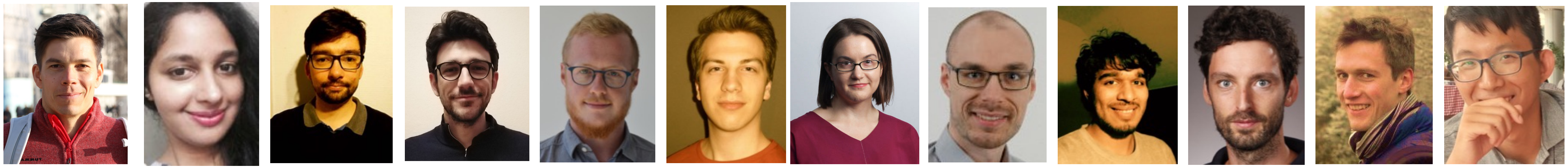
22d IEEE International Symposium on Parallel and Distributed Computing (ISPDC 2023)
Bucharest, July 10, 2023

The Role of Software in HPC – Lessons Learnt in the US Exascale Computing Project

EXASCALE COMPUTING PROJECT

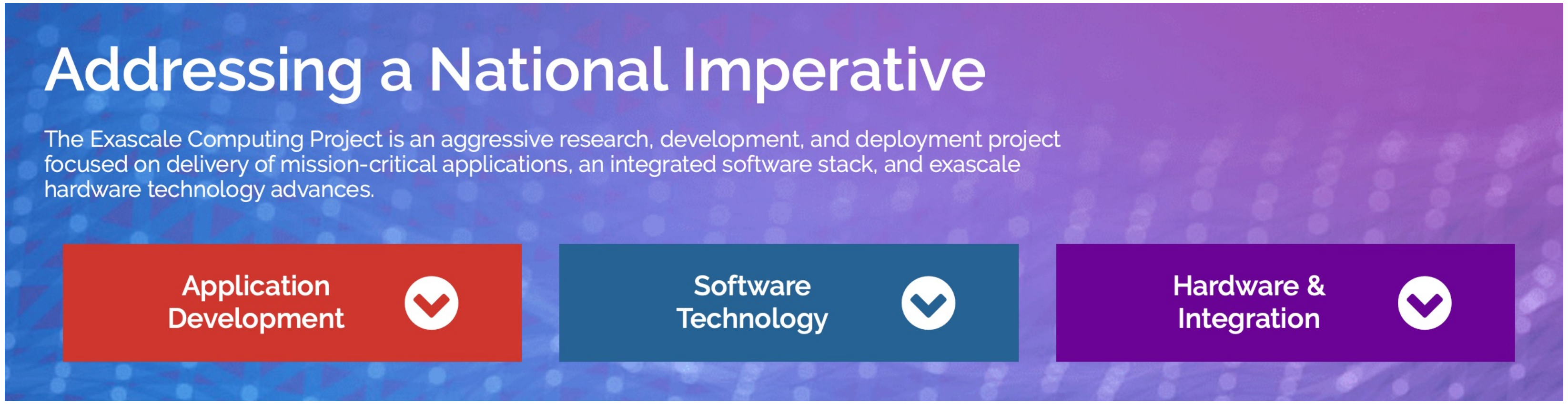
Hartwig Anzt, University of Tennessee



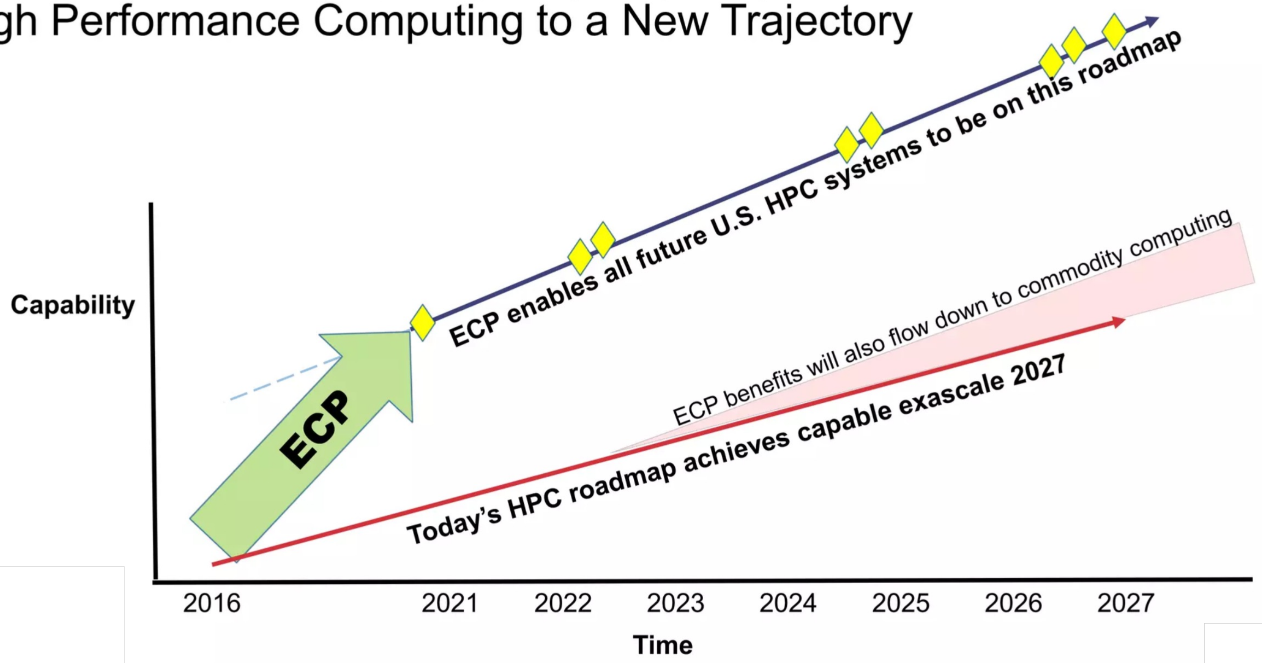
National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY | Office of Science

The US Exascale Computing Project



Vision: Exascale Computing Project (ECP) Lifts all U.S. High Performance Computing to a New Trajectory



The US Exascale Computing Project

Advancing Scientific Discovery

The ECP aims to ensure availability of the exascale computing ecosystem necessary for developing clean energy systems, improving the resilience of our infrastructure, designing new materials that can perform in extreme environments, adapting to changes in the water cycle, developing smaller and more powerful accelerators for use in medicine and industry, and much more. Several projects focus on data-intensive problems to enable effective use of the data streams from powerful scientific facilities, complex environmental genomes, and cancer research (patient genetics, tumor genomes, molecular simulations, and clinical data).

Strengthening National Security

The ECP teams are also developing new applications for supporting the NNSA Stockpile Stewardship Program, which is responsible for maintaining the readiness and reliability of our nuclear weapons systems—without underground testing. Assessing the performance of weapons systems subject to hostile environments and potential threat scenarios exceeds the capabilities of current HPC systems and codes. NNSA application projects are focused on providing the sophisticated modeling and analysis tools needed to sustain the U.S. nuclear deterrence.

Improving Industrial Competitiveness

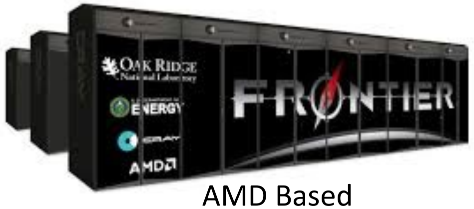
Exascale systems will be used to accelerate research that leads to innovative products and speeds commercialization, creating jobs and driving US competitiveness across industrial sectors, such as the emerging energy economy. To ensure alignment with US industry needs, the ECP is engaging senior technology decision makers from among the country’s most prominent private sector companies.

The US Exascale Computing Project

US$4B – what is it spent on?

