Towards scalable preconditioners

Sherry Li

Lawrence Berkeley National Laboratory

LQCD SciDAC-5 kickoff, December 1-2, 2022

Proposal subteam

- Eloy Romero
- Sherry Li
- Andreas Stathopoulos

- Development & deployment in SuperLU framework
  - Existing capabilities
  - New features for LQCD problems

Existing Capabilities

- Multi-node GPU support for NVIDIA, AMD and Intel GPUs
- Communication-avoiding 3D sparse LU factorization and sparse triangular solves
- Use of one-sided MPI and NVSHMEM / ROCSHMEM to mitigate communication cost in sparse triangular solves
- Mixed-precision algorithms
- Batch of multiple linear solvers on GPUs
- Integrated in multiple upstream math libraries as their building blocks
  - Hypre, PETSc, SUNDIALS, Trilinos, …
- Softtware dependencies
  - MPI, OpenMP, CUDA / HIP / SYCL, NVSHMEM / ROCSHMEM
  - BLAS, LAPACK, (Par)METIS

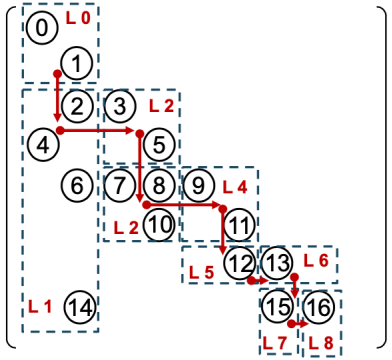
Sparse triangular solve: reduce communication

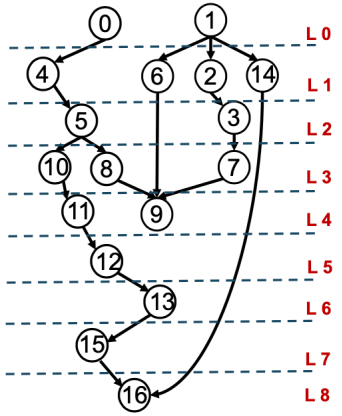
SuperLU

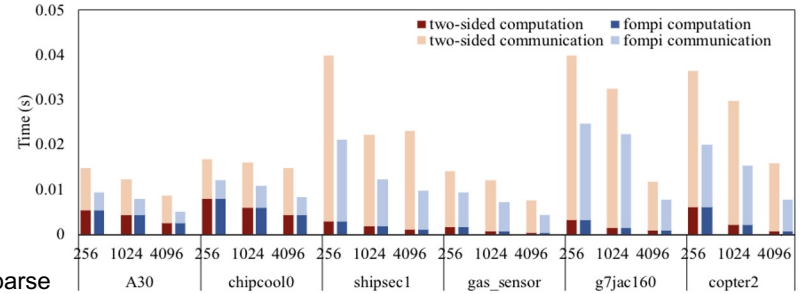
- Preconditioner time is dominated by two repeated SpTRSVs (L-solve and U-Solve w/ 1 RHS)
- Each can be viewed as walking a DAG …
  - Nodes are small (<< 128x128) GEMVs/TRSVs
  - Edges are small (<< 1KB) MPI messages
- Performance is highly dependent on…
  - MPI Overhead (messaging rate)
  - DAG Critical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

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

➢ ***Performance model highlights nuances***







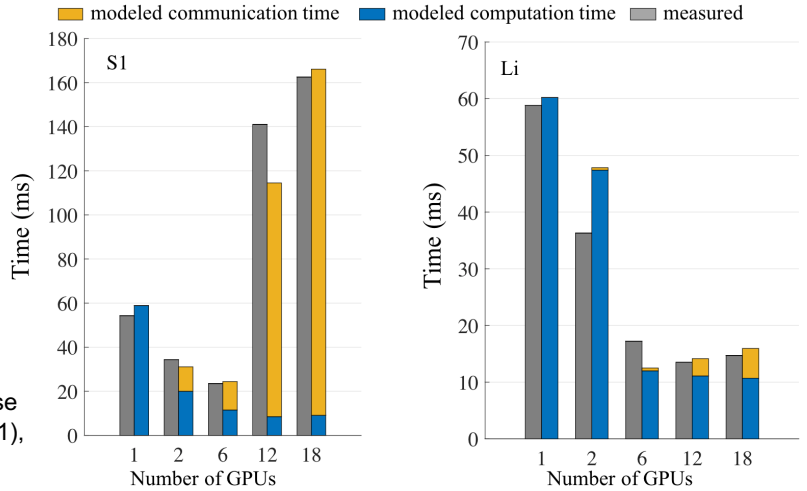
Nan Ding, Samuel Williams, Yang Liu, Xiaoye S. Li, "Leveraging One-Sided Communication for Sparse Triangular Solvers", SIAM Conference on Parallel Processing for Scientific Computing, 2020.

