FIND OUT MORE AT **http://icl.utk.edu**

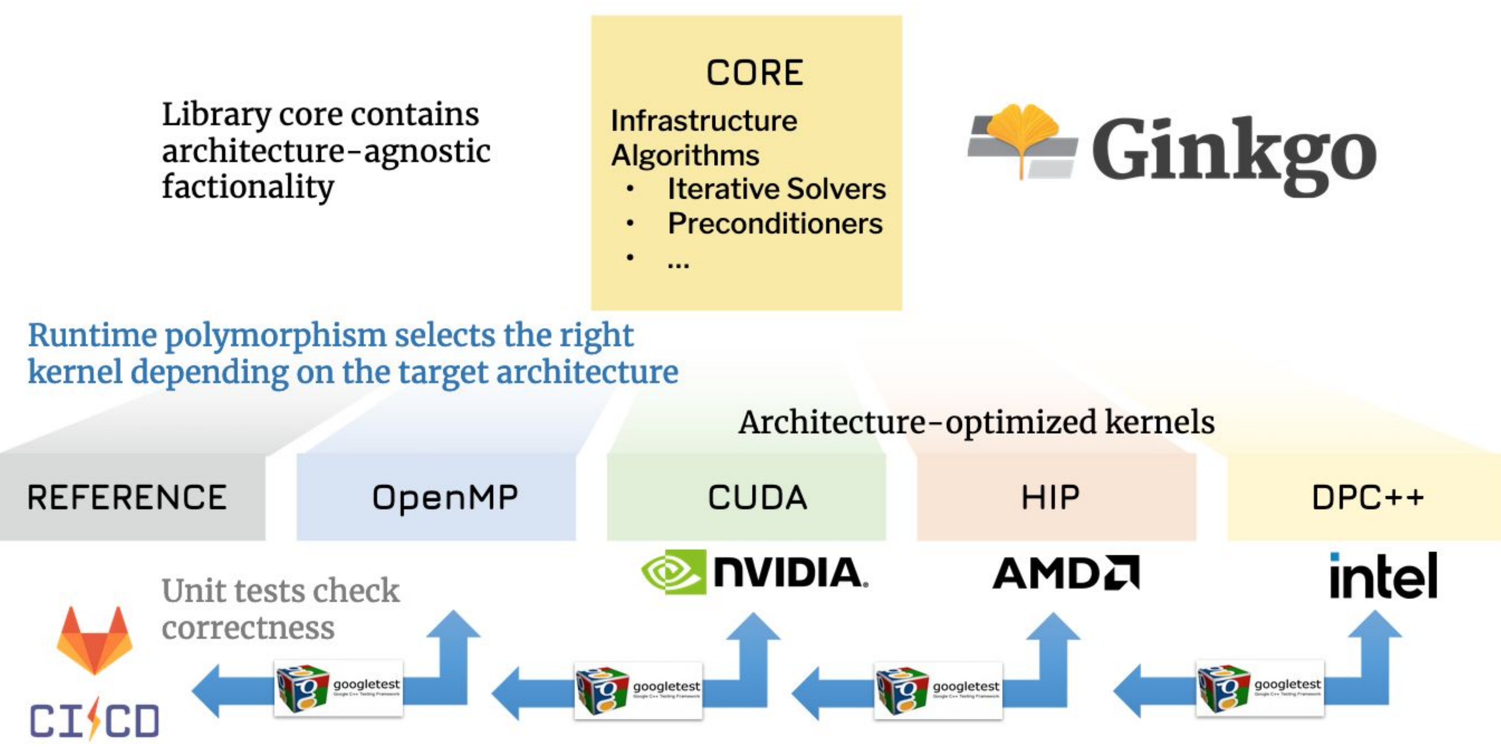
Ginkgo

High Performance Linear Algebra on GPUs

I. Aggarwal, H. Anzt, N. Beams, T. Cojean, Y. Chen, T. Grützmacher, F. Göbel, M. Koch, G. Olenik, P. Nayak, T. Ribizel, Y. Tsai
University of Tennessee, USA
Karlsruhe Institute of Technology, Germany

