jh180029-NAJ

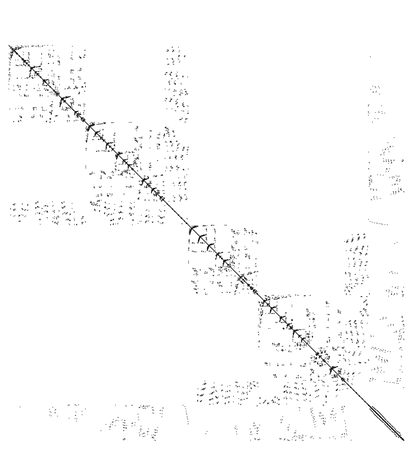
鈴木 厚 (大阪大学 サイバーメディアセンター)

Implementation of parallel sparse solver on CPU-GPU hybrid architecture

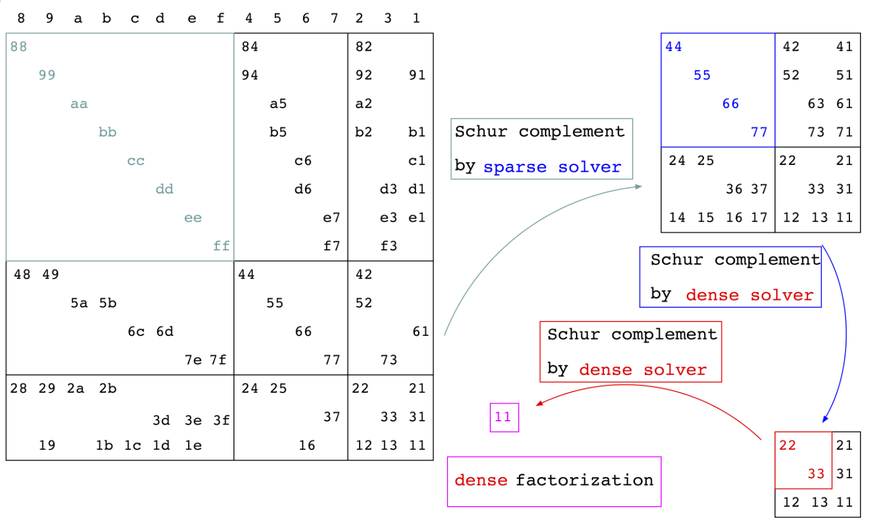
abstract

A sparse direct solver, Dissection performs LDU-factorization of large sparse matrix in parallel on shared memory architecture. With graph decomposition of the sparse matrix and block-wise procedure of dense sub-matrices, BLAS Level 3 operations, e.g. DGEMM and DTRSM are efficiently used. LDU-factorization can keep good accuracy by pivoting with postponing, even after introducing these strategies. This project aims to implement Dissection code on CPU-GPU hybrid architecture, by keeping LDU-factorization with pivoting on CPU and migrating BLAS L.3 operations to GPU.

nested-dissection ordering of sparse matrix



7 stencil of Poisson equation, 11^3 nodes
nested dissection ordering by SCOTCH



multi-frontal LDU-factorization by recursive computation of Schur complements

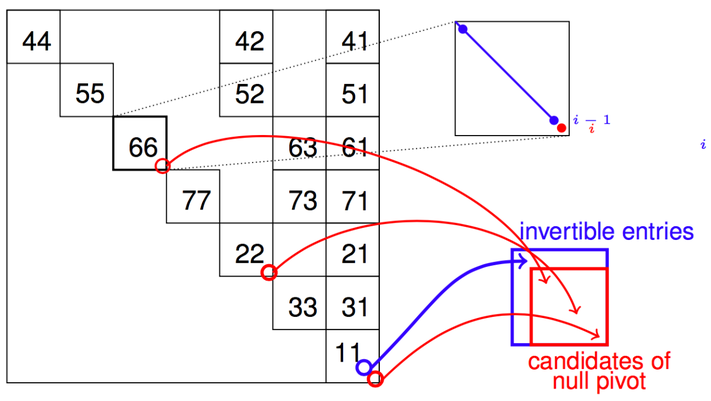
$$\begin{bmatrix} A_{11} & A_{21} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & S_{22} \end{bmatrix} \begin{bmatrix} I_1 & A_{11}^{-1}A_{12} \\ 0 & I_2 \end{bmatrix}$$

$$\begin{aligned} S_{22} &= A_{22} - A_{21}A_{11}^{-1}A_{12} \\ &= A_{22} - A_{21}(L_{11}D_{11}U_{11})^{-1}A_{12} \\ &= A_{22} - (U_{11}^{-T}A_{21}^{T})^{T}(D_{11}^{-1}L_{11}^{-1}A_{12}) \end{aligned}$$

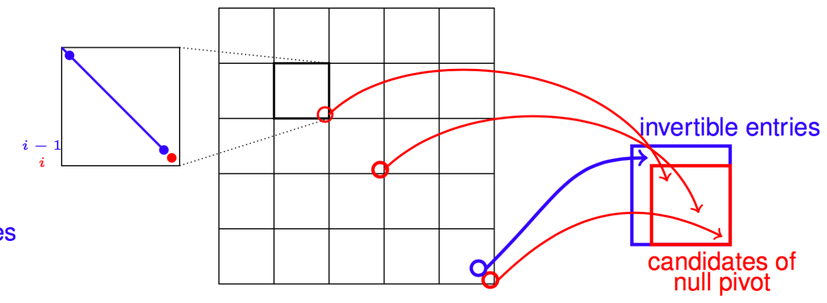
block LDU-factorization by forward substitution of multiple-RHS and updating Schur complement

symmetric pivoting with postponing for multi-frontal and block factorization

τ : given threshold for null pivot $\simeq 10^{-2}$
$|A(i,i)|/|A(i-1,i-1)| < \tau \ \Rightarrow |A(i,i)|$ is null pivot.

