LBL Updates

March 2023

Topics

- One-sided Solvers on Crusher/Spock/Frontier GPUs (Nan)
- 3D One-sided Solvers (Yang)
- Batch 1D Toroidal Solves (Hans)
- Memory Usage and Savings (Sherry)
- Randomized/Block methods (Sam)
- Q&A

3D Triangular Solve

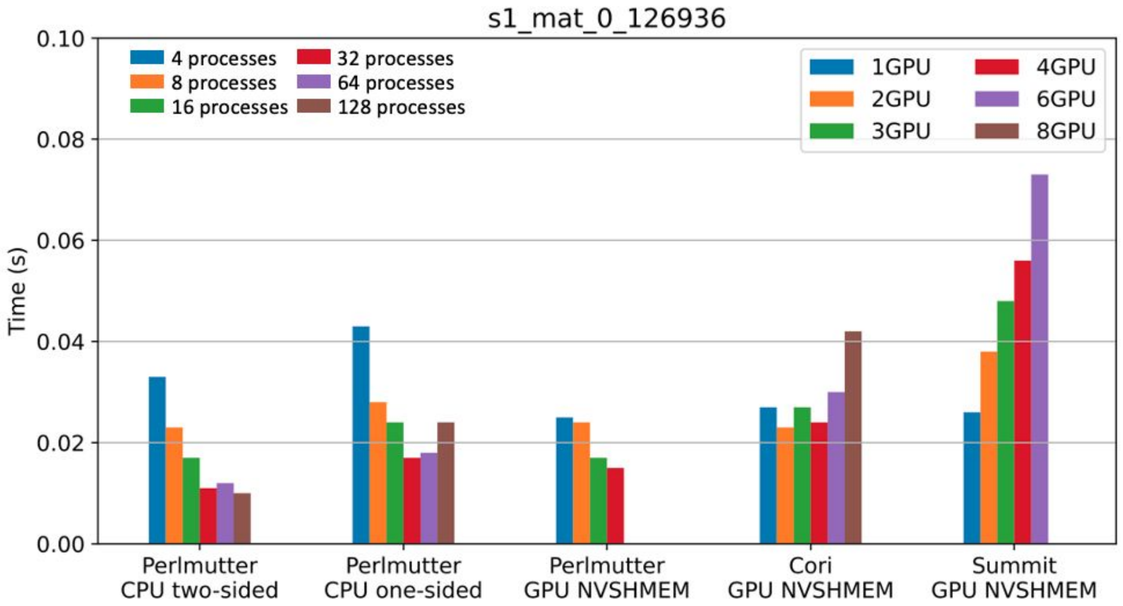
- GPU implementation done
  - 3D decomposition
  - each 2D plane is a NVSHMEM triangular solve (multi GPU)
- currently focused on analyzing performance

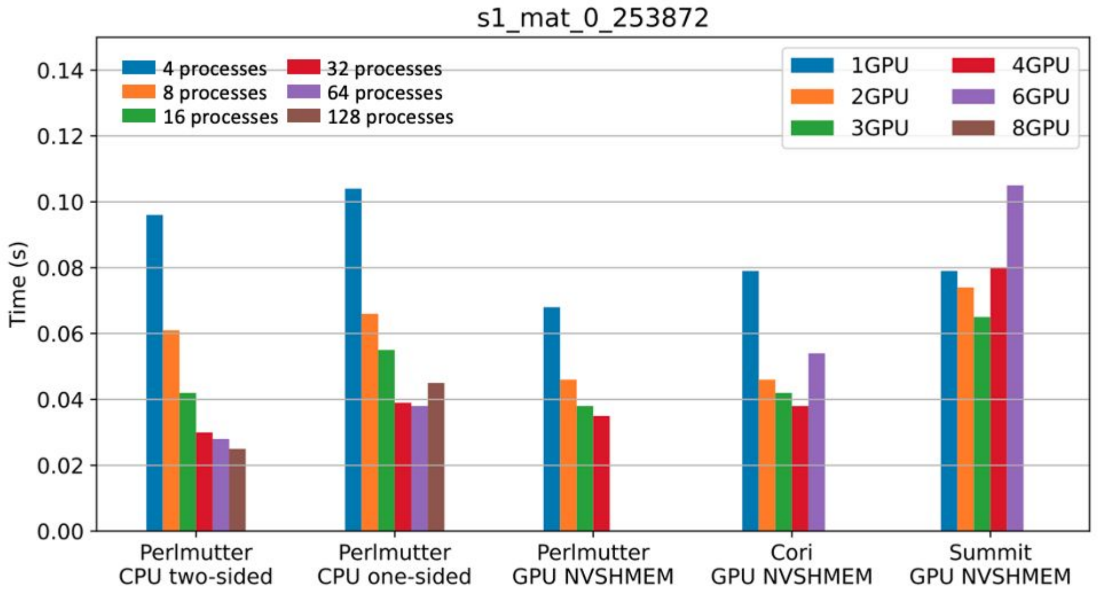
One-sided Solvers on Crusher/Spock/Frontier GPUs