OVERVIEW

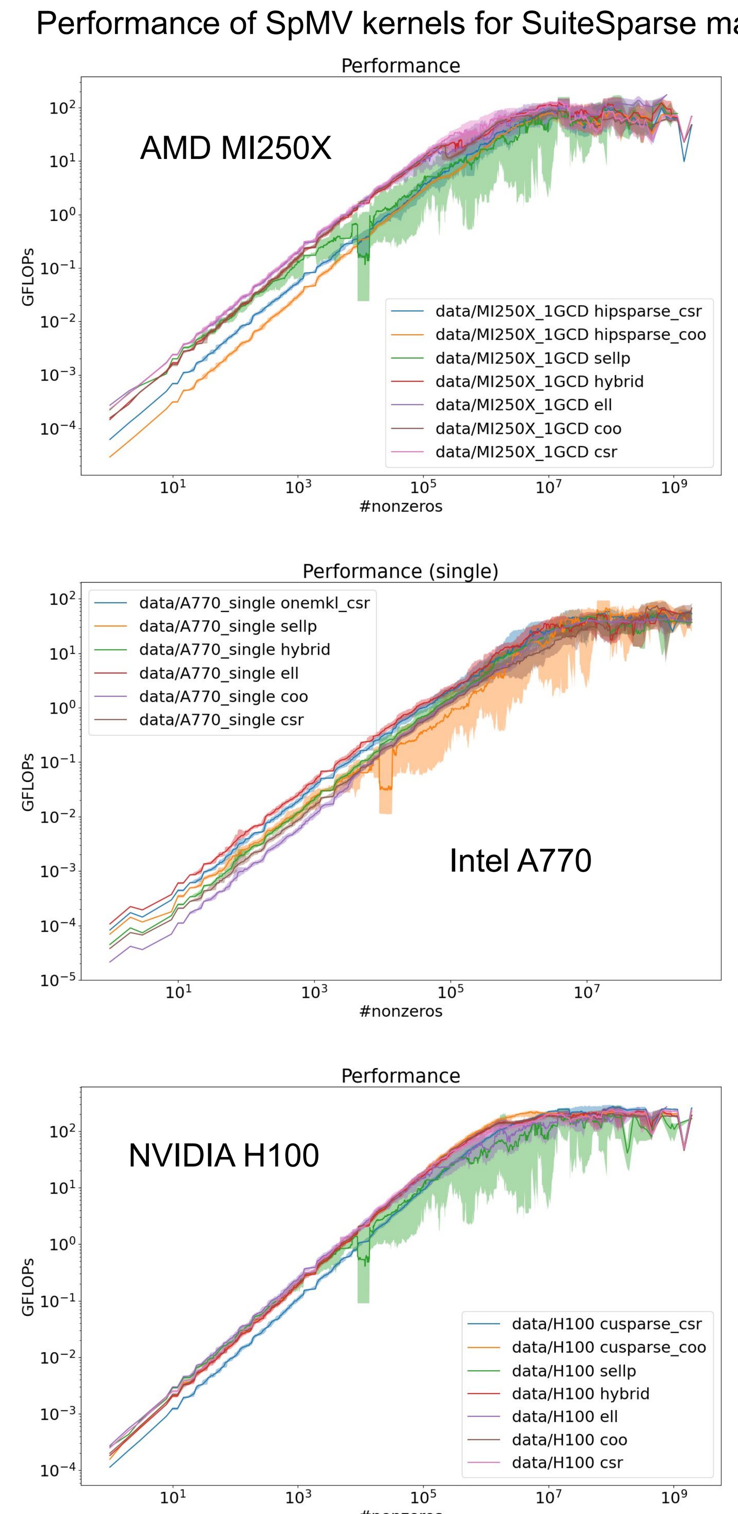
Ginkgo is a C++ framework for **sparse linear algebra**. Using a universal **linear operator abstraction,** Ginkgo provides **basic building blocks** like the sparse matrix vector product for a variety of matrix formats, **iterative solvers**, **preconditioners**, and **batched routines**. Ginkgo targets **multi- and many-core** systems, and currently features back-ends for AMD GPUs, Intel GPUs, NVIDIA GPUs, and OpenMP-supporting architectures. Core functionality is separated from hardware-specific kernels for easy extension to other architectures, with **runtime polymorphism** selecting the proper kernels.