- 3 computers
  - $600M each
  - $400M to vendors for Design, Path, Fast - Forward

21 Applications



AMD Based
(Up & running)

Intel Based
(Being installed)

AMD Based
(Planned)

Domain*	Base Challenge Problem
Wind Energy	2x2 5 MW turbine array in 3x3x1 km^3 domain
Nuclear Energy	Small Modular Reactor with complete in-vessel coolant loop
Fossil Energy	Burn fossil fuels cleanly with CLRs
Combustion	Reactivity controlled compression ignition
Accelerator Design	TeV-class 10^{2-3} times cheaper & smaller
Magnetic Fusion	Coupled gyrokinetics for ITER in H-mode
Nuclear Physics: QCD	Use correct light quark masses for first principles light nuclei properties
Chemistry: GAMESS	Heterogeneous catalysis: MSN reactions
Chemistry: NWChemEx	Catalytic conversion of biomass
Extreme Materials	Microstructure evolution in nuclear matls
Additive Manufacturing	Born-qualified 3D printed metal alloys

Domain*	Challenge Problem
Quantum Materials	Predict & control matls @ quantum level
Astrophysics	Supernovae explosions, neutron star mergers
Cosmology	Extract "dark sector" physics from upcoming cosmological surveys
Earthquakes	Regional hazard and risk assessment
Geoscience	Well-scale fracture propagation in wellbore cement due to attack of CO_2-saturated fluid
Earth System	Assess regional impacts of climate change on the water cycle @ 5 SYPD
Power Grid	Large-scale planning under uncertainty; underfrequency response
Cancer Research	Scalable machine learning for predictive preclinical models and targeted therapy
Metagenomics	Discover and characterize microbial communities through genomic and proteomic analysis
FEL Light Source	Protein and molecular structure determination using streaming light source data

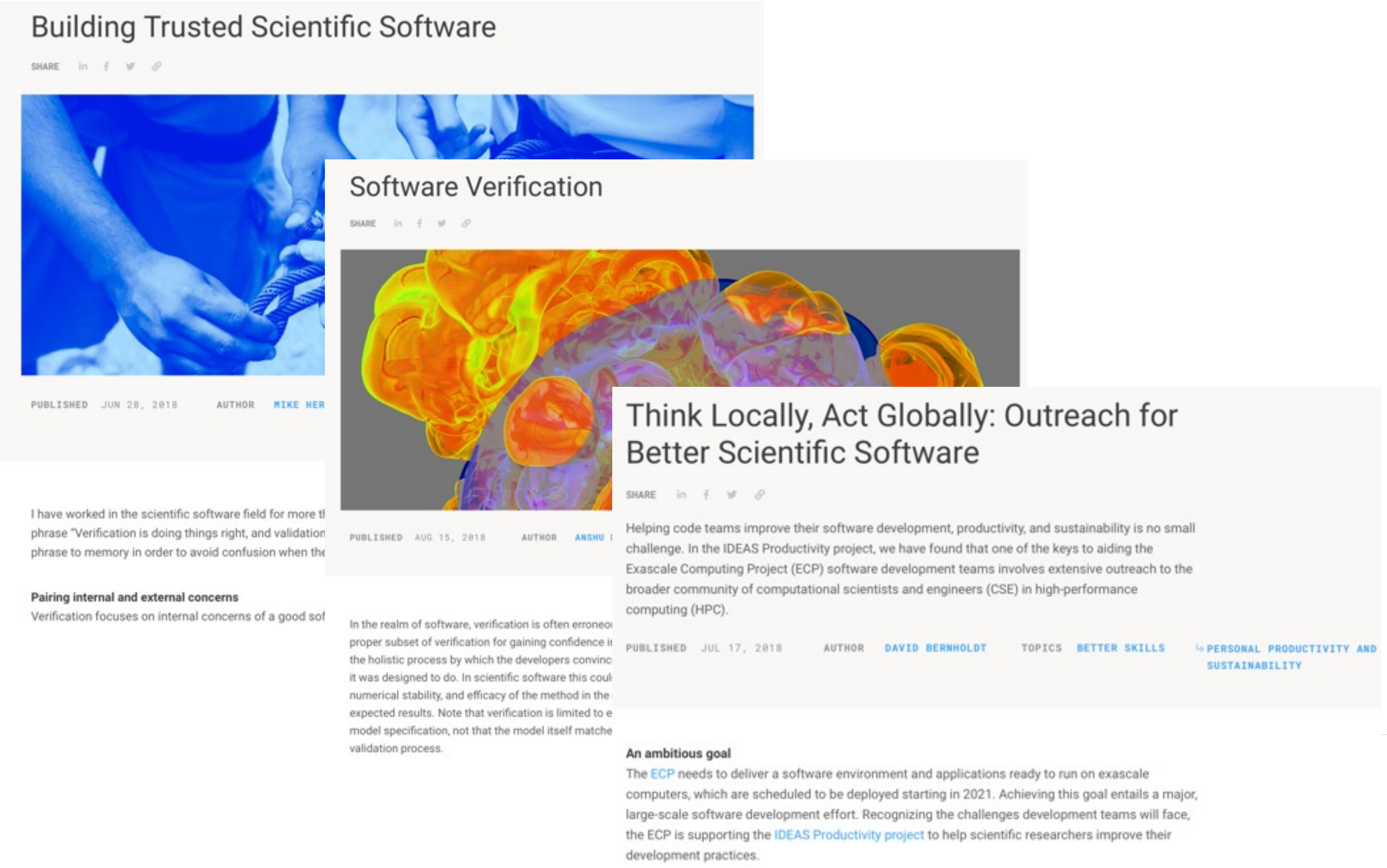
Sustainable software development

PMR Core (17)	Compilers and Support (7)	Tools and Technology (11)	xSDK (16)	Visualization Analysis and Reduction (9)	Data mgmt, I/O Services, Checkpoint restart (12)	Ecosystem/E4S at-large (12)
QUO	openarc	TAU	hypre	ParaView	SCR	mpiFileUtils
Papyrus	Kitsune	HPCToolkit	FleSCI	Catalyst	FAODEL	TriBITS
SICM	LLVM	Dyninst Binary Tools	MFEM	VTK-m	ROMIO	MarFS
Legion	CHiLL autotuning comp	Gotcha	Kokkoskernels	SZ	Mercury (Mochi suite)	GUFI
Kokkos (support)	LLVM openMP comp	Caliper	Trilinos	zfp	HDF5	Intel GEOPM
RAJA	OpenMP V & V	PAPI	SUNDIALS	VisIt	Parallel netCDF	BEE
CHAI	Flang/LLVM Fortran comp	Program Database Toolkit	PETSc/TAO	ASCENT	ADIOS	FSEFI
PaRSEC*		Search (random forests)	libEnsemble	Cinema	Darshan	Kitten Lightweight Kernel
DARMA		Siboka	STRUMPACK	ROVER	UnifyCR	COOLR
GASNet-EX		C2C	SuperLU		VeloC	NRM
Qthreads		Sonar	ForTrilinos		IOSS	ArgoContainers
BOLT			SLATE		HXHIM	Spack
UPC++			MAGMA			
MPICH			DTK			
Open MPI			Tasmanian			
Umpire			Ginkgo			
AML						

Legend:
- PMR
- Tools
- Math Libraries
- Data and Vis
- Ecosystems and delivery

ICL INNOVATIVE COMPUTING LABORATORY

Designing an ECP library for sustaining simulation performance



Building Trusted Scientific Software

SHARE

PUBLISHED JUN 28, 2018 AUTHOR MIKE HER

I have worked in the scientific software field for more th
phrase "Verification is doing things right, and validation
phrase to memory in order to avoid confusion when the

Pairing internal and external concerns

Verification focuses on internal concerns of a good sof

Software Verification

SHARE

PUBLISHED AUG 15, 2018 AUTHOR ANSHU I

In the realm of software, verification is often erroneo
proper subset of verification for gaining confidence i
the holistic process by which the developers convinc
it was designed to do. In scientific software this coul
numerical stability, and efficacy of the method in the
expected results. Note that verification is limited to e
model specification, not that the model itself matche
validation process.

Think Locally, Act Globally: Outreach for Better Scientific Software

SHARE

Helping code teams improve their software development, productivity, and sustainability is no small challenge. In the IDEAS Productivity project, we have found that one of the keys to aiding the Exascale Computing Project (ECP) software development teams involves extensive outreach to the broader community of computational scientists and engineers (CSE) in high-performance computing (HPC).

PUBLISHED JUL 17, 2018 AUTHOR DAVID BERNHOLDT TOPICS BETTER SKILLS PERSONAL PRODUCTIVITY AND SUSTAINABILITY

An ambitious goal

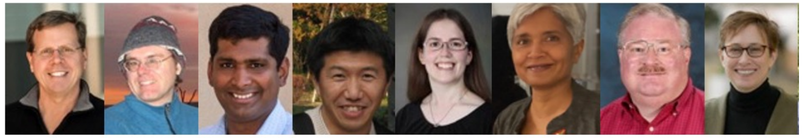
The ECP needs to deliver a software environment and applications ready to run on exascale computers, which are scheduled to be deployed starting in 2021. Achieving this goal entails a major, large-scale software development effort. Recognizing the challenges development teams will face, the ECP is supporting the IDEAS Productivity project to help scientific researchers improve their development practices.

