CENTER FOR TOKAMAK TRANSIENTS SIMULATION

FES-ASCR Collaboration: Linear Solvers

Samuel Williams, Nan Ding, Yang Liu, Sherry Li
Lawrence Berkeley National Lab

FES Project Review

Virtual

October, 2021

FES-ASCR Collaboration: Linear Solvers

CHALLENGES

- M3D and NIMROD require solving large, ill-conditioned systems
- Both use SuperLU-preconditioned GMRES
  1. Factor a block Jacobi preconditioner
  2. Apply the preconditioner (L&U solves) on every GMRES iteration
  3. System can change every time step
- Bottlenecks…
  - Factorization time is expensive
    <<1 factorization per solve
  - Preconditioner (SpTRSV) dominates solve time
    Two SpTRSV per iteration; many iterations

MULTI-FACETED APPROACH

- Ensuring M3D/NIMROD teams can leverage GPU supercomputers
- Parameter Tuning in SuperLU
- Preconditioner Performance
  - GPU-accelerated L&U solves
  - GPU-accelerated Factorization
- Factorization and Preconditioner Scalability

Close Collaborations with PPPL and TechX Teams

- LBNL dedicated ~20% effort to helping teams without expectation of publications

- Monthly Telecons with each team
  - Ensure ASCR activities are coupled with FES needs
  - Provide guidance on GPU programming and future DOE machines

- M3D-C1 joint code work: PPPL-LBNL-PETSc
  - Compiling GPU-accelerated PETSc with SuperLU presented a number of issues (number of zoom calls to resolve)
  - LBNL-PPPL resolved installation/performance issues on NERSC's Cori Haswell and KNL machines
  - LBNL-PPPL worked together to install M3D-C1 with GPU use on "Traverse" IBM/NVIDIA machine at PPPL
  - Parameter tuning using GPTune

- NIMROD joint code work: TechX-LBNL
  - Separate code review/debugging sessions to incorporate new features of SuperLU
  - Upgraded NIMROD interface to use SuperLU's new Fortran90 interface
  - Adapted SuperLU 3D algorithm interface (for reducing communication) to NIMROD matrix layout
  - Parameter tuning using GPTune

Parameter Tuning for M3D-C1 and NIMROD

- SuperLU/NIMROD/M3D-C1 have a number of parameters that control mathematical properties, parallelism, matrix assembly block size, etc…
  - Some parameters can be set cognizant of how SuperLU will be used throughout an application
  - Others are machine and problem-dependent

- Exploiting diagonal dominance and reusing sparsity ordering **improved M3D-C1 factorization time by 3x and solver performance by 15%**

- LBNL developed a Gaussian Process Bayesian optimization framework (**GPTune**) to help users efficiently explore the combinatoric parameter space
  - ***Improves overall NIMROD performance by 10%***
  - ***Improves overall M3D-C1 performance by 17%***

Y. Liu, W. Sid-Lakhdar, O. Marques, X. Zhu, J.W. Demmel, X.S. Li, “GPTune: Multitask Learning for Autotuning Exascale Applications’’, Proc. of Principles and Practice of Parallel Programming (PPoPP), 2021.
Xinran Zhu, Yang Liu, Pieter Ghysels, David Bindel, Xiaoye S. Li, "GPTuneBand: Multi-task and Multi-fidelity Autotuning for Large-scale High Performance Computing Applications", in submission to NeurIPS, 2021.

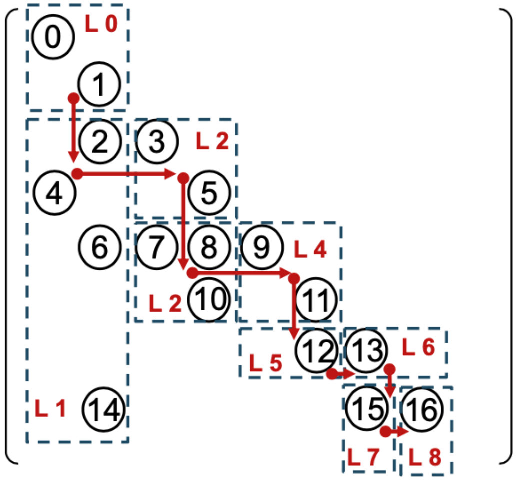
Algorithms & Software Advances from Applied Math Side