FUNCTIONALITY

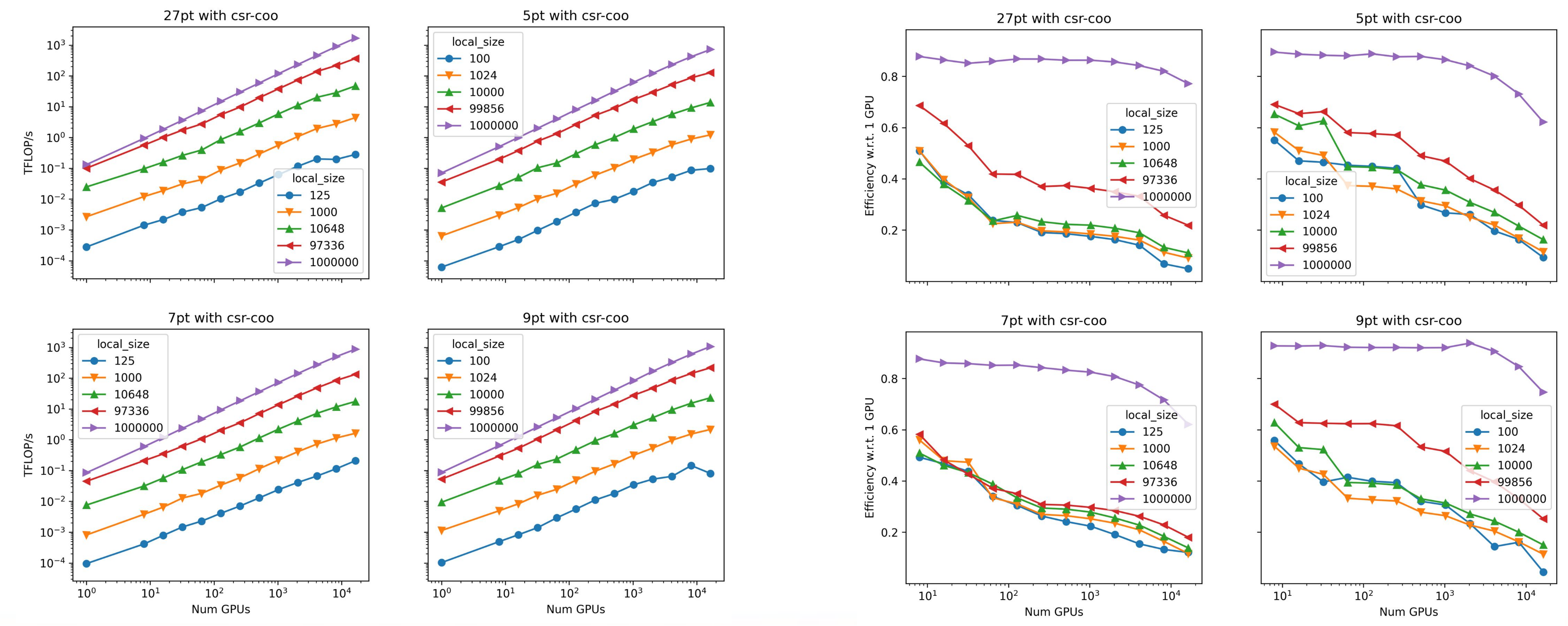
	Functionality	OMP	CUDA	HIP	DPC++
Basic	SpMV	✓	✓	✓	✓
	SpMM	✓	✓	✓	✓
	SpGeMM	✓	✓	✓	✓
Krylov solvers	BiCG	✓	✓	✓	✓
	BiCGSTAB	✓	✓	✓	✓
	CG	✓	✓	✓	✓
	CGS	✓	✓	✓	✓
	GMRES	✓	✓	✓	✓
	IDR	✓	✓	✓	✓
Preconditioners	(Block-)Jacobi	✓	✓	✓	✓
	ILU/IC		✓	✓	✓
	Parallel ILU/IC	✓	✓	✓	✓
	Parallel ILUT/ICT	✓	✓	✓	✓
	Sparse Approximate Inverse	✓	✓	✓	✓
Batched	Batched BiCGSTAB	✓	✓	✓	✓
	Batched CG	✓	✓	✓	✓
	Batched GMRES	✓	✓	✓	✓
	Batched ILU	✓	✓	✓	✓
	Batched ISAI	✓	✓	✓	✓
	Batched Jacobi	✓	✓	✓	✓
AMG	AMG preconditioner	✓	✓	✓	✓
	AMG solver	✓	✓	✓	✓
	Parallel Graph Match	✓	✓	✓	✓
Sparse direct	Symbolic Cholesky	✓	✓	✓	✓
	Numeric Cholesky	UNDER DEVELOPMENT			
	Symbolic LU	✓	✓	✓	✓
	Numeric LU	✓	✓	✓	
	Sparse TRSV	✓	✓	✓	
Utilities	On-Device Matrix Assembly	✓	✓	✓	✓
	MC64/RCM reordering	✓			
	Wrapping user data	✓ (all)			
	Logging	✓ (all)			
	PAPI counters	✓ (all)			