IDEAS productivity

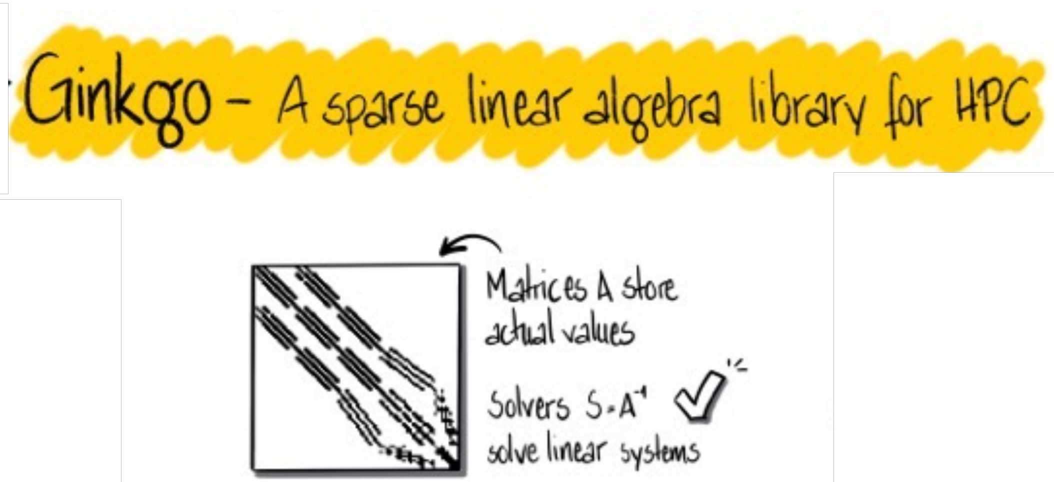
better scientific software

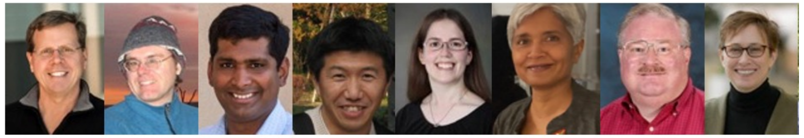
TRILINOS

EXASCALE COMPUTING PROJECT

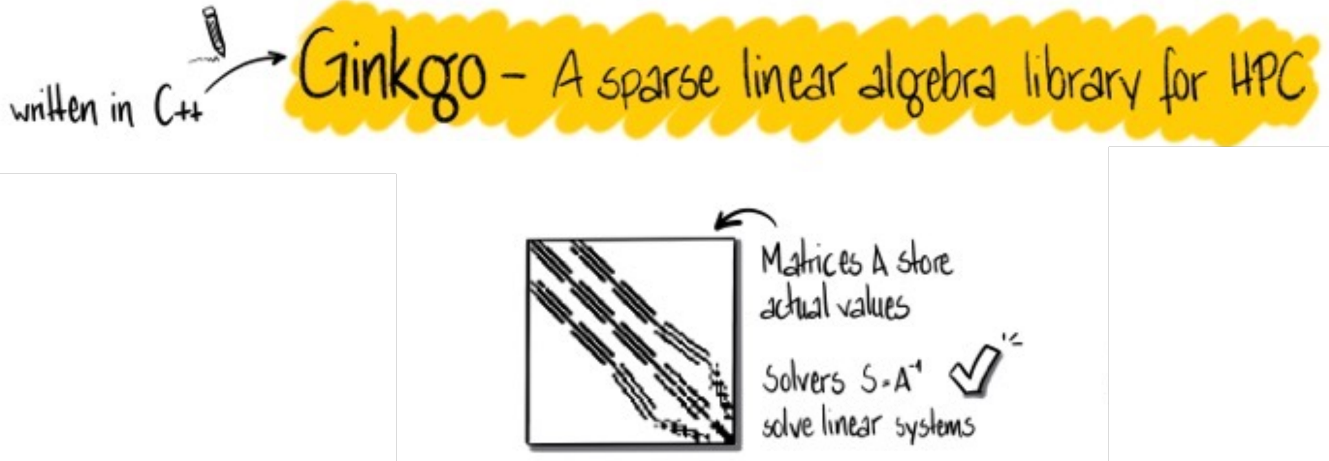


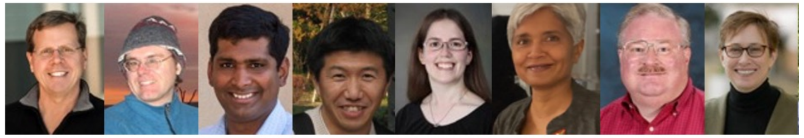
Designing an ECP library for sustaining simulation performance





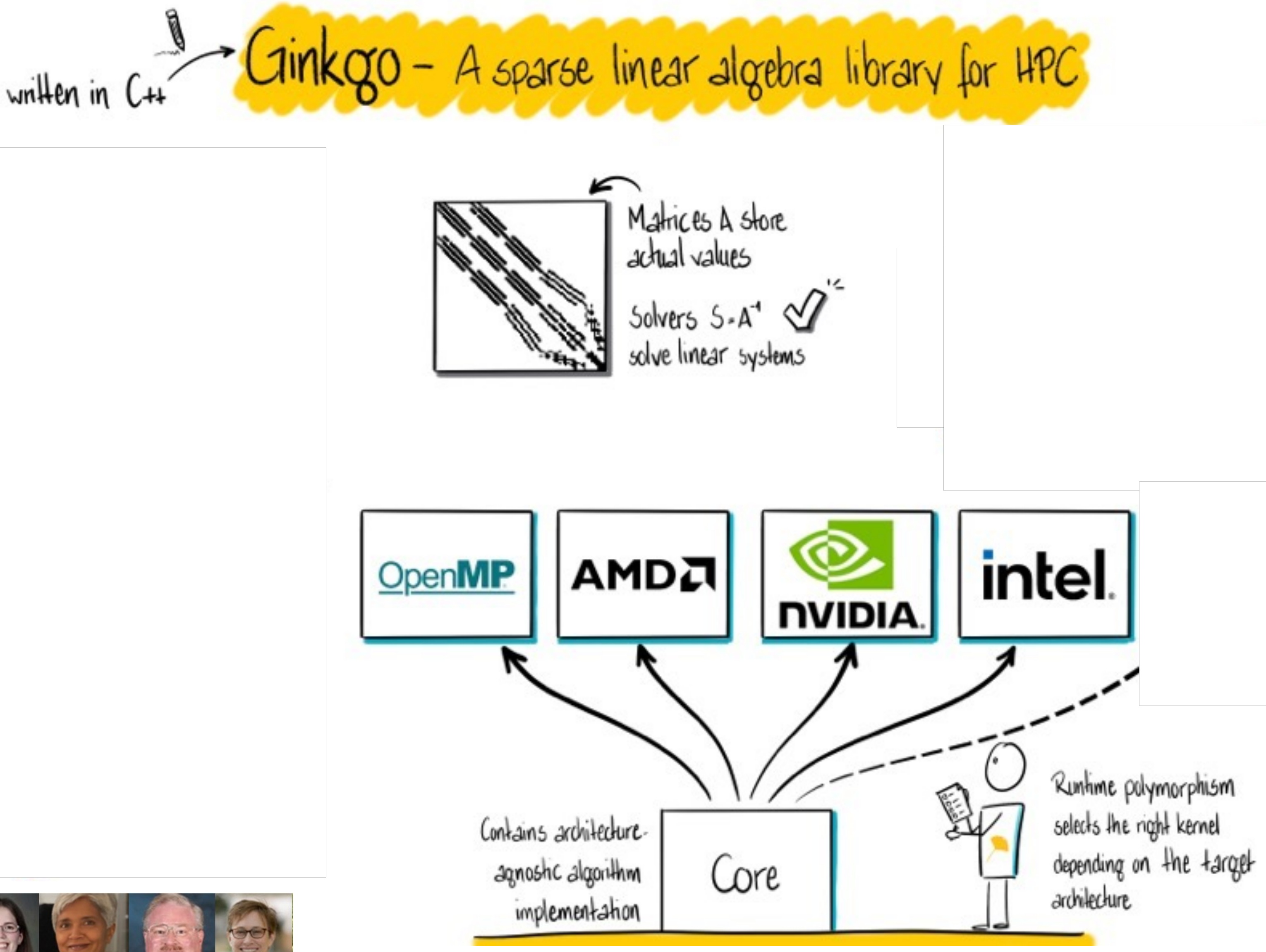
Designing an ECP library for sustaining simulation performance

