LBL Updates

November, 2021

LBL Updates

- Continued work on multi-GPU U-Solve
  - works with NVSHMEM on Summit (can run on Traverse, refers to example_scripts/run_cmake_build_summit_nvshmem_gpu.sh for compiling)
  - available from the **nvshmem_multiGPUsolve branch**
- Perlmutter(NVIDIA) and Spock(AMD)
  - NVSHMEM deadlock issue on Perlmutter traced to CUDA runtime behavior
  - ROCSHMEM still has runtime issue on Spock
- Continued updates to 3D Solve / interface
- Note, Nan will be on parental leave thru January

New U-Solve data structure on GPU

- Background: SuperLU is designed for sparse LU with non-symmetric nonzero patterns in L and U factors. We use supernode partition for the L-factor, to get supernode size (dense submatrix) as large as possible. In general, the U-factor does not have the same nonzero pattern as the L-factor, so the U data structure uses a skylined representation that is compatible with the L supernodes partition.
  - The big advantage is that we do not store any extra zeros
  - The big disadvantage is that it is not friendly for fine-grained parallelism ⇒ U-solve performance is poor on GPU

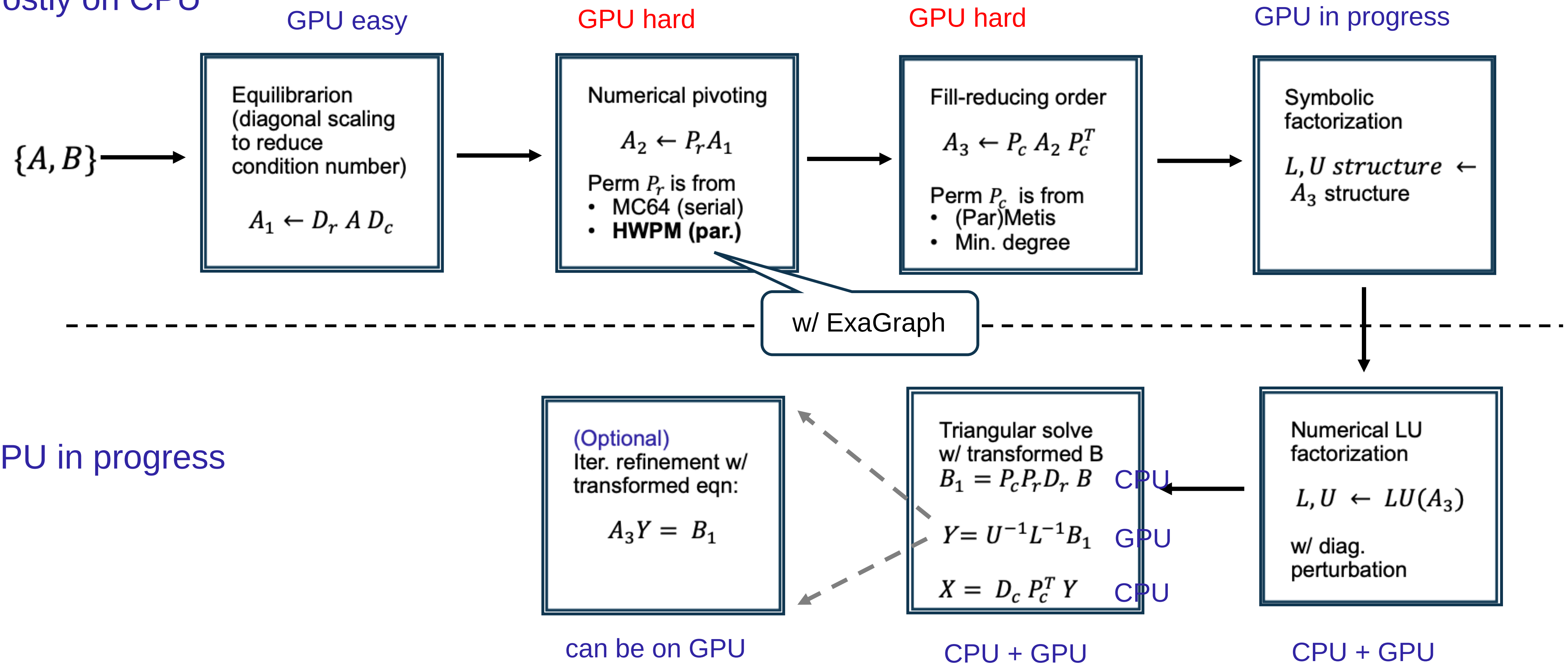
- New design to mitigate L- and U-solve performance discrepancies:
  - on GPU, use the same supernode data structure for both L and U, which requires padding zeros in the U data structure ⇒ trade off memory for speed
  - On CPU, use the existing data structures.

- Code status: factorization is mostly working
- Next step: implement the new U-solve on GPU with this structure

SuperLU_DIST algorithms flowchart

Nonsymmetric pattern

Preprocessing
mostly on CPU



GPU easy

Equilibrarion (diagonal scaling to reduce condition number)

$A_1 \leftarrow D_r\, A\, D_c$

GPU hard

Numerical pivoting

$A_2 \leftarrow P_r A_1$

Perm P_r is from

- MC64 (serial)
- **HWPM (par.)**

w/ ExaGraph

GPU hard

Fill-reducing order

$A_3 \leftarrow P_c\, A_2\, P_c^T$

Perm P_c is from

- (Par)Metis
- Min. degree

GPU in progress

Symbolic factorization

L, U *structure* ← A_3 structure

GPU in progress

Numerical LU factorization

$L, U \leftarrow LU(A_3)$

w/ diag. perturbation

CPU + GPU

Triangular solve w/ transformed B

$B_1 = P_c P_r D_r\, B$ CPU

$Y = U^{-1} L^{-1} B_1$ GPU

$X = D_c\, P_c^T\, Y$ CPU

CPU + GPU

(Optional)
Iter. refinement w/ transformed eqn:

$A_3 Y = B_1$

can be on GPU

Solve the transformed system: $(P_c\, P_r\, D_r\, A\, D_c\, P_c^T)(P_c\, D_c^{-1}\, X) = P_c\, P_r\, D_r\, B$

Q&A