Performance of SpMV kernels for SuiteSparse matrices



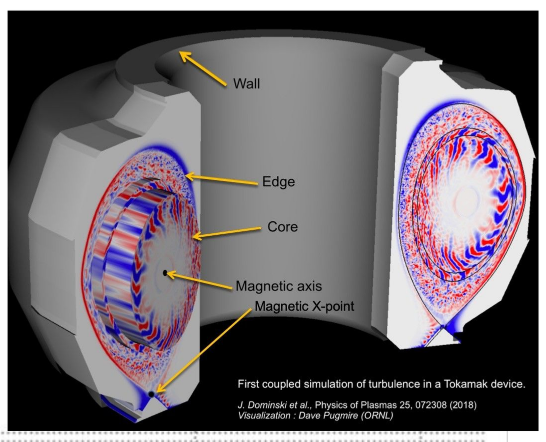
FRONTIER SCALING

Ginkgo was able to run on the ECP Frontier Exascale machine (current TOP500 #1) and reach **up to 2 PFlop/s** on **20.000 GPUs** with weak scaling, therefore reaching an **efficiency of over 80%** with respect to using 1 GPU.

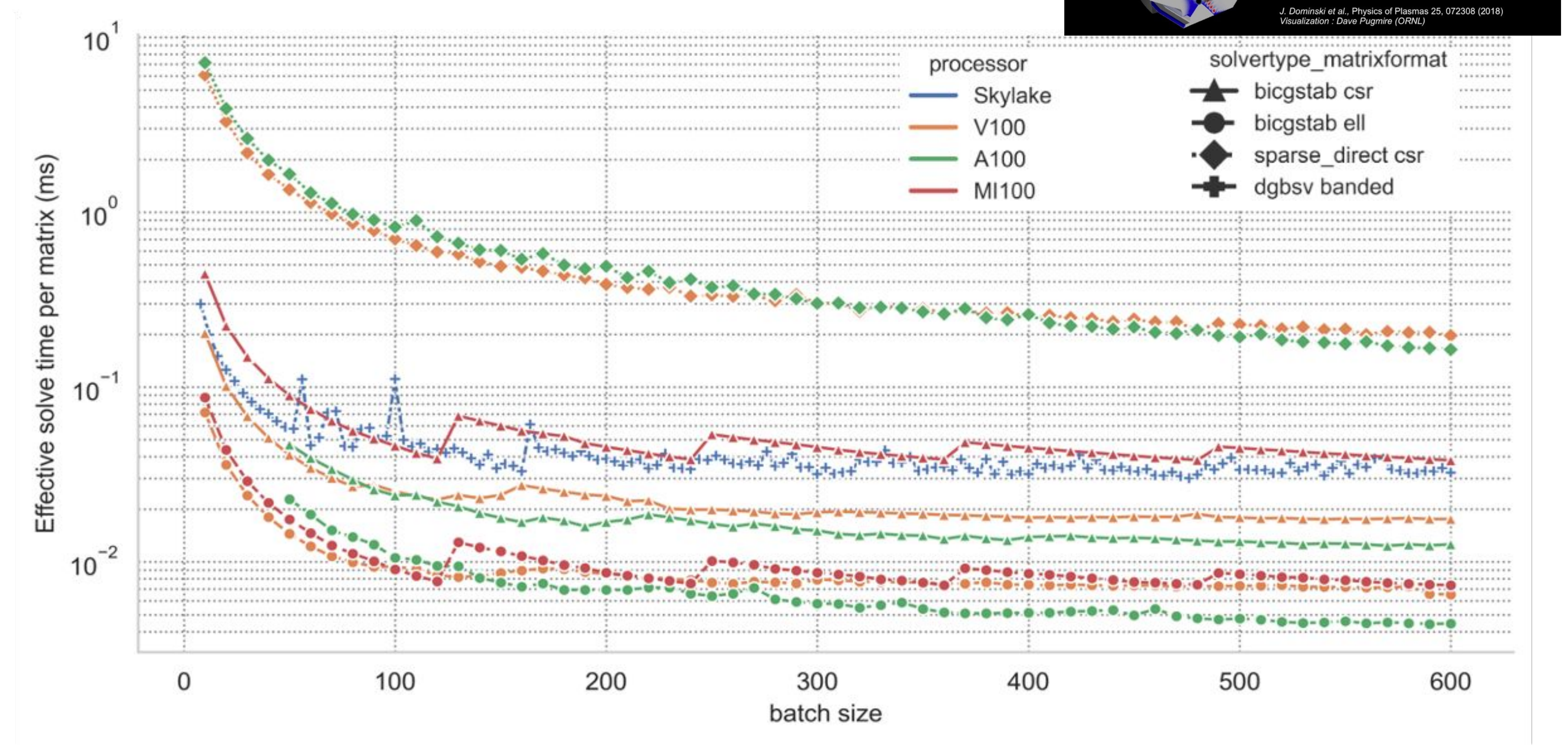


APPLICATION INTEGRATION

Batched Iterative Solvers in Plasma Simulations

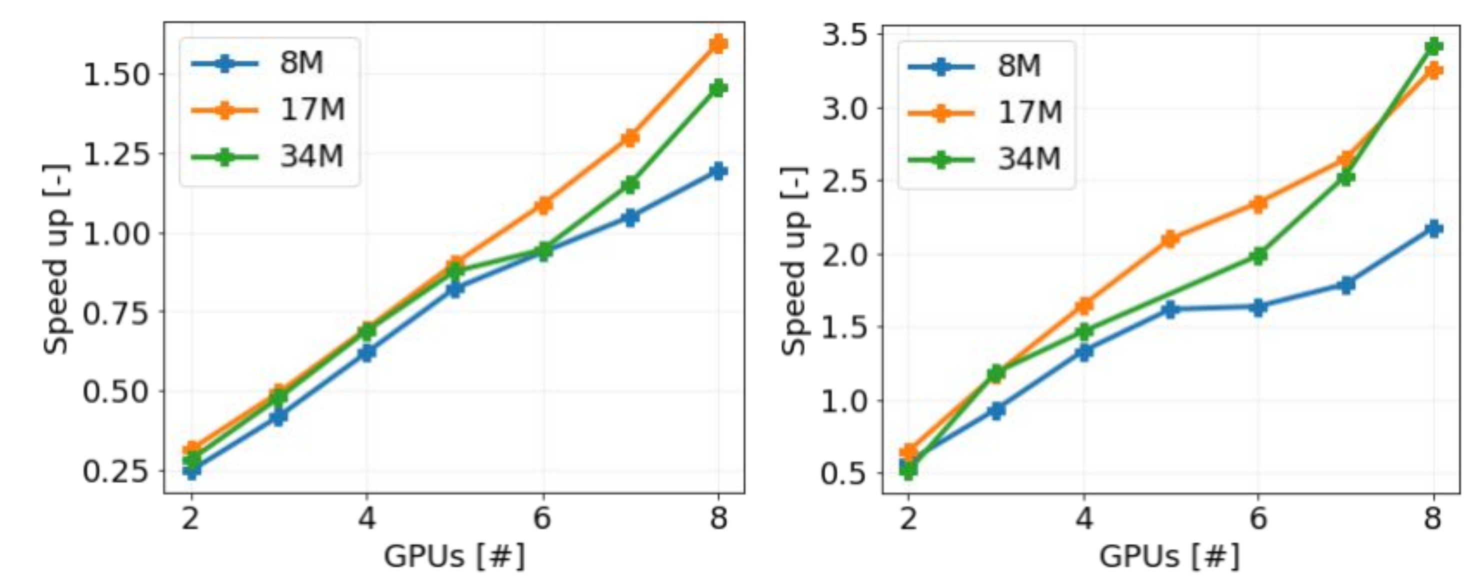


XGC is a gyrokinetic particle-in-cell code, which specializes in the simulation of the edge region of magnetically confined thermonuclear fusion plasma. Ginkgo is used to solve the velocity problems at different mesh nodes in a batched manner. https://xgc.pppl.gov/



Accelerating OpenFOAM Simulations

OpenFOAM is an open source CFD code heavily used for research and development in academia and industry. Ginkgo can be used as numerical backend to accelerate OpenFOAM simulations.