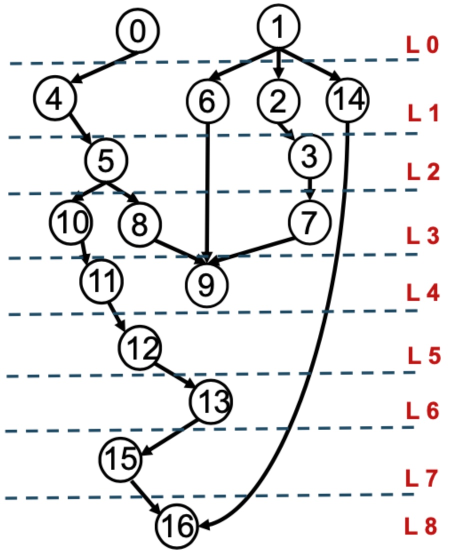
Preconditioner Performance

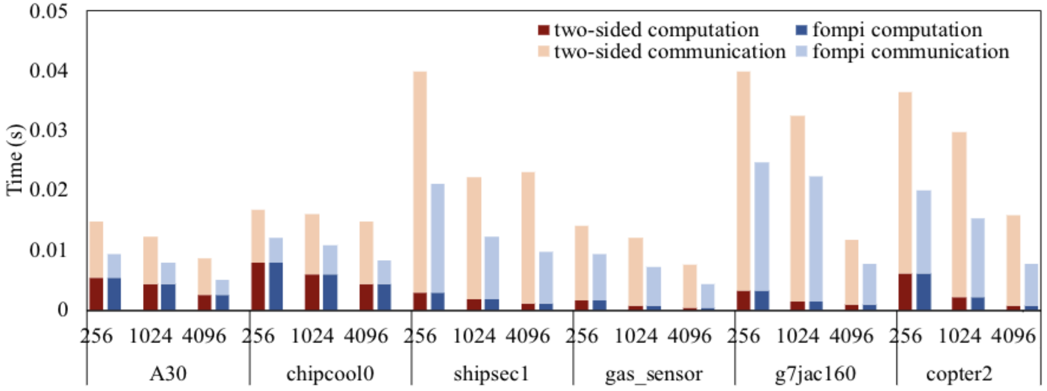
- Preconditioner time is dominated by two repeated SpTRSVs (L-solve and U-Solve w/ 1 RHS)
- Each can be viewed as walking a DAG…
  - Nodes are small (<< 128x128) GEMVs/TRSVs
  - Edges are small (<< 1KB) MPI messages
- Performance is highly dependent on…
  - MPI Overhead (messaging rate)
  - DAG Critical Path