Sparse triangular solve: multi-GPUs

SuperLU

- Created a single-GPU SpTRSV solvers for NVIDIA (CUDA) and AMD (HIP) GPUs
  - Works best if entire L & U can fit on one GPU
- Extended with one-sided GPU libraries (NVSHMEM, ROCSHMEM*)
  - ***Enables scalable, distributed memory, GPU-accelerated solvers***
  - ***With 18 GPUs, up to 6x speedup over Nvidia cusparse_csrsv2()***
  - Performance and scalability are highly dependent on matrix sparsity and inter-node communication performance
- Modeled alternative process mappings for GPUs
  - ***Potential 2x speedup over default 1D block cyclic mapping using 6 GPUs***

**AMD evaluation delayed due to waiting for AMD software updates*



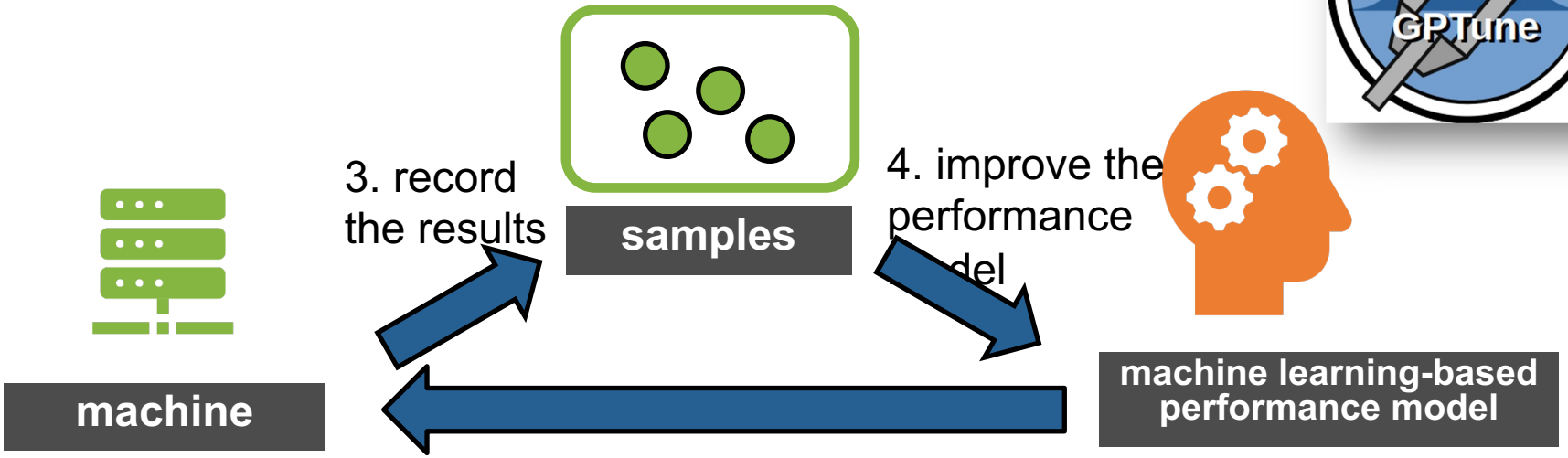
Nan Ding, Yang Liu, Samuel Williams, Xiaoye S. Li, "A Message-Driven, Multi-GPU Parallel Sparse Triangular Solver", SIAM Conference on Applied and Computational Discrete Algorithms (ACDA21), 2021.