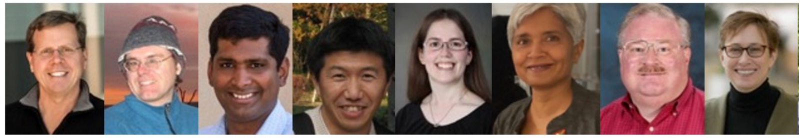




EXASCALE COMPUTING PROJECT

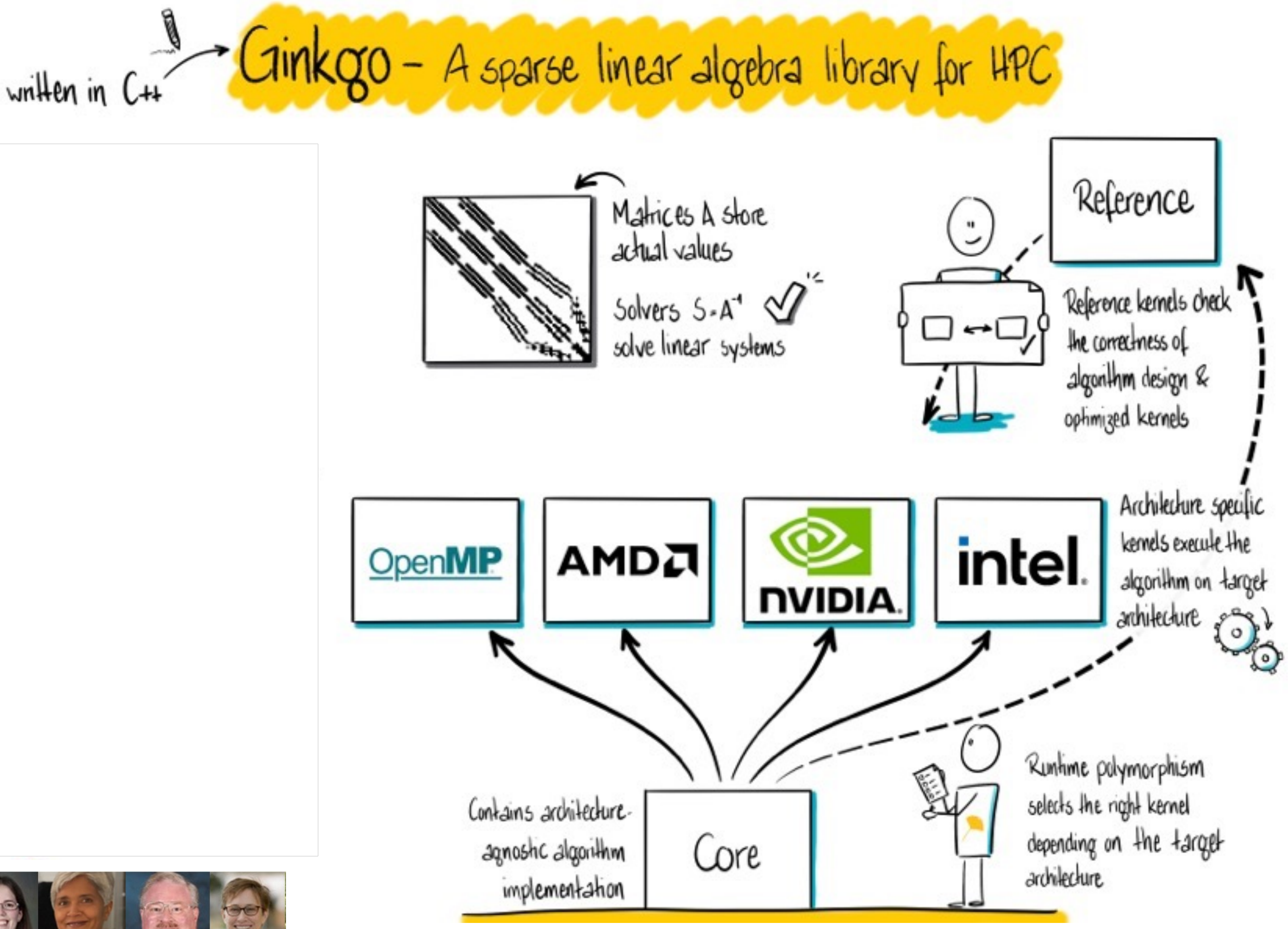
Designing an ECP library for sustaining simulation performance

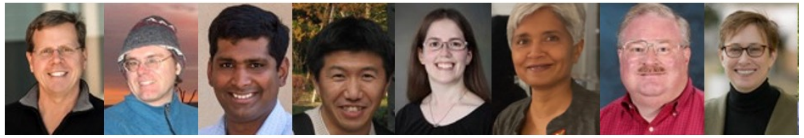




EXASCALE COMPUTING PROJECT

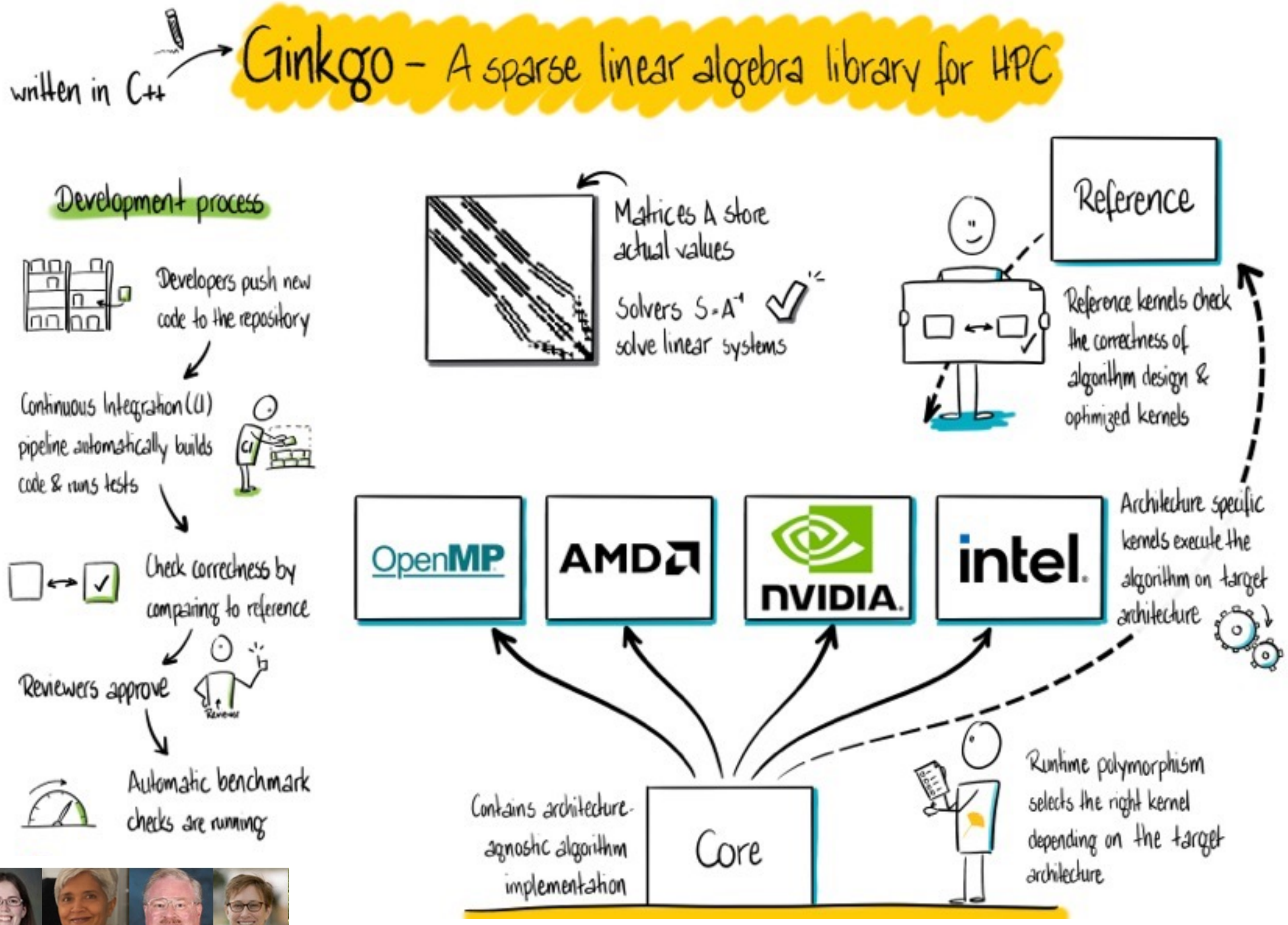
Designing an ECP library for sustaining simulation performance

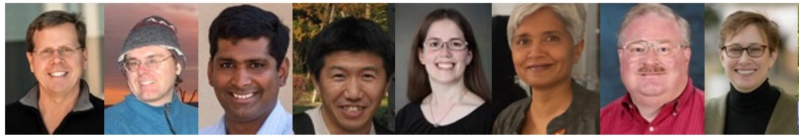




EXASCALE COMPUTING PROJECT

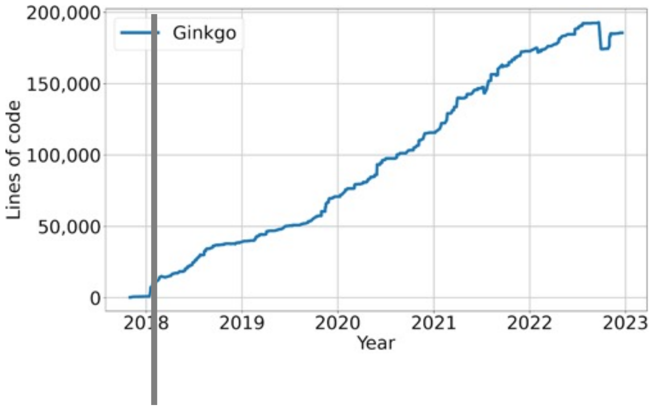
Designing an ECP library for sustaining simulation performance