➢ ***One-sided communication (foMPI) can improve SpTRSV by 2.2x on Cori KNL***

➢ ***Performance model highlights nuances***



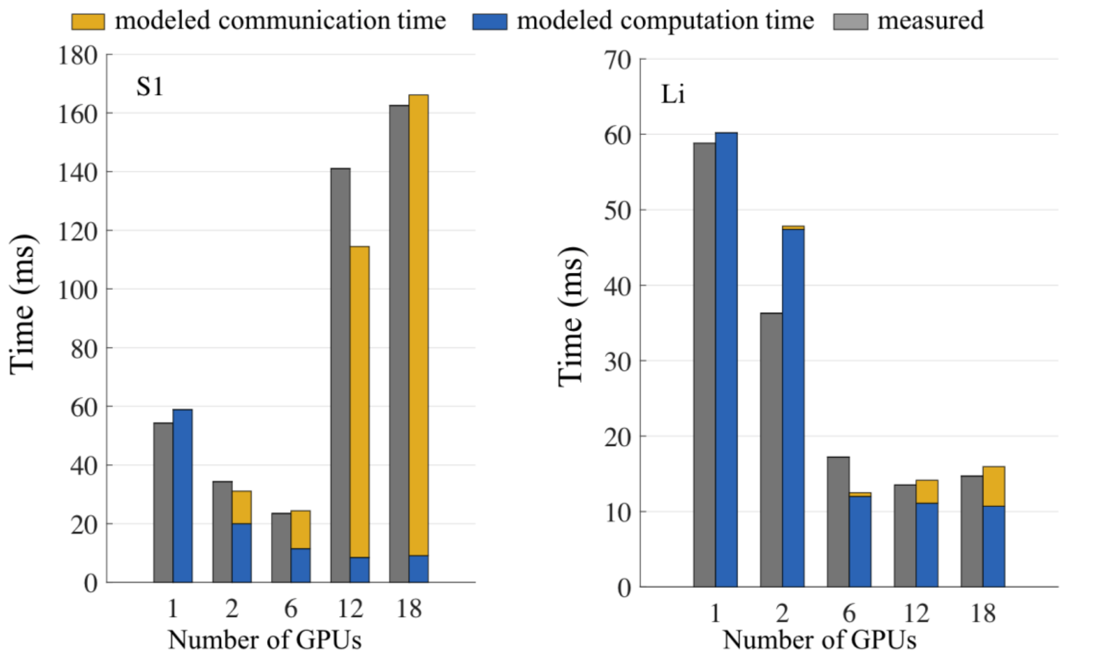




Nan Ding, Samuel Williams, Yang Liu, Xiaoye S. Li, "Leveraging One-Sided Communication for Sparse Triangular Solvers", SIAM Conference on Parallel Processing for Scientific Computing, 2020.

GPU-Accelerated Preconditioners

- LBL has created single-GPU SpTRSV solvers for NVIDIA (CUDA) and AMD (HIP) GPUs
  - Works best if a poloidal plane can fit on one GPU
- Extended with one-sided GPU libraries (NVSHMEM, ROCSHMEM*)
  - ***Enables scalable, distributed memory, GPU-accelerated solvers***
  - Performance and scalability are highly dependent on matrix sparsity and inter-node communication performance
- Modeled alternate process mappings for GPUs
  - ***Potential for 2x speedup over default 1D block cyclic mapping using 6 GPUs***



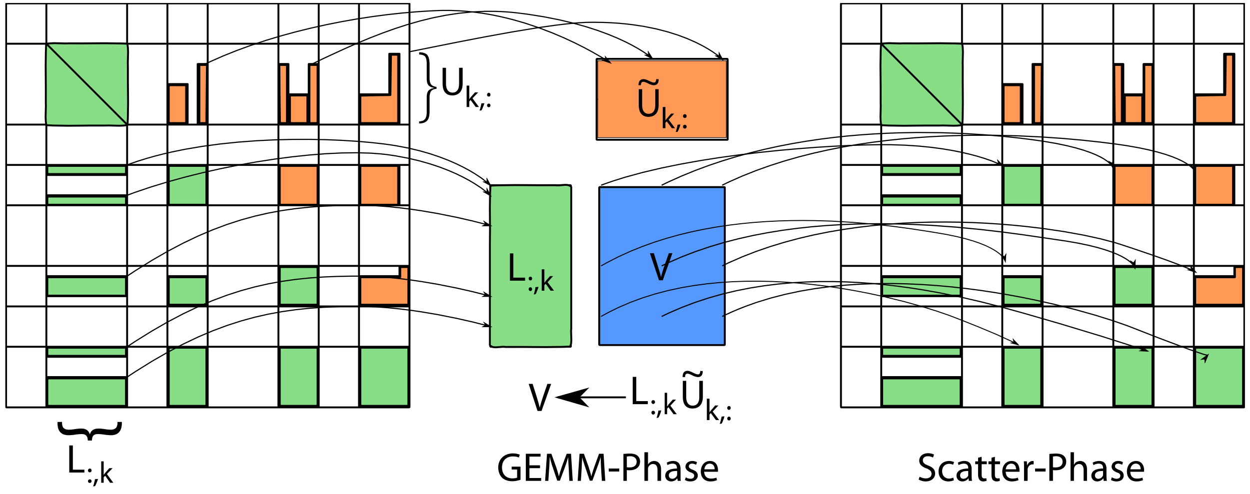
AMD performance evaluation delayed due to waiting for AMD software updates

Nan Ding, Yang Liu, Samuel Williams, Xiaoye S. Li, "A Message-Driven, Multi-GPU Parallel Sparse Triangular Solver", SIAM Conference on Applied and Computational Discrete Algorithms (ACDA21), 2021.