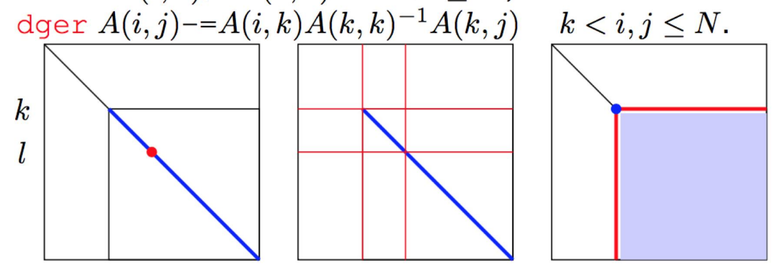


postponed pivots in multi-frontal structure



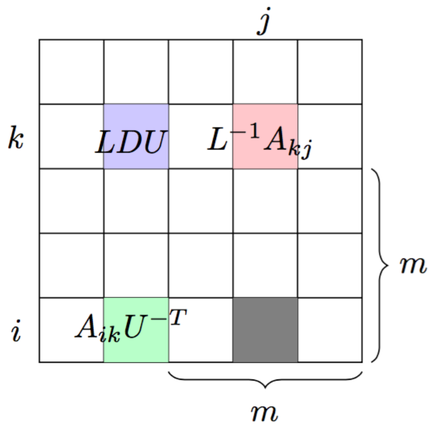
postponed pivots in block dense structure

do $k = 1, \cdots, N$
find $k < l \leq n$ max$|A(l,l)|$,
exchange rows and columns : $A(k,*) \leftrightarrow A(l,*)$, $A(*,k) \leftrightarrow A(*,l)$.
dscal $A(k,j)/=A(k,k) \quad k < j \leq N$,
dscal $A(i,k)/=A(k,k) \quad k < i \leq N$,
dger $A(i,j)-=A(i,k)A(k,k)^{-1}A(k,j) \quad k < i,j \leq N$.



LDU-factorization with symmetric pivoting

strategy for GPU implementation with block factorization



A_{kk} : k-th diagonal,
$b \times b$-squared

- LDU-factorization $A_{kk} = L_{kk}D_{kk}U_{kk}$
- solve multiple-RHS $L_{kk}X_j = A_{kj}, \quad 1 \leq j \leq m$
- solve multiple-RHS $U_{kk}^T Y_i^T = A_{ik}^T, \quad 1 \leq i \leq m$
- update Schur complement $S_{ij} = A_{ij} - Y_i^T(D_{kk}^{-1}X_j)$

$$\frac{\text{operation}}{\text{memory access}} = (m + m^2)b \geq 3,072$$

$$(b = 512, m \geq 2)$$

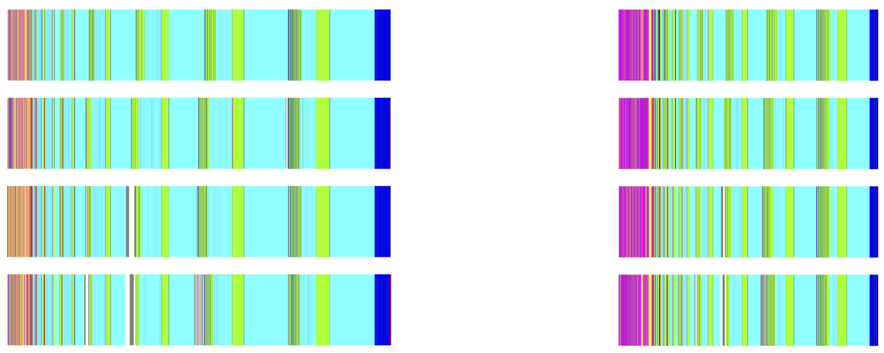
$$\frac{\text{Flop/s}}{\text{word/s}} = \frac{4.6\text{T}}{2\text{G}} = 2,300$$

- data transfer GPU $\Leftrightarrow$ CPU : $2 \times b^2$
- LDU with pivoting on CPU : $\gamma \times b^3$
- DTRSM on GPU : $2 \times m \times b^3$
- DGEMM on GPU : $2 \times m^2 \times b^3$

nVIDIA Pascal : 32FMA @ 1.3GHz $\times 56 = 4.6$TFlop/s
GPU $\Leftrightarrow$ CPU, PCI-express 16GB/s = 2Gword/s

Major two parts of block LDU-factorization, DTRSM and DGEMM are migrated to GPU. Storing factorized matrix in GPU and data transfer of diagonal LDU block will achieve good ratio of arithmetic and data movement, which is comparable to hardware value in CPU-GPU hybrid architecture with rather slow PCI-express connection.

parallel efficiency of Dissection solver



asynchronous parallel task execution on Intel Xeon v2(left) and NEC SX-ACE (right)

target system

GPU node of Octopus, Cybermedia Center, Osaka University
nVIDIA Pascal P100 GPU x4 +
Intel Xeon Gold 6126 x2

member of the project

Atsushi Suzuki : Cybermedia Center, Osaka University
Daisuke Furihata : Cybermedia Center, Osaka University
François-Xavier Roux: ONERA / Laboratoire Jacques-Louis Lions, Sorbonne Universite