HMC needs

- Time to compute the approximate L and U factors is critical because the preconditioners are not reused much
- Explore techniques to construct factors iteratively or asynchronously for better strong scaling
  - E.g.: Use fixed-point iteration to compute each entry of L and U (Chow)
  - Works particularly well for ILU(0) preconditioners because of their predetermined sparsity pattern
- Chow et al. (2015) demonstrated its effectiveness on a single GPU, but there is no large-scale multi-GPU or distributed-memory implementation

“Automatic” performance tuning with GPTune

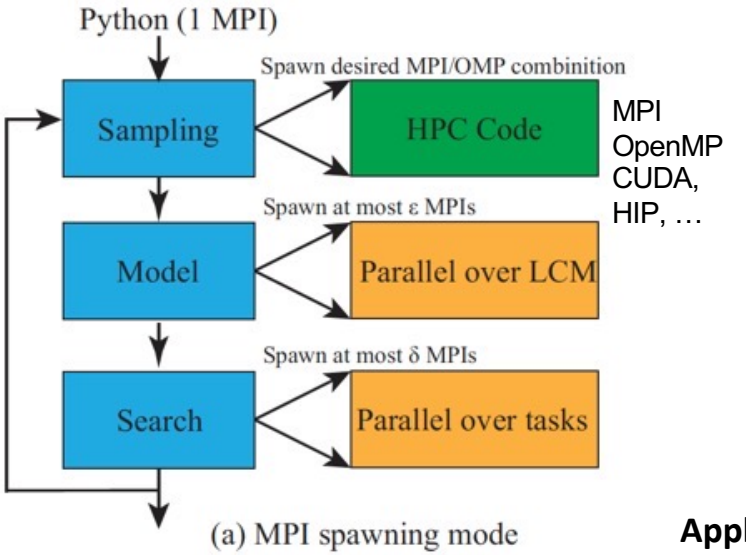
- Performance depends on input, machines, software stack …
- Find optimal parameter configuration using small number of runs



1. choose a promising parameter configuration
2. run the app for the chosen parameter configuration

GPTune tuning workflow on parallel machines

Parallel execution model



GPTune advanced features

- History database
- Multi-objective optimization
- Multi-fidelity optimization
- Hybrid model (MCST + GP) for mixed input space
- Clustered GP for non-smooth function surface
- Users' performance models or hardware performance counters to guide tuning

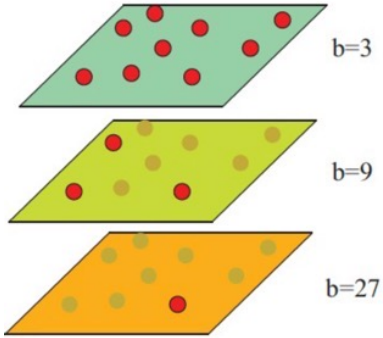
Applications: Hypre, MFEM, STRUMPACK, SuperLU_DIST, PLASMA, SLATE, ScaLAPACK, NIMROD, M3D-C1, IMPACT-Z, CNN, GCN, KRR, sketching-based linear least-square solvers

GPTune: Multitask Learning for Autotuning Exascale Applications, Proc. of Principles and Practice of Parallel Programming, 2021.
GPTuneBand: Multi-task and Multi-fidelity Autotuning for Large-scale High Performance Computing Applications, SIAM PP22.
Non-smooth Bayesian Optimization in Tuning Problems, arXiv preprint arXiv:2109.07563
Enhancing Autotuning Capability with a History Database, McSOC-2021, Special Session: Autotuning for Multicore & GPU.

Multi-fidelity tuning for hypre

parallel multigrid solver

- Multi-armed bandit strategy (MAB)
- Each arm corresponds to a fidelity
  - Use more samples at low fidelity to reduce space
  - Use fewer samples at high fidelity
- LCM is built across arms and tasks



- 3d convection-diffusion equation in a k^3 grid, $10 < \mathrm{k} < 100$

$$-c\,\Delta u + a\,\nabla \cdot u = f, \qquad a, c \in [0,1]$$

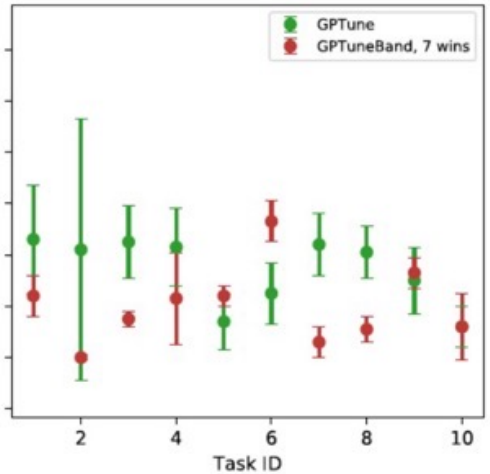
Zhu, Liu, Ghysels, Bindel, L, SIAM PP2020 Proceedings