GPU-Accelerated Factorization

- SuperLU performs sparse LU factorization
- For efficiency, at run time, SuperLU identify dense blocks for Schur complement updates (allows DGEMM calls)
- DGEMMs can be executed on the CPU or offloaded to the GPU
- Subsequent work offloaded gather/scatter operations to the GPU as well

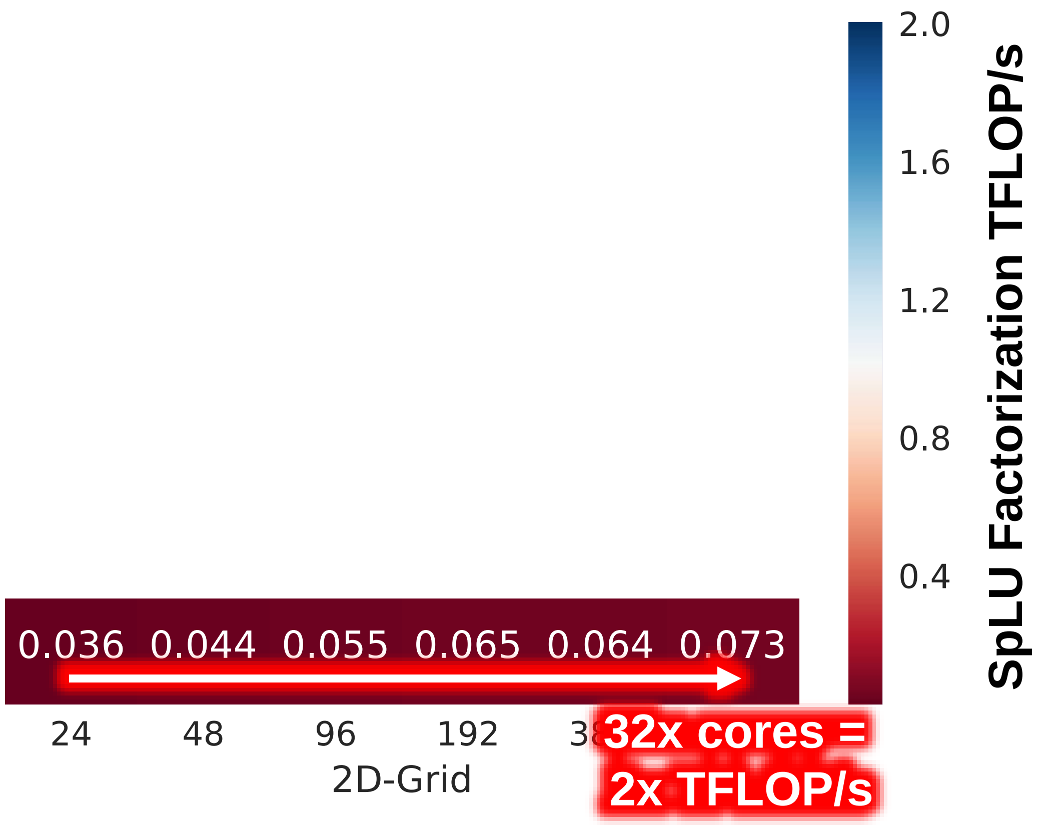
➢ ***Using GPUs can accelerate factorization by 3x***



