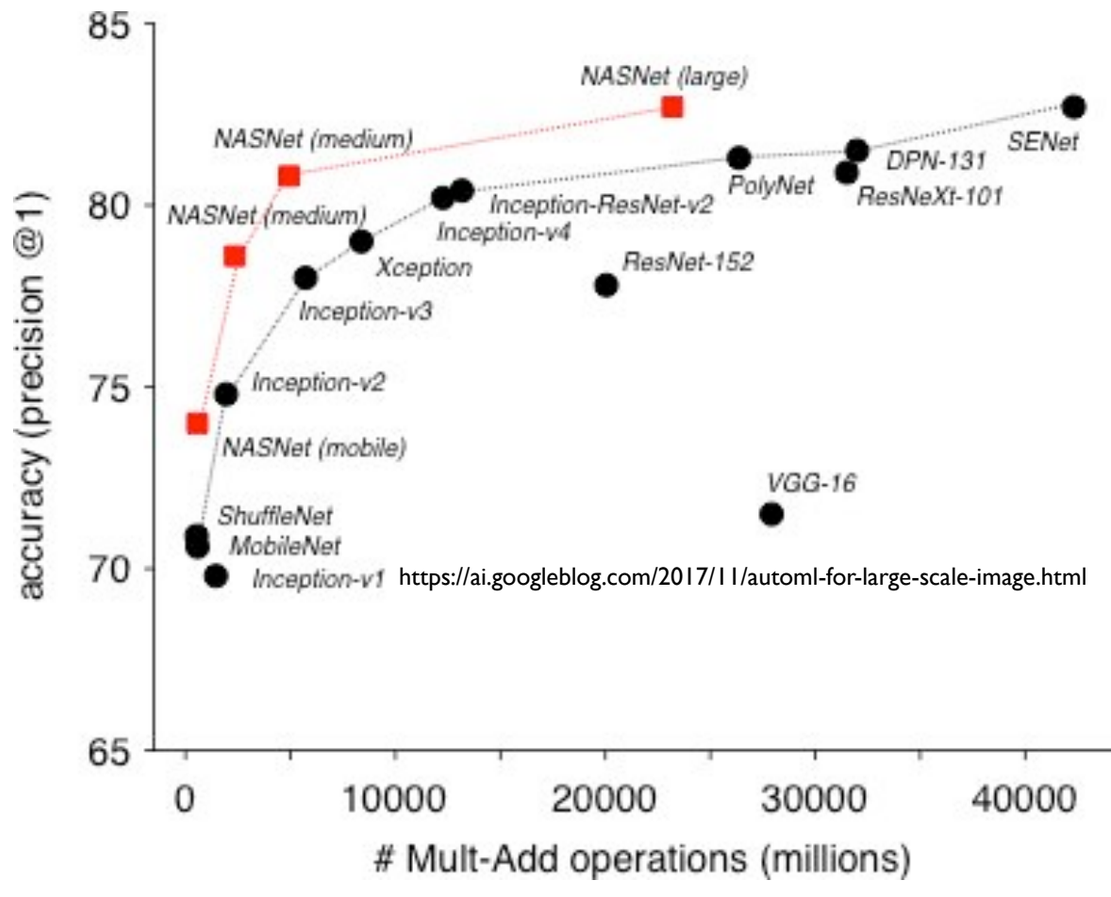
低ランク近似を用いてNN自体を圧縮



汎化性能と敵対的データに対する
堅牢性が維持できるのか？

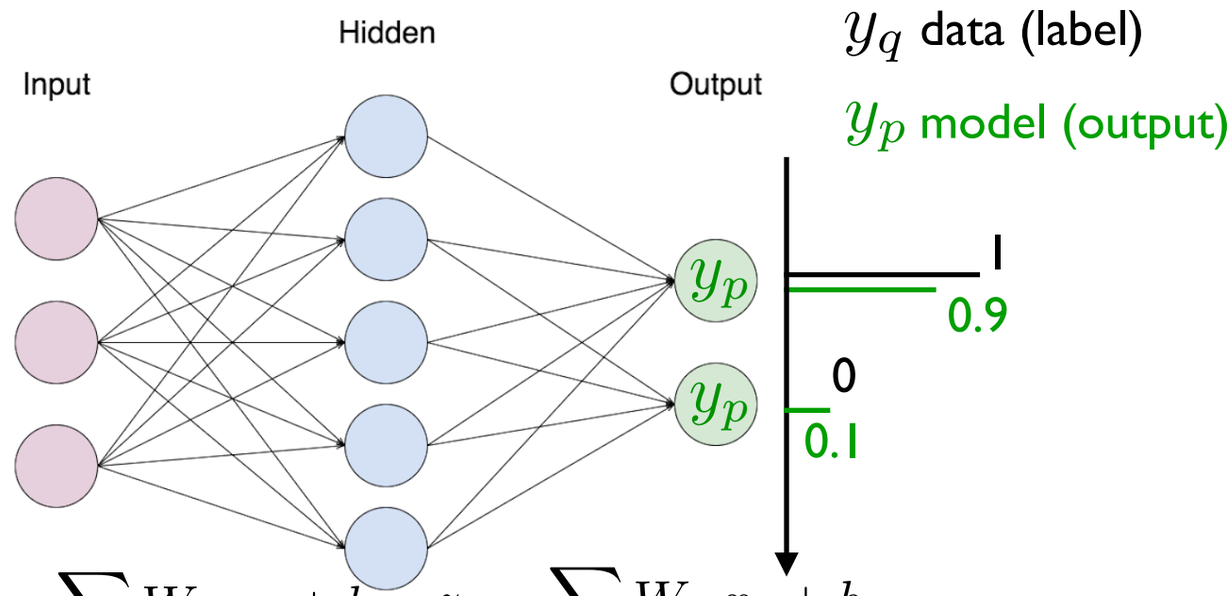
全結合層にしか有効でない

AutoMLの出現



https://ai.googleblog.com/2017/11/automl-for-large-scale-image.html

H行列とKronecker因子分解



$$z_i = \sum_j W_{ij} x_j + b \qquad z_i = \sum_j W_{ij} x_j + b$$

$$x_i = \max(0, z_i) \qquad y_{pi} = \frac{exp(z)}{\sum_i \exp(z_i)}$$

ReLU softmax

Cross entropy loss function

$$J(\theta) = E_q(-\log p_\theta(y|\mathbf{x}))$$
$$= \sum_{(\mathbf{x},y)} -q(y|\mathbf{x}) \log p_\theta(y|\mathbf{x})$$