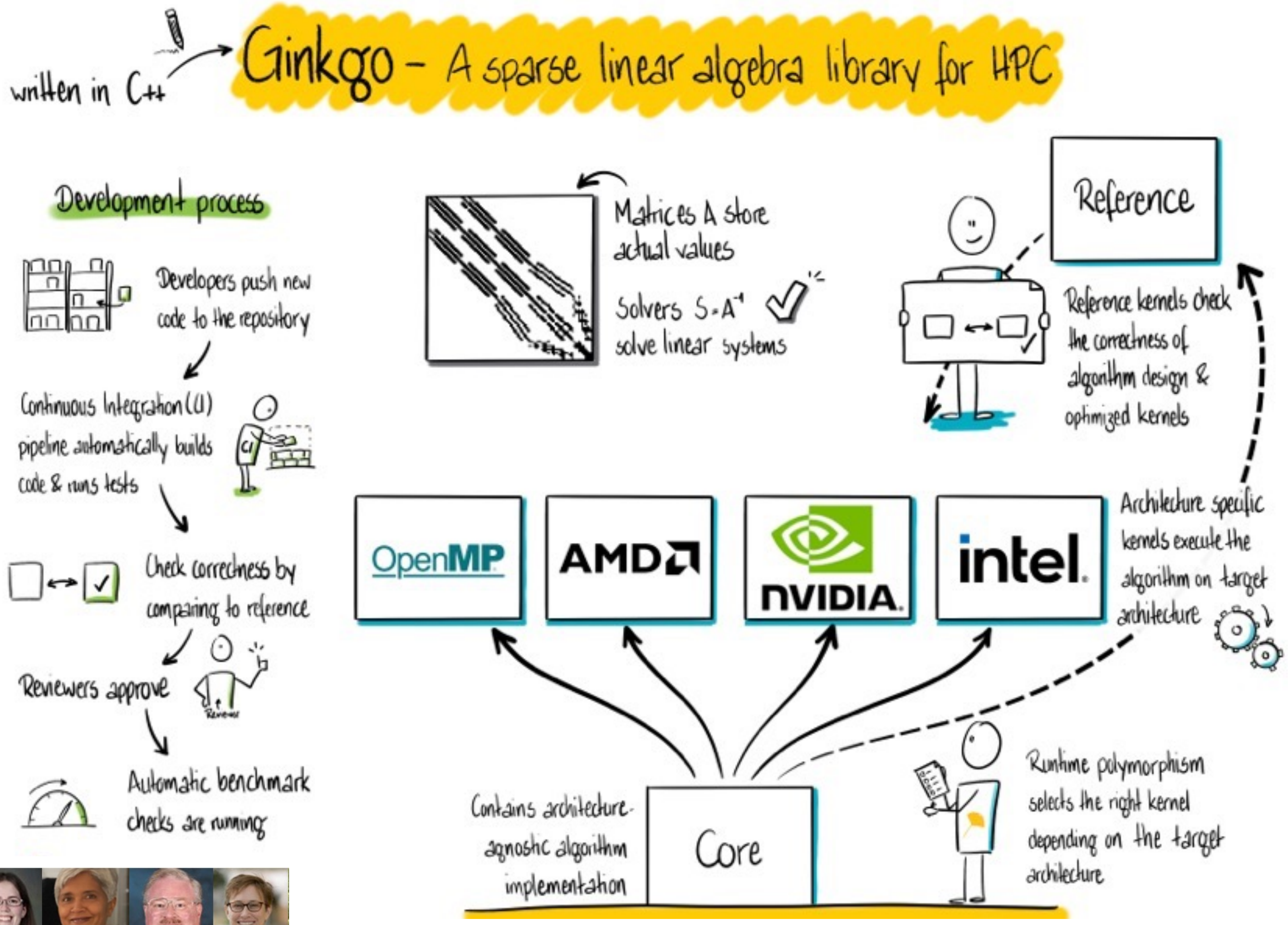


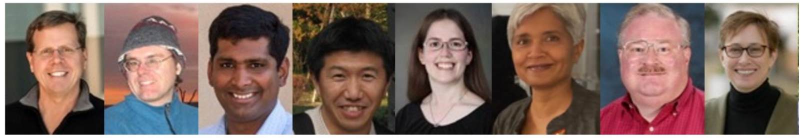


EXASCALE COMPUTING PROJECT

Designing an ECP library for sustaining simulation performance

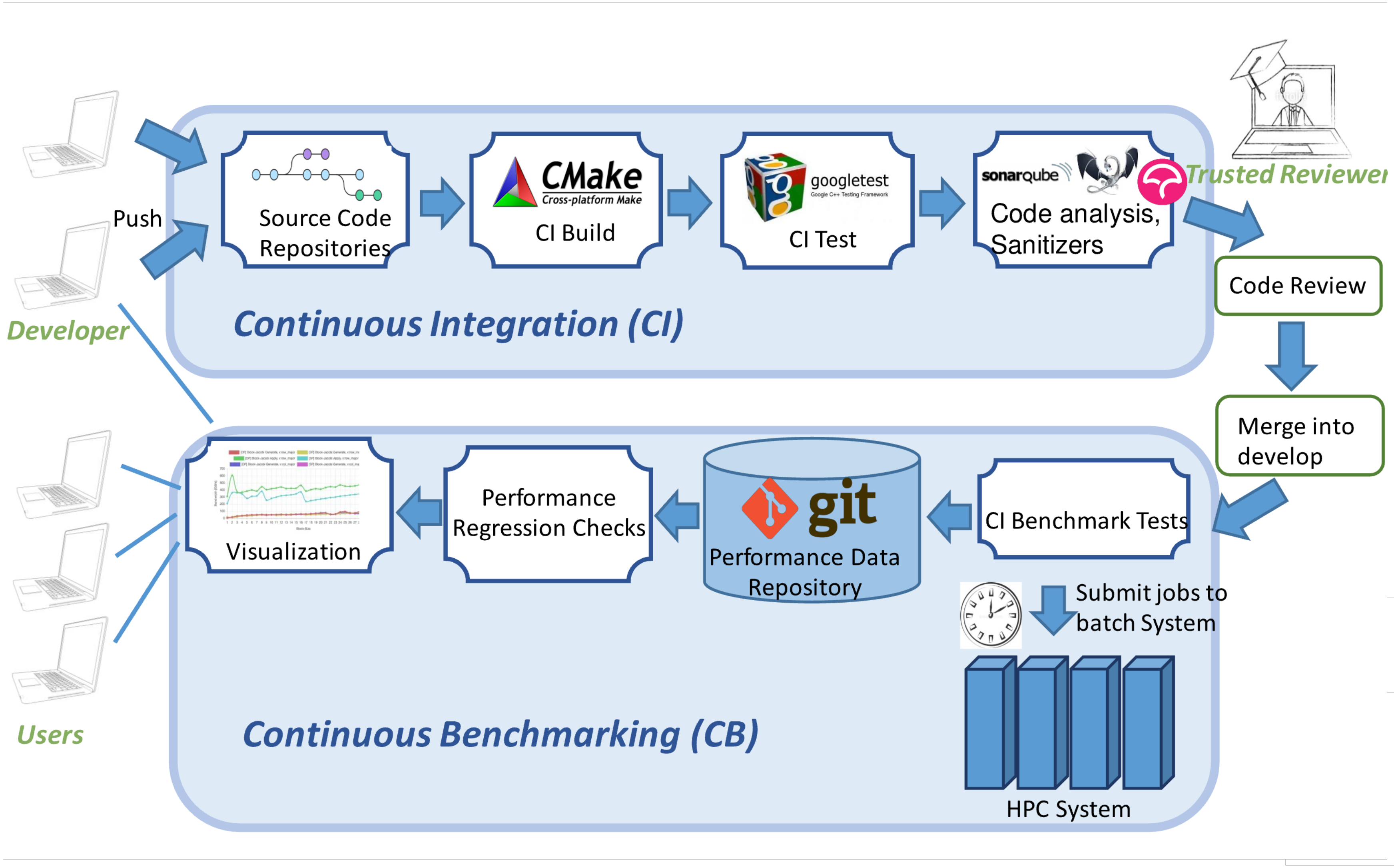






EXASCALE COMPUTING PROJECT

Sustainable software development & CI/CD



Starting with the CUDA backend

Library core contains architecture-agnostic factionality

CORE
Infrastructure
Algorithms
- Iterative Solvers
- Preconditioners
- …

Ginkgo

Runtime polymorphism selects the right kernel depending on the target architecture

REFERENCE | **OpenMP** | **CUDA**

NVIDIA.