P. Sao, R. Vuduc, and X.S. Li, "A distributed CPU-GPU sparse direct solver", Proc. of Euro-Par Parallel Processing, 2014

Factorization and Preconditioner Scalability

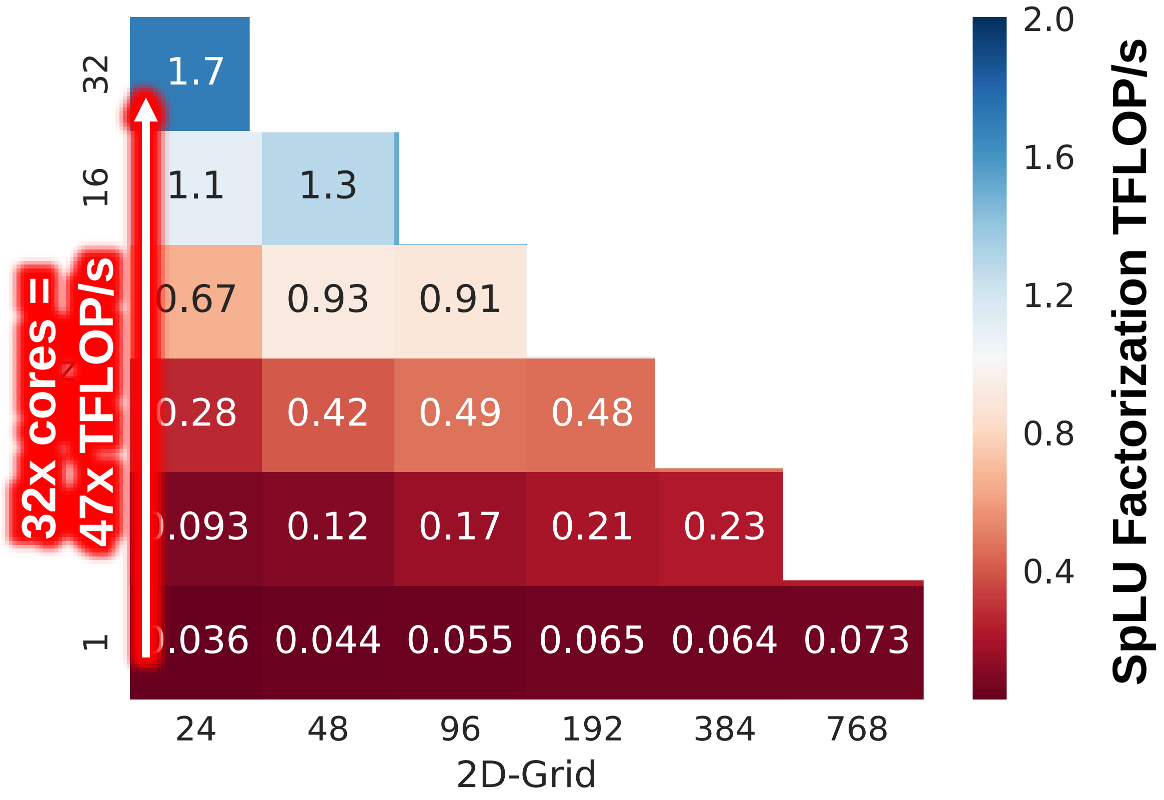
- SpLU and SpTRSV use 2D block cyclic process decompositions
  - Hard to attain perfect scaling
- LBNL explored 3D approaches to factorization and solve to reduce communication
  - Selective copies of Schur complement updates along 3rd dimension of process grid
  - Reduce number of messages by log(n)
  - Reduce message volume by sqrt(log(n))
  - Less than 2x increase in memory usage



P. Sao, R. Vuduc, X. Li, "A communication-avoiding 3D algorithm for sparse LU factorization on heterogeneous systems", JPDC, 2019
P. Sao, R. Kannan, X.S. Li, R. Vuduc, "A communication-avoiding 3D sparse triangular solver", ICS, 2019

