hypre: High Performance Linear Solvers and Preconditioners

Rob Falgout, Ruipeng Li, Sarah Osborn, Bjorn Sjogreen, and Ulrike Meier Yang
Center for Applied Scientific Computing, Lawrence Livermore National Laboratory

Overview

What is hypre?

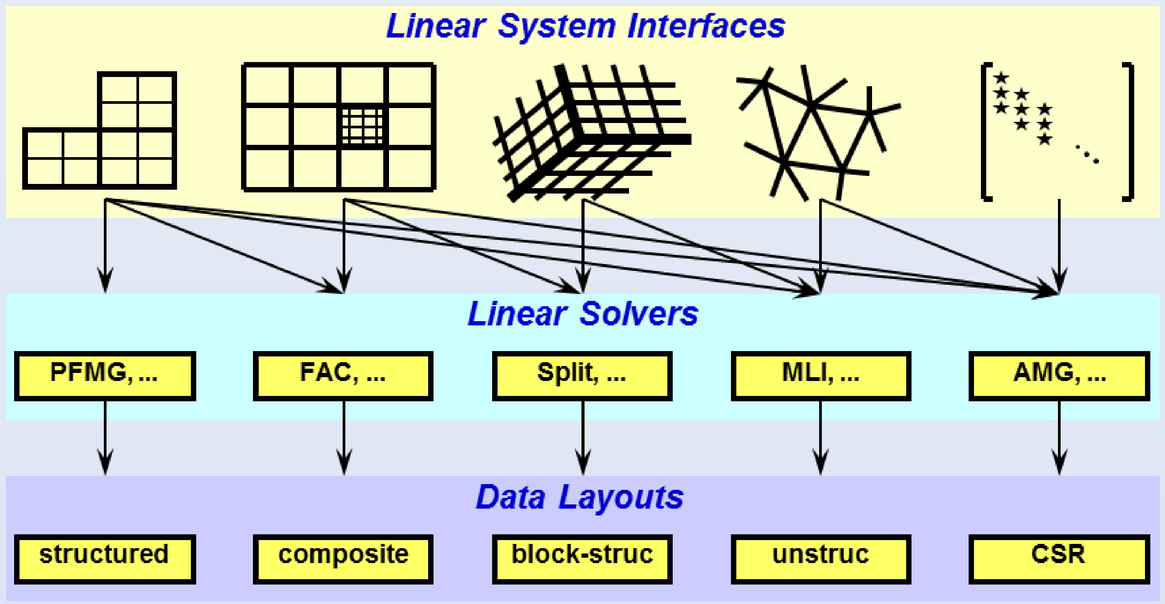
A library of high-performance preconditioners and solvers for the solution of linear systems of equations on massively parallel computers.

Notable features:

- A variety of "conceptual" interfaces that allow users to describe their problem in a natural way.
- State-of-the-art preconditioners and solvers that have been proven scalable on more than a million of processes.
- Easy to use and scalable!

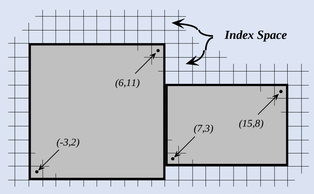
Conceptual Interfaces

- Multiple interfaces are necessary to provide the most efficient solvers and data layouts.



Structured Grid System Interface (Struct)

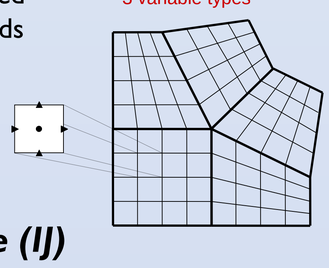
- Appropriate for scalar problems on logically rectangular grids
- Describe problem via grids and stencils

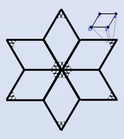


Semi-Structured Grid System Interface (SStruct)

- Allows more general grids that are not entirely structured: block-structured grids, structured adaptive mesh refinement grids, overset grids
- Allows multiple variables
- Describe problem via multiple grids, stencils, and unstructured connections

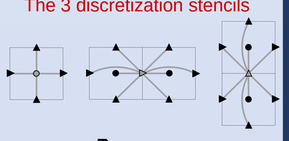
A block-structured grid with 3 variable types





Finite element discretization

The 3 discretization stencils



Linear-Algebraic System Interface (IJ)

- Appropriate for general sparse linear systems
- Describe problem in terms of row and column entries

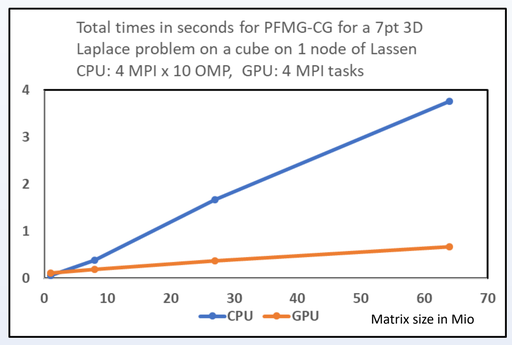
Solvers

Linear solver availability via *hypre*'s conceptual interfaces for selected solvers