Multi-fidelity tuning for hypre

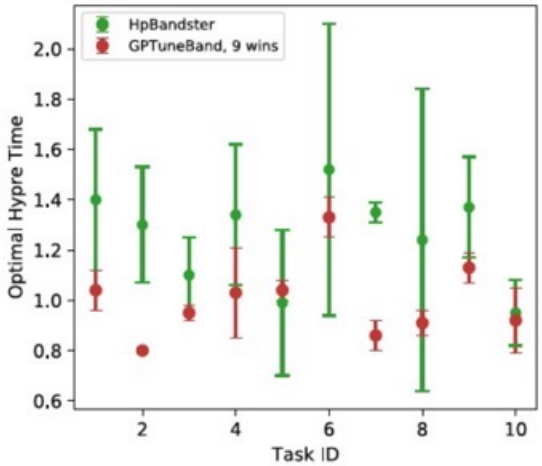
parallel multigrid solver

- $\mathbb{IS} = [\mathrm{a}, \mathrm{c}]$, $\mathbb{PS} = 12$ integer/real/categorical, $\mathbb{OS} = [\mathrm{time}]$
- Fidelity / budget ~ k^3
- 2 Cori nodes @ NERSC, 32 cores

Comparison of GPTuneBand vs GPTune & HpBandster



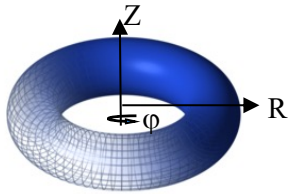
GPTuneBand wins GPTune 7/10

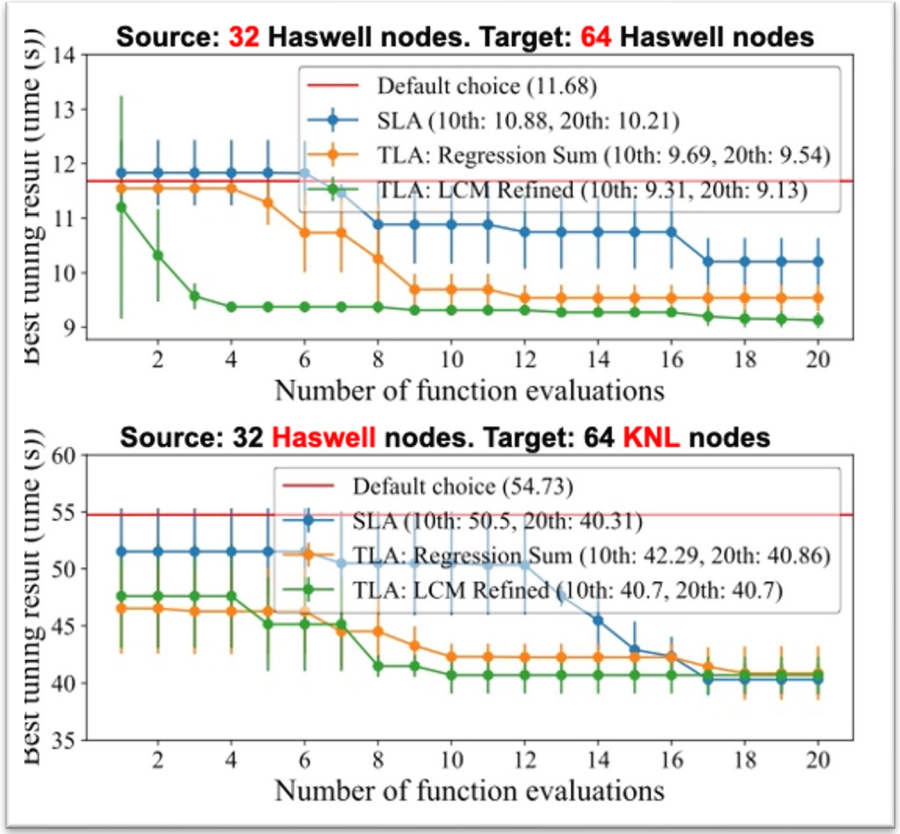


GPTuneBand wins HpBandster 9/10

Tuning ITER tokamak design for fusion energy

- Being constructed in St. Paul-lez-Durance, France
- Cost $20B+
- NIMROD and M3D-C1 modeling codes
  - GMRES to solve 3D, SuperLU_DIST for each 2D plane preconditioner
- Transfer learning (TLA) aided by database





NIMROD (fusion simulation)