Unit tests check correctness

CI/CD

googletest

	Functionality	OMP	CUDA
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	✓	✓

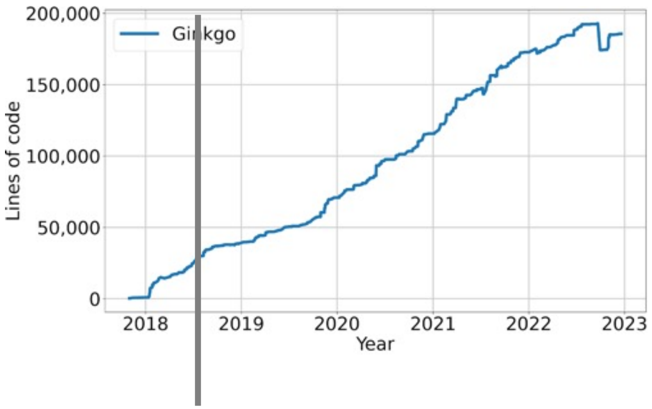
Linear Operator Interface

We express everything as Linear Operator.
- *Internally, we leverage C++ class inheritance.*
- *Applications can apply any functionality as a linear operator.*

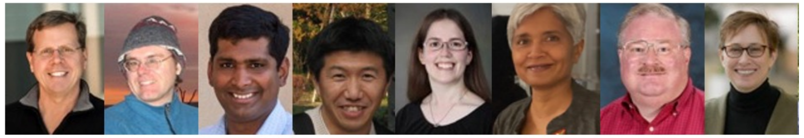
Matrix-Vector Product	Preconditioner (for matrix A)	Solver (for system $Ax = b$)
$x := A \cdot b$	$x := M^{-1} \cdot b$	$x := S \cdot b$
	$M^{-1} \approx A^{-1}$	$S \approx A^{-1}$
	$M^{-1} = \Pi(A)$	$S = \Sigma(A)$

All of them can be expressed as

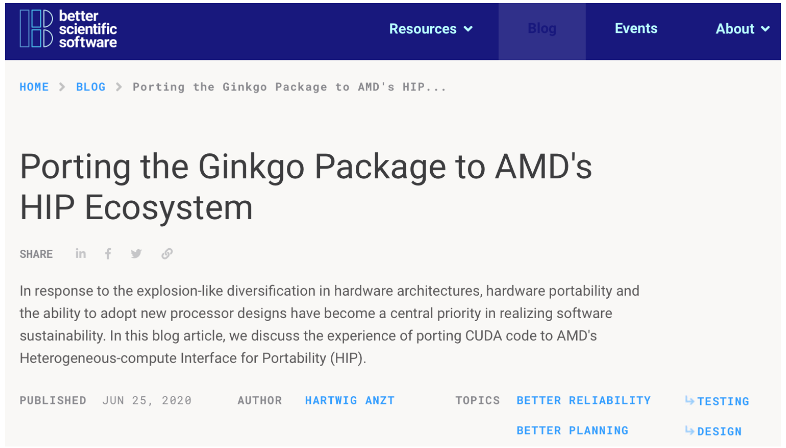
Application of a linear operator* (LinOp) $L : \mathbb{F}^m \to \mathbb{F}^m$



EXASCALE COMPUTING PROJECT



Extending to AMD GPUs


better scientific software | Resources | Blog | Events | About

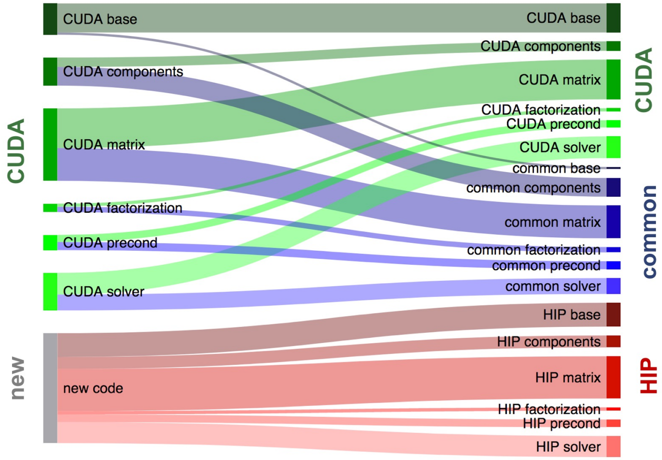
HOME > BLOG > Porting the Ginkgo Package to AMD's HIP...

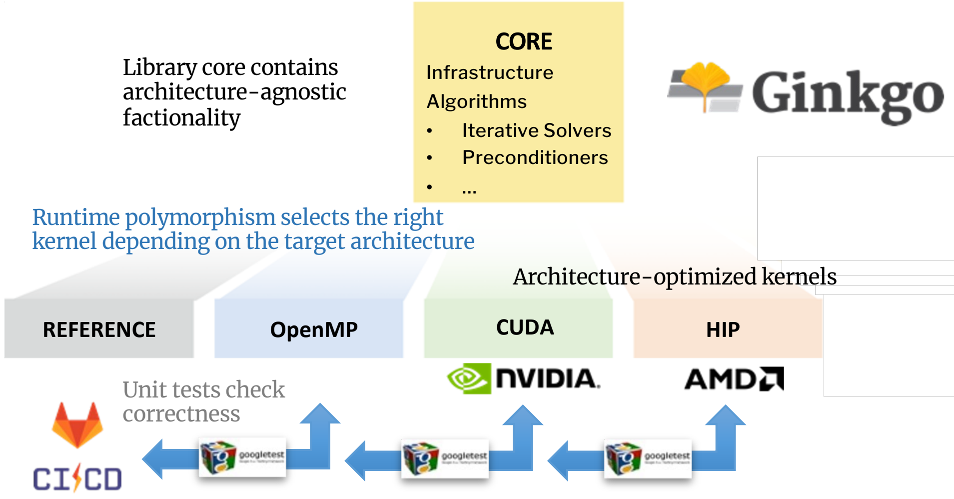
Porting the Ginkgo Package to AMD's HIP Ecosystem

SHARE

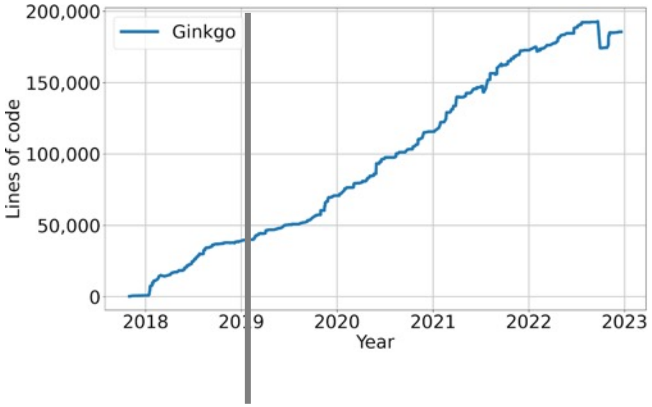
In response to the explosion-like diversification in hardware architectures, hardware portability and the ability to adopt new processor designs have become a central priority in realizing software sustainability. In this blog article, we discuss the experience of porting CUDA code to AMD's Heterogeneous-compute Interface for Portability (HIP).

PUBLISHED JUN 25, 2020 AUTHOR HARTWIG ANZT TOPICS BETTER RELIABILITY ↳TESTING BETTER PLANNING ↳DESIGN





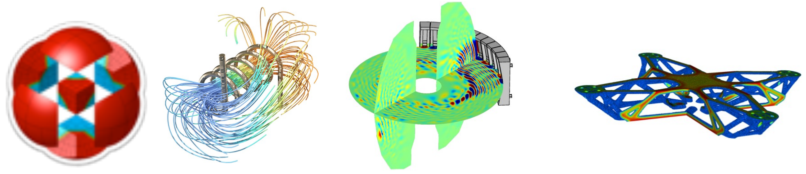
	Functionality	OMP	CUDA	HIP
Basic	SpMV	✓	✓	✓
	SpMM	✓	✓	✓
	SpGeMM	✓	✓	✓
Krylov solvers	BiCG	✓	✓	✓
	BiCGSTAB	✓	✓	✓
	CG	✓	✓	✓
	CGS	✓	✓	✓
	GMRES	✓	✓	✓
	IDR	✓	✓	✓
Preconditioners	(Block-)Jacobi	✓	✓	✓
	ILU/IC		✓	✓
	Parallel ILU/IC	✓	✓	✓
	Parallel ILUT/ICT	✓	✓	✓
	Sparse Approximate Inverse	✓	✓	✓



Spack xSDK E4S

EXASCALE COMPUTING PROJECT

Input from the “first customer”



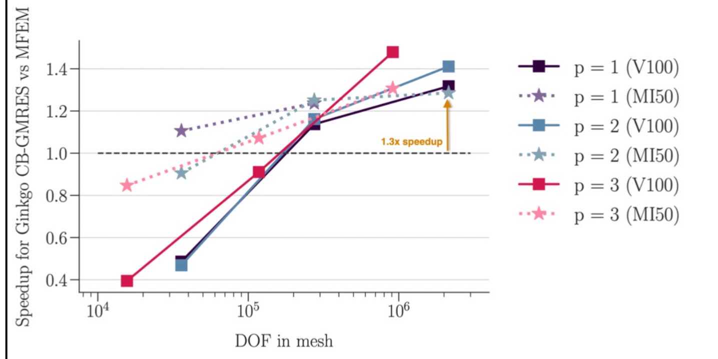
MFEM is a *free*, *lightweight*, *scalable* C++ library for finite element methods.