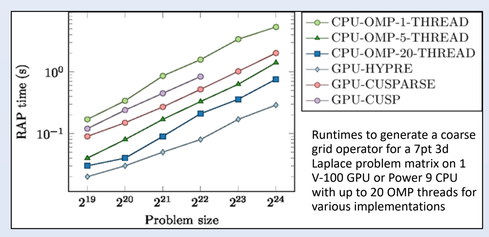
Data Layouts	Solvers	Interfaces: Struct	SStruct	IJ
Structured	SMG	✓	✓	
	PFMG	✓	✓	
Semi-structured	Split		✓	
	SysPFMG		✓	
Sparse matrix	ADS		✓	✓
	AMS		✓	✓
	BoomerAMG		✓	✓
	ParaSails		✓	✓
	Euclid		✓	✓
	PCG	✓	✓	✓
Matrix free	GMRES	✓	✓	✓
	BiCGSTAB	✓	✓	✓
	Hybrid	✓	✓	✓

Enabling *hypre* for GPUs

- ***Efforts in Structured Interface***
  - For SMG, PFMG and structured Krylov solvers loops on data structures are performed using macros called BoxLoops.
  - BoxLoops were redesigned to allow use of Cuda, OpenMP 4.5, RAJA, and Kokkos.
  - Consequently both setup and solve phases of these solvers are now fully GPU-enabled.
  - No use of unified memory

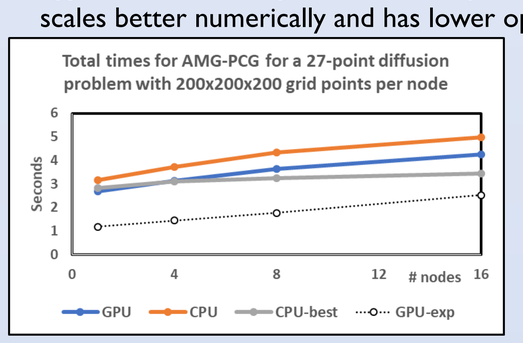


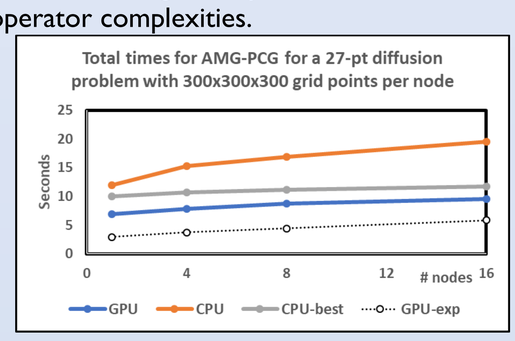
- ***Efforts in Unstructured Interface***
  - Solve phase can be performed on the GPU using Jacobi-type smoothers:
    - Requires unified memory
    - Both CUDA, and OpenMP 4.5 available
    - Uses CuSparse matrix-vector multiplication routine
  - Implementation of the following AMG setup components in CUDA:
    - Generation of strength matrix
    - PMIS coarsening
    - Matrix-matrix multiplication for coarse grid generation
    - Direct interpolation



Runtimes to generate a coarse grid operator for a 7pt 3d Laplace problem matrix on 1 V-100 GPU or Power 9 CPU with up to 20 OMP threads for various implementations

  - It is now possible to perform the AMG setup completely on the GPU using PMIS and direct interpolation!
  - Weak scaling study of AMG-PCG using *hypre* release 2.18.2 on up to 16 nodes of Lassen with 4 MPI tasks per node (and 10 OMP threads per MPI task for CPU versions) of GPU-only version and 2 CPU-only versions.
    'CPU' uses the same parameters as 'GPU'. 'CPU-best' replaces direct interpolation with aggressive coarsening and ext+i interpolation, which is generally recommended, since it scales better numerically and has lower operator complexities.



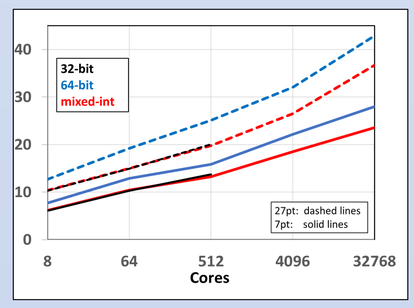


  - Significant improvements of 'GPU' possible through removing of unified memory use and the application of memory pools, as was done for 'GPU-exp'.

- ***Future Plans***
  - Remove the use of unified memory and include memory pools to improve performance on GPU for next *hypre* release
  - Continue to enhance GPU support for AMG, including more smoothers, GPU-enhanced AMG setup routines and improved interfacing between applications and *hypre*
  - Redesign interpolation operators in terms of sparse matrix-matrix operations for easier GPU implementation and improved portability
  - Add OpenMP 4.5 capabilities
  - Design and implement semi-structured AMG solver

Mixed Integer Use in *hypre*

- Implemented a new integer type HYPRE_BigInt in hypre to reduce use of 64-bit integers for large problems and improve performance and memory use
- The new mixed-int capability performs about 20-25% better than the 64-bit version while using less memory and can solve larger problems than 32-bit.



Weak scaling study: Total runtimes in seconds for AMG-PCG using 1M points/core for 2 different 3D diffusion problems on Vulcan

Additional Information

- **Public repository**: https://github.com/hypre-space/hypre
- **Website:** https://www.llnl.gov/casc/hypre
- **For bug reports and questions contact** hypre-support@llnl.gov

This research was supported by the ExascaleComputing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nucear Security Administration.
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.