Factorization and Preconditioner Scalability

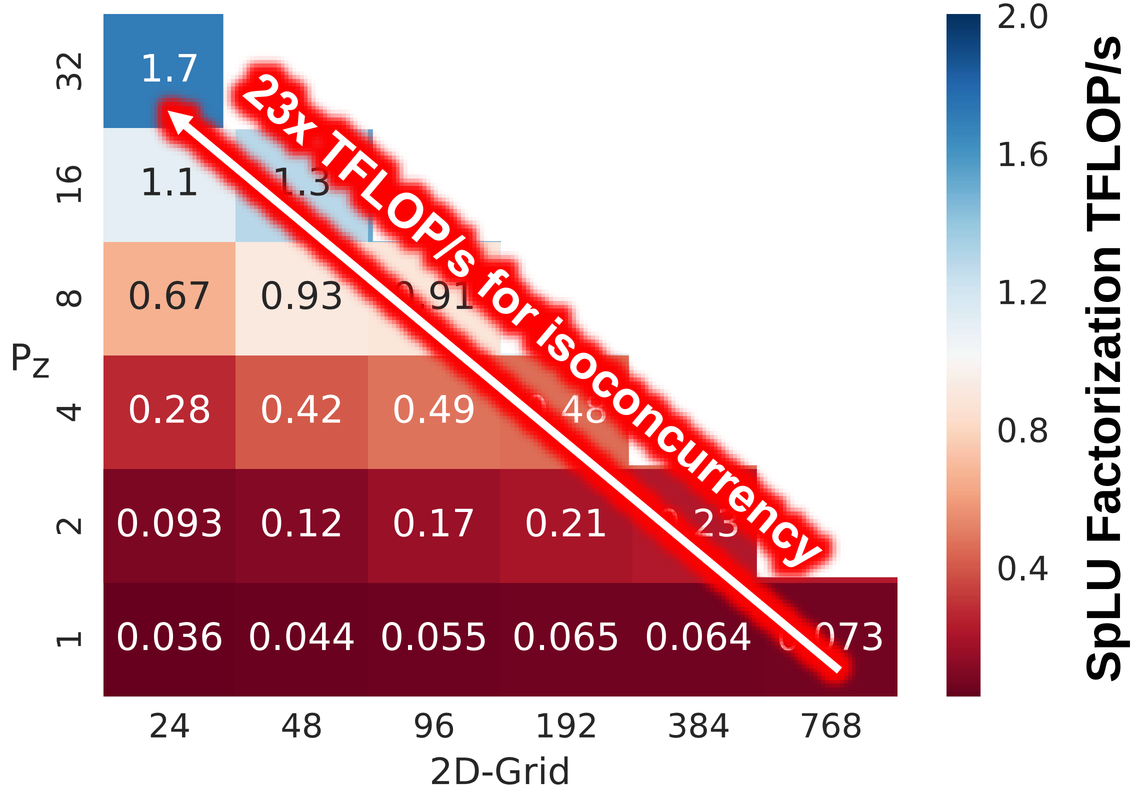
- SpLU and SpTRSV use 2D block cyclic process decompositions
  - Hard to attain perfect scaling
- LBNL explored 3D approaches to factorization and solve to reduce communication
  - Selective copies of Schur complement updates along 3rd dimension of process grid
  - Reduce number of messages by log(n)
  - Reduce message volume by sqrt(log(n))
  - Less than 2x increase in memory usage



P. Sao, R. Vuduc, X. Li, "A communication-avoiding 3D algorithm for sparse LU factorization on heterogeneous systems'', JPDC, 2019
P. Sao, R. Kannan, X.S. Li, R. Vuduc, "A communication-avoiding 3D sparse triangular solver'', ICS, 2019

Factorization and Preconditioner Scalability

- SpLU and SpTRSV use 2D block cyclic process decompositions
  - Hard to attain perfect scaling
- LBNL explored 3D approaches to factorization and solve to reduce communication
  - Selective copies of Schur complement updates along 3rd dimension of process grid
  - Reduce number of messages by log(n)
  - Reduce message volume by sqrt(log(n))
  - Less than 2x increase in memory usage