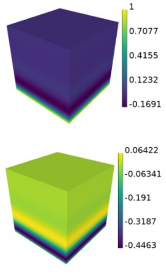
Speeding up MFEM’s “example 22” on GPUs

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:

$$-\nabla \cdot (\mathrm{a}\nabla u) - \omega^2 \mathrm{b}u + i\omega \mathrm{c}u = 0$$

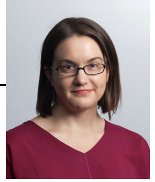
with a = 1, b = 1, ω = 10, c = 20

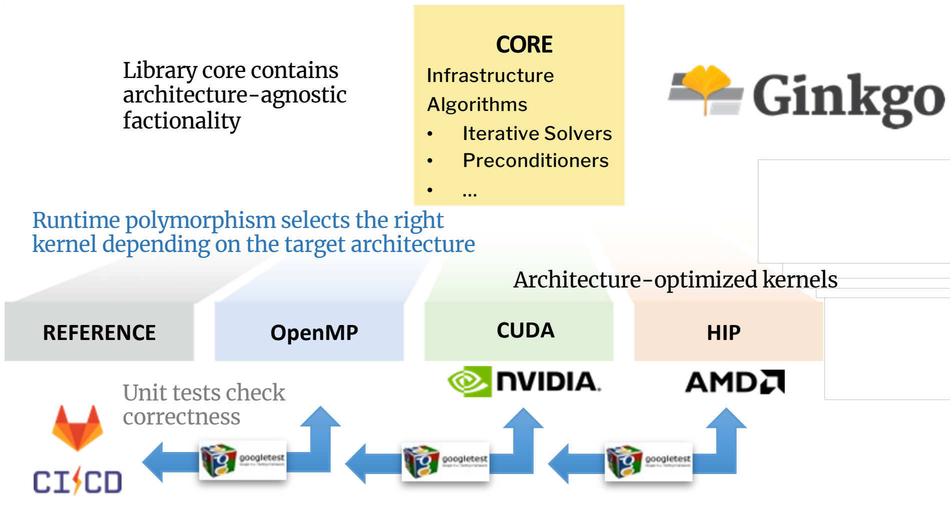


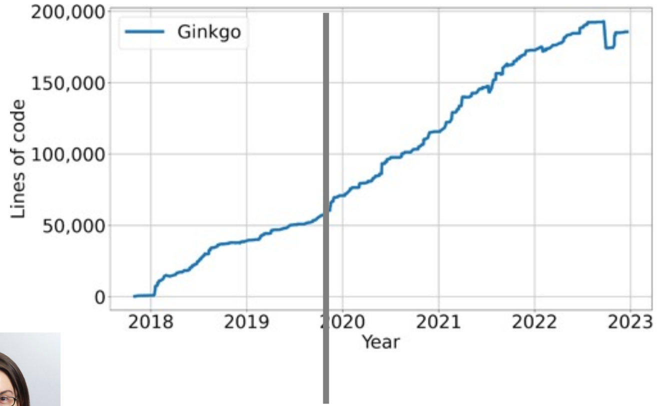


Real part of solution (top), imaginary part of solution

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.







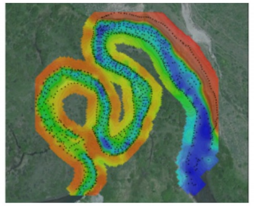
	Functionality	OMP	CUDA	HIP
Basic	SpMV	✓	✓	✓
	SpMM	✓	✓	✓
	SpGeMM	✓	✓	✓
Krylov solvers	BiCG	✓	✓	✓
	BiCGSTAB	✓	✓	✓
	CG	✓	✓	✓
	CGS	✓	✓	✓
	GMRES	✓	✓	✓
	IDR	✓	✓	✓
Preconditioners	(Block-)Jacobi	✓	✓	✓
	ILU/IC		✓	✓
	Parallel ILU/IC	✓	✓	✓
	Parallel ILUT/ICT	✓	✓	✓
	Sparse Approximate Inverse	✓	✓	✓
Utilities	On-Device Matrix Assembly	✓	✓	✓
	MC64/RCM reordering	✓		
	Wrapping user data		✓	

Spack xSDK E4S

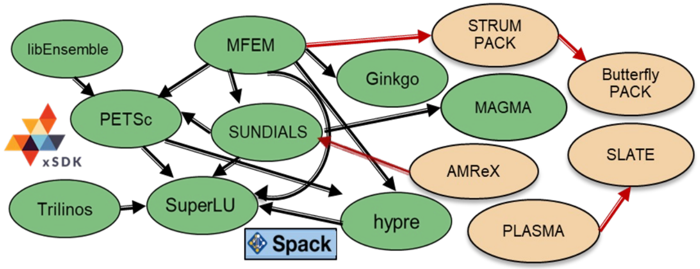
EXASCALE COMPUTING PROJECT

Part of the xSDK effort

xSDK: Extreme-scale Scientific Software Development Kit



Integrated surface-subsurface hydrology simulations of river meanders require the combined use of xSDK packages.

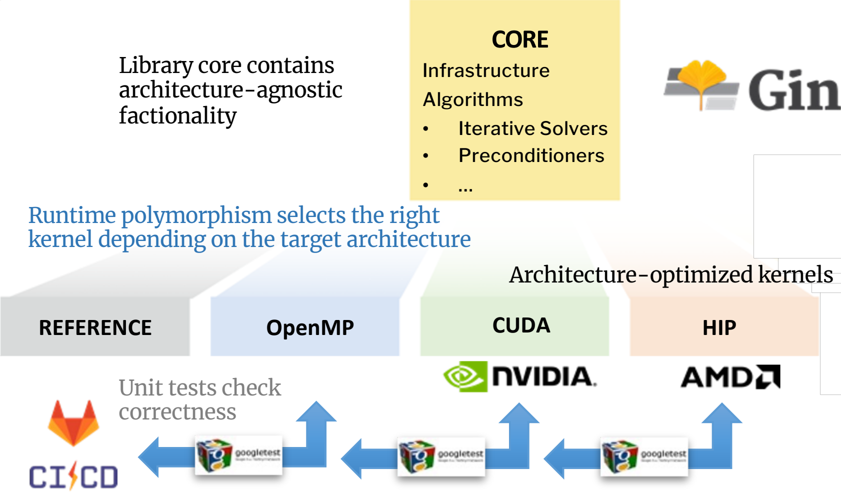


xsdk-examples v.0.3.0

The xSDK provides infrastructure for and interoperability of a **collection of related and complementary software elements**—developed by diverse, independent teams throughout the high-performance computing (HPC) community—that provide the building blocks, tools, models, processes, and related artifacts for rapid and efficient development of high-quality applications.

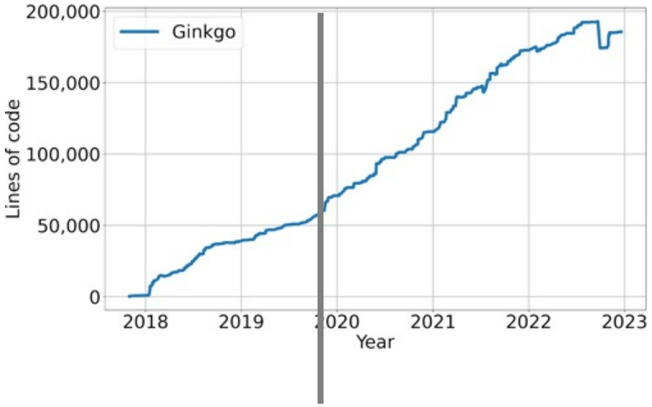
November 2022
- 26 math libraries
- 2 domain components
- 16 mandatory xSDK community policies
- Spack xSDK installer

xSDK community policies:
- 16 mandatory policies,
- 8 recommended policies,
- 4 Spack variant guidelines
- Available on Github https://xsdk.info/policies/