$$= \sum_{\mathbf{x}} \{-y_q \log y_p - (1-y_q)\log(1-y_p)\}$$
$$= \sum_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \theta)$$

Back propagation

$$\frac{\partial J}{\partial W_{ij}} = \frac{\partial J}{\partial y_{pi}} \frac{\partial y_{pi}}{\partial z_i} \frac{\partial z_i}{\partial W_{ij}}$$

Kronecker Product

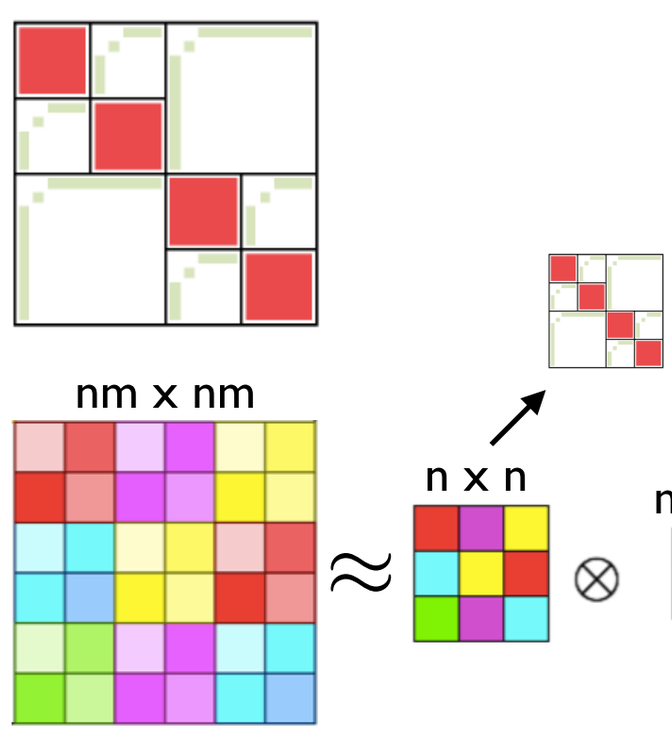
$$= \sum_{\mathbf{x}} \frac{\partial \mathcal{L}}{\partial W_{ij}} = \sum_{\mathbf{x}} \frac{\partial \mathcal{L}}{\partial z_i} \frac{z_i}{W_{ij}} = \sum_{\mathbf{x}} g_i a_j$$

Hierarchical Low-rank?