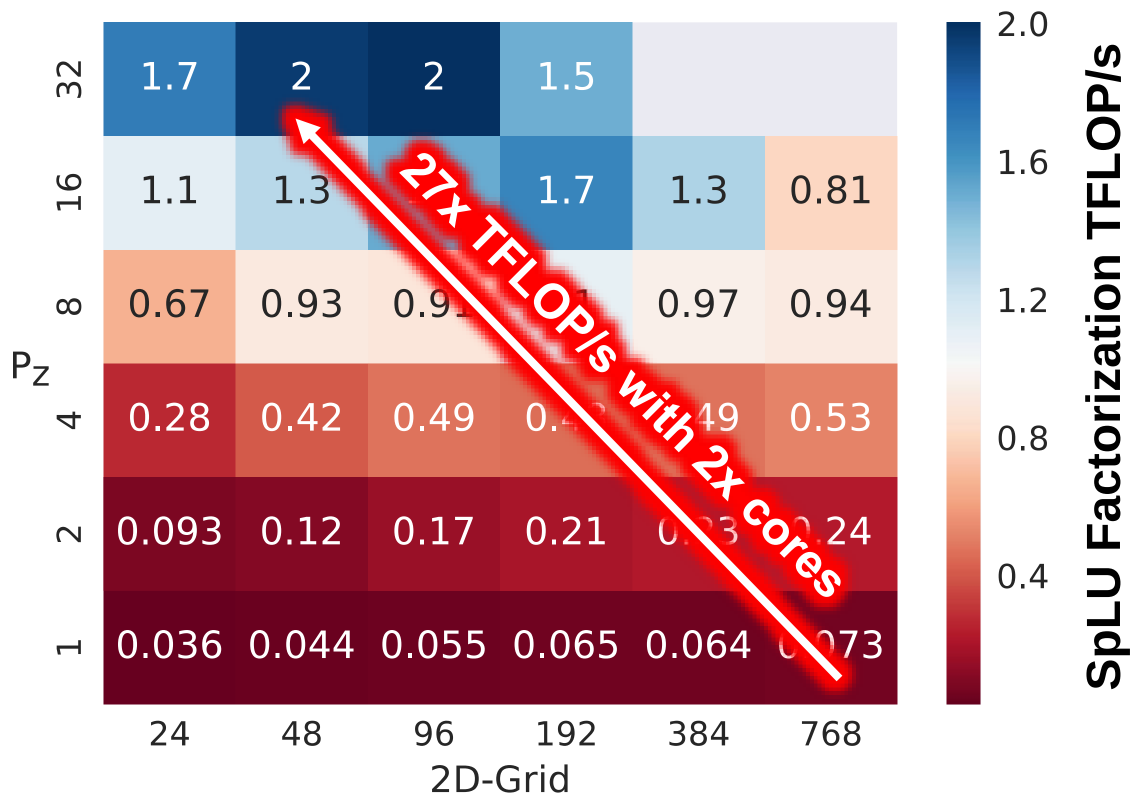


P. Sao, R. Vuduc, X. Li, “A communication-avoiding 3D algorithm for sparse LU factorization on heterogeneous systems'', JPDC, 2019
P. Sao, R. Kannan, X.S. Li, R. Vuduc, “A communication-avoiding 3D sparse triangular solver'', ICS, 2019

Factorization and Preconditioner Scalability

- SpLU and SpTRSV use 2D block cyclic process decompositions
  - Hard to attain perfect scaling
- LBNL explored 3D approaches to factorization and solve to reduce communication
  - Selective copies of Schur complement updates along 3rd dimension of process grid
  - Reduce number of messages by log(n)
  - Reduce message volume by sqrt(log(n))
  - Less than 2x increase in memory usage

➢ ***For larger problems, this can improve performance/scalability by 3.5x to 27x***