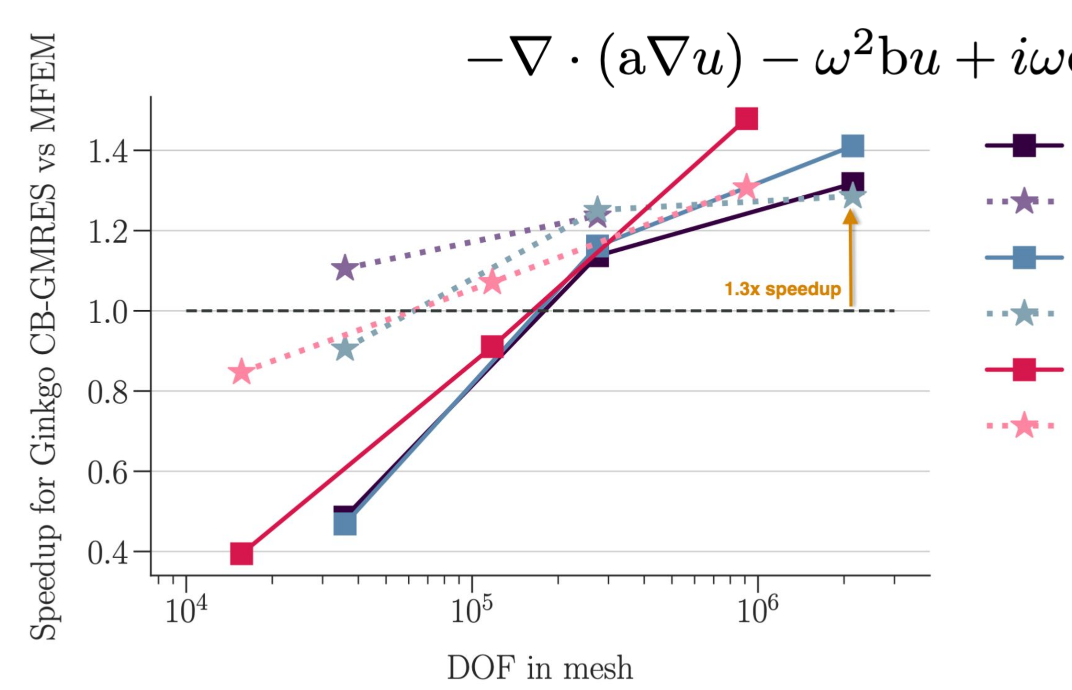


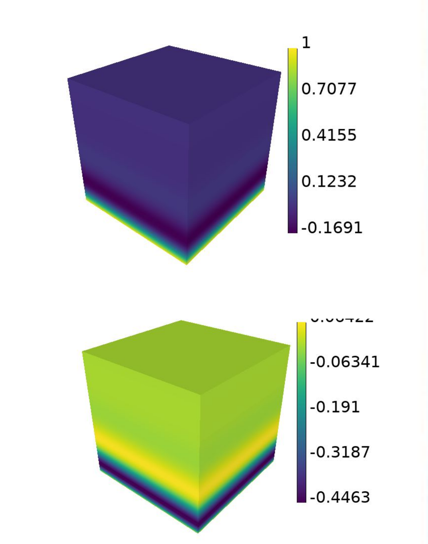
Speedup of a CG solver using a distributed Schwarz preconditioner with ISAI (left), and Multigrid (right) as local inner solver on multiple MI100s in comparison to the execution on 36 MPI ranks on an AMD EPYC 7302 with an IC preconditioner.

Speeding up MFEM's "example 22" on GPUs

Example 22 of the MFEM finite element library solves harmonic oscillation problems, with a forced oscillation imposed at the boundary. In this test, we use variant 1:

$$-\nabla \cdot (a\nabla u) - \omega^2 bu + i\omega cu = 0$$





Real part of solution (top), imaginary part of solution

Speedup of Ginkgo's Compressed Basis-GMRES solver vs MFEM's GMRES solver for three different orders of basis functions (p), using MFEM matrix-free operators and the Ginkgo-MFEM integration wrappers in MFEM. CUDA 10.1/V100 and ROCm 4.0/MI50.

Stakeholder

deal.II HYTEG OpenFOAM *The Open Source CFD Toolbox* openCARP sundials Spack xSDK E4S

EXASCALE COMPUTING PROJECT
EuroHPC Joint Undertaking
SPONSORED BY THE Federal Ministry of Education and Research
ICL INNOVATIVE COMPUTING LABORATORY
THE UNIVERSITY OF TENNESSEE KNOXVILLE
KIT Karlsruher Institut für Technologie