Fisher Matrix

100Mx100M

$$\frac{\partial J}{\partial \boldsymbol{\theta}}^T \frac{\partial J}{\partial \boldsymbol{\theta}}$$



Kronecker Factorization

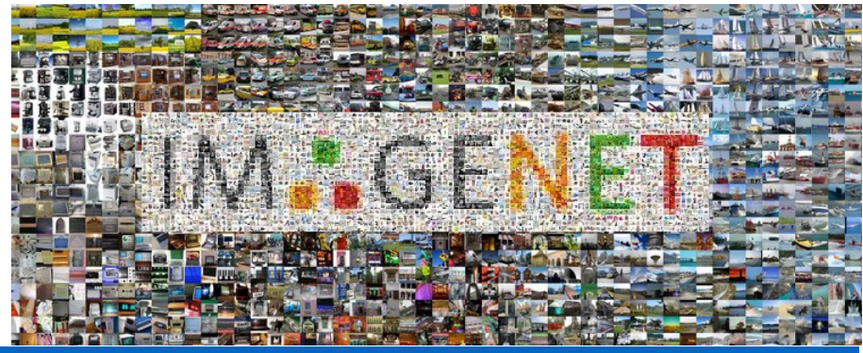
$$\left(\frac{\partial J}{\partial \boldsymbol{\theta}}^T \frac{\partial J}{\partial \boldsymbol{\theta}}\right)^{-1} = \left\{\left(\sum_{\mathbf{x}} \mathbf{g} \otimes \mathbf{a}\right)^T \left(\sum_{\mathbf{x}} \mathbf{g} \otimes \mathbf{a}\right)\right\}^{-1}$$
$$\approx \left(\sum_{\mathbf{x}} \mathbf{g}^T \mathbf{g}\right)^{-1} \otimes \left(\sum_{\mathbf{x}} \mathbf{a}^T \mathbf{a}\right)^{-1}$$
$$= G^{-1} \otimes A^{-1}$$

Large-scale Distributed Second-order Optimization Using Kronecker-factored Approximate Curvature for Deep Convolutional Neural Networks

Kazuki Osawa[1] Yohei Tsuji[1] Yuichiro Ueno[1] Akira Naruse[3] Rio Yokota[2] Satoshi Matsuoka[4,1]

[1]School of Computing, Tokyo Institute of Technology
[2]Global Scientific Information and Computing Center, Tokyo Institute of Technology
[3]NVIDIA
[4]RIKEN Center for Computational Science
{oosawa.k.ad, tsuji.y.ae, ueno.y.ai}@m.titech.ac.jp
anaruse@nvidia.com, rioyokota@gsic.titech.ac.jp, matsu@is.titech.ac.jp

https://arxiv.org/pdf/1811.12019.pdf



2018/12/2 Assert

Top 10 Arxiv Papers Today in Computer Science

2.06 Mikeys

#1. Second-order Optimization Method for Large Mini-batch: Training ResNet-50 on ImageNet in 35 Epochs

Kazuki Osawa, Yohei Tsuji, Yuichiro Ueno, Akira Naruse, Rio Yokota, Satoshi Matsuoka

Large-scale distributed training of deep neural networks suffer from the generalization gap caused by the increase in the effective mini-batch size. Previous approaches try to solve this problem by varying the learning rate and batch size over epochs and layers, or some ad hoc modification of the batch normalization. We propose an alternative approach using a second-order optimization method that shows similar generalization capability to first-order methods, but converges faster and can handle larger mini-batches. To test our method on a benchmark where highly optimized first-order methods are available as references, we train ResNet-50 on ImageNet. We converged to 75% Top-1 validation accuracy in 35 epochs for mini-batch sizes under 16,384, and achieved 75% even with a mini-batch size of 131,072, which took 100 epochs.

more | pdf | html

Figures

None.

	#GPU/TPU	time
Facebook	512	30 min
Preferred Networks	1024	15 min
UC Berkeley	2048	14 min
Tencent (7/30/2018)	2048	6.6 min
Sony (11/13/2018)	2048	3.7 min
Google (11/16/2018)	1024	2.2 min
Sony (3/6/2019)	3456	2 min
Fujitsu (4/1/2019)	2048	75 sec
This work	4096	? sec

Accepted to CVPR2019

ATMG2019

Special Session: Auto-Tuning for Multicore and GPU (ATMG) In conjunction with the IEEE MCSoC-19 at Nanyang Executive Centre, Nanyang Technological University, Singapore.

- Optimized Algorithms for Numerical Libraries
- Automatic Code Generation and Empirical Compilation Hybrid Programming for Threads and Processes
- Communication Optimization
- Mixed Precision and Accuracy Assurance for Numerical Computing
- Power Consumption Optimization
- Fault Tolerance

Important Dates

Abstract submission due:	May 20, 2019
Full paper submission due:	May 20, 2019
Author notification:	June 23, 2019
Camera ready manuscript:	July 21, 2019
MCSoC-19 Date:	October 1-4, 2019
Session Date:	TBD