- ROC_SHMEM for direct GPU-GPU communication on AMD GPUs
  - reverse offload (RO): offload GPU communications to standard MPI one-sided routines on CPUs
  - GIO: bypass CPUs, comparable to NVSHMEM
  - both are transparent to users, program like NVSHMEM
- Crusher and Spock are testbed for Frontier
  - perform RO on Crusher and Frontier (available in next release, April)
    - lacks Infiniband hardware required by GIO
  - perform GIO on Spock (waiting for ROC_SHMEM team)
  - lack support on MPI sub_communicators (pushing ROC_SHMEM team to support it)

One-sided Solvers on Perlmutter/Summit/Cori GPUs

- Explored process grids for CPUs and GPUs
  - Px1 process grid send data per thread block
  - 1xP process grid sends data per thread and stresses network more than Px1
  - Px1 is generally faster and more scalable.
- Evaluated performance on CPUs (both traditional 2-sided and 1-sided) and GPUs (all 1-sided)
  - 2-sided CPU scales well to 32 cores (processes) and then saturates… highlights ultimate limits of concurrency on performance
  - stock 1-sided MPI performance on Perlmutter is now just slightly slower than 2-sided (no foMPI)
  - 1 GPU is < 2x faster than 1 CPU core
  - 128 cores is usually faster than 4 GPUs
  - Perlmutter GPU (A100) scales better than Cori GPU (V100) or Summit (P9+V100) due to superior inter-GPU network performance
- NVSHMEM Messaging microbenchmark:
  - Perlmutter GPU = 1.8x Cori GPU
  - Perlmutter GPU = 17x Summit GPU





Toroidal direction C1 solver

- Got Steve's F90 C1 code working on Cori
  - Block-tridiagonal algorithm, targeting ~128 x 2 DoF's in toroidal direction
  - <u>Next step</u>: parallel decomp in each poloidal plane
  - Each "line solve" needs 2 DoF's from each toroidal coor
- Continued progress and moving towards integration in M3D
  - needed POC to get M3D running on Cori/Perlmutter
- Next step: get FGMRES in PETSc working

Randomized and Block Methods as preconditioners

- started a series of discussions with
  - Riley Murray (LBL), Michael Mahoney (LBL), Burlen Loring (LBL),
  - Laura Grigori (INRIA), Edouard Timsit (INRIA)
- Three sub approaches…
  - randomized projection + pseudoinverse (textbook)
  - block projection into randomized higher dimensional space (Riley)
  - Randomized Orthogonal Projection Methods (ROPM) over Krylov subspace (Edouard)
- Goals:
  - lower memory & compute complexity preconditioning
  - inherently parallel operations (amenable to GPUs/MPPs)
  - make it easier for preconditioner to capture all coupling
- Only started ~2 weeks ago; progress delayed due to availability/SIAM/etc…

Feedback for LBL

- C1 Hermite MG (interpolation)

- Ask Kokkos/RAJA/Cameron team on 1-sided communication

- Test GPU preconditioner in PETSc (example or M3D)
  - are good GPU SpTRSV speedups indicative of good GMRES speedups?
  - M3D w/1 plane as direct solver
  - M3D w/multiple planes and block preconditioner
  - FGMRES configuration
- Push Riley to consider preconditioned block method SpTRSM