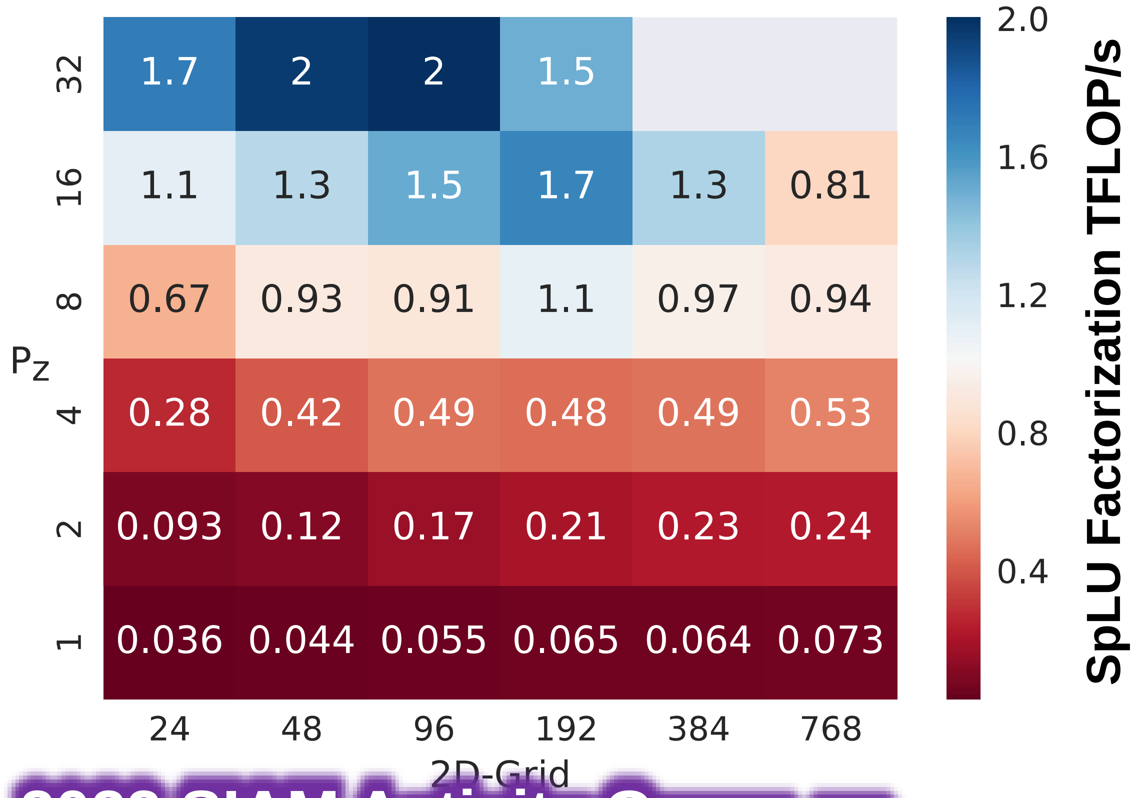


P. Sao, R. Vuduc, X. Li, "A communication-avoiding 3D algorithm for sparse LU factorization on heterogeneous systems'', JPDC, 2019
P. Sao, R. Kannan, X.S. Li, R. Vuduc, "A communication-avoiding 3D sparse triangular solver", ICS, 2019

Factorization and Preconditioner Scalability

- SpLU and SpTRSV use 2D block cyclic process decompositions
  - Hard to attain perfect scaling
- LBNL explored 3D approaches to factorization and solve to reduce communication
  - Selective copies of Schur complement updates along 3rd dimension of process grid
  - Reduce number of messages by log(n)
  - Reduce message volume by sqrt(log(n))
  - Less than 2x increase in memory usage

➢ ***For larger problems, this can improve performance/scalability by 3.5x to 27x***



2022 SIAM Activity Group on Supercomputing Best Paper

P. Sao, R. Vuduc, X. Li, “A communication-avoiding 3D algorithm for sparse LU factorization on heterogeneous systems’’, JPDC, 2019
P. Sao, R. Kannan, X.S. Li, R. Vuduc, “A communication-avoiding 3D sparse triangular solver", ICS, 2019

Year 5 Plans and Future Directions

Year 5 Plans

- Optimization of GPU-accelerated preconditioners on Perlmutter (A100/NVSHMEM) and Frontier (MI200/ROCSHMEM)
- Implementation of new U-solve in SuperLU data structure that trades increased memory usage for parallelism-friendly memory access

Future Directions

- Explore use of STRUMPACK (low rank approximation) as an alternative to direct factorization
- Development of GPU-friendly sparse linear solvers for ill-conditioned problems

Questions?