Ginkgo

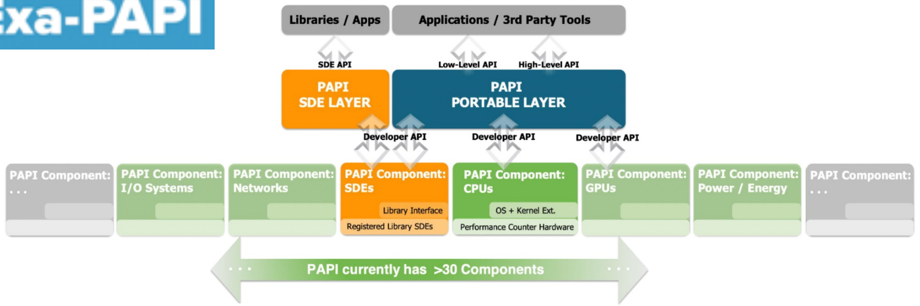
	Functionality	OMP	CUDA	HIP
Basic	SpMV	✓	✓	✓
	SpMM	✓	✓	✓
	SpGeMM	✓	✓	✓
Krylov solvers	BiCG	✓	✓	✓
	BiCGSTAB	✓	✓	✓
	CG	✓	✓	✓
	CGS	✓	✓	✓
	GMRES	✓	✓	✓
	IDR	✓	✓	✓
Preconditioners	(Block-)Jacobi	✓	✓	✓
	ILU/IC		✓	✓
	Parallel ILU/IC	✓	✓	✓
	Parallel ILUT/ICT	✓	✓	✓
	Sparse Approximate Inverse	✓	✓	✓
Utilities	On-Device Matrix Assembly	✓	✓	✓
	MC64/RCM reordering	✓		
	Wrapping user data	✓		

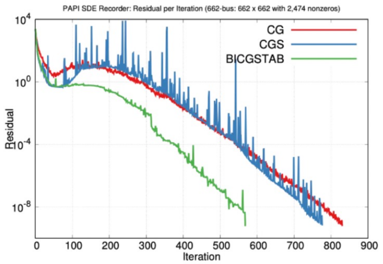


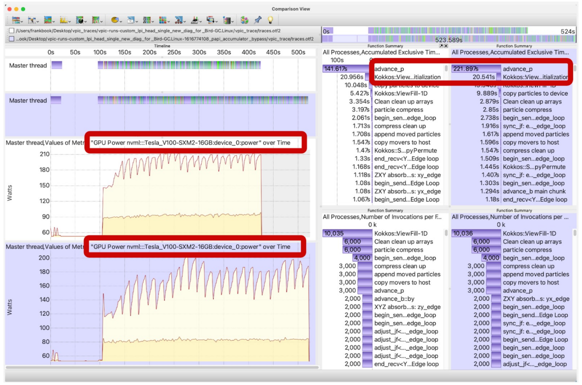
Spack xSDK E4S

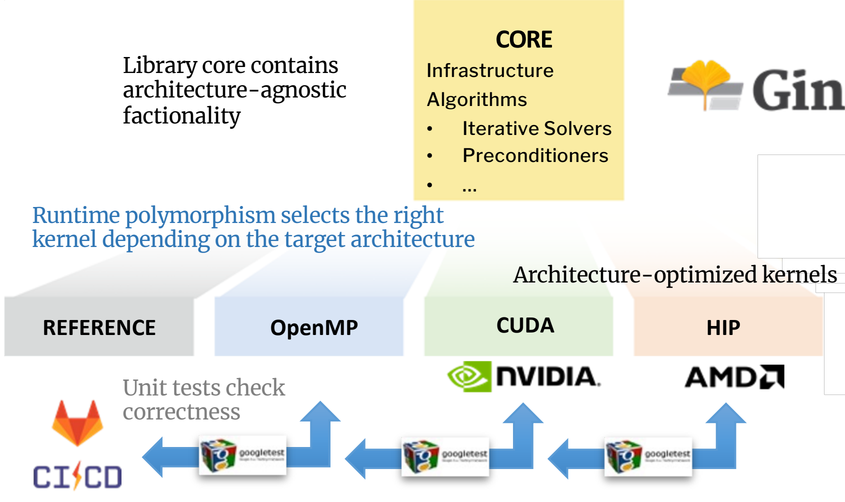
Adding profiling functionality

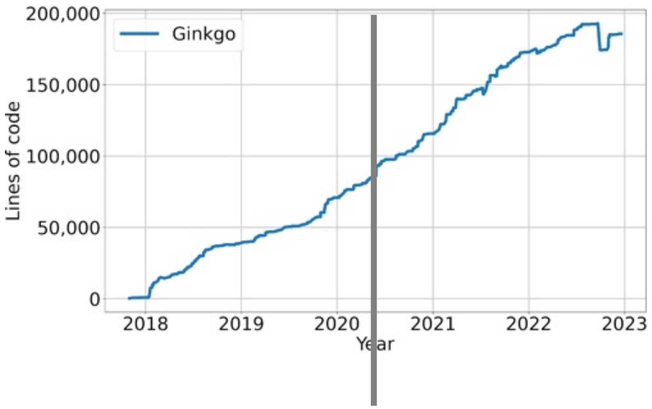
Exa-PAPI







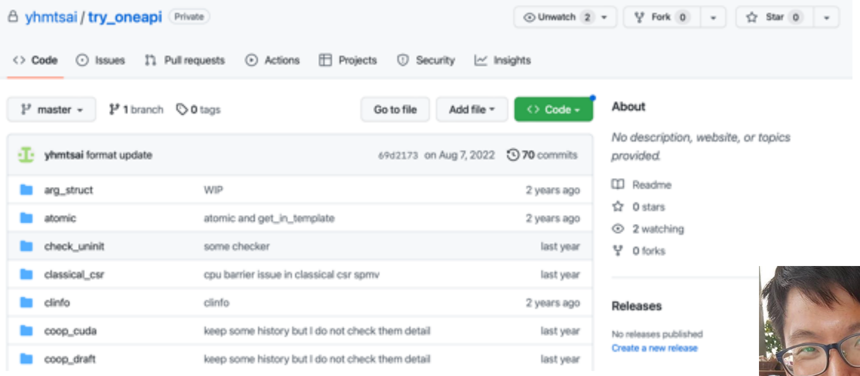


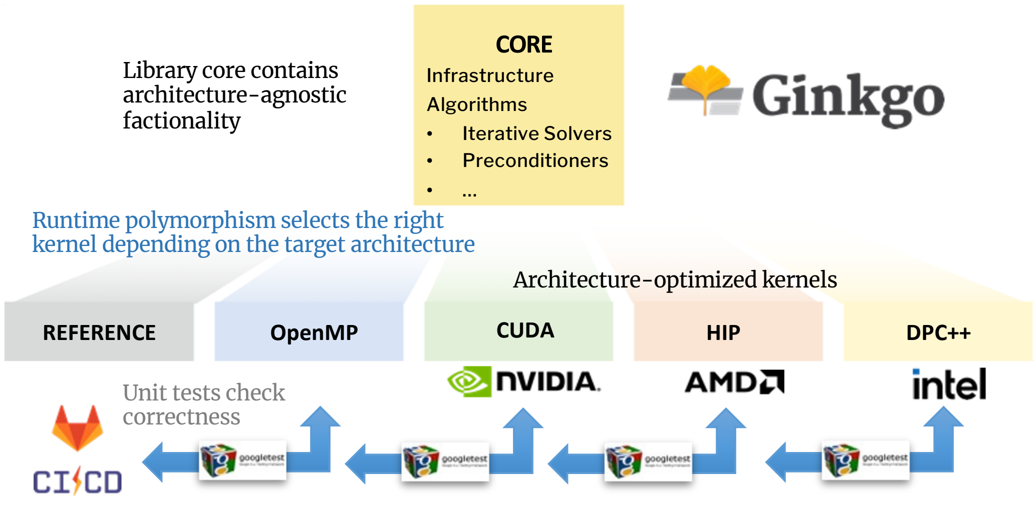


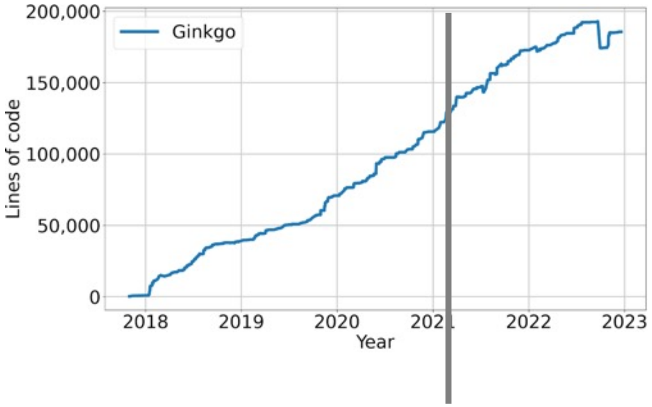
	Functionality	OMP	CUDA	HIP
Basic	SpMV	✓	✓	✓
	SpMM	✓	✓	✓
	SpGeMM	✓	✓	✓
Krylov solvers	BiCG	✓	✓	✓
	BiCGSTAB	✓	✓	✓
	CG	✓	✓	✓
	CGS	✓	✓	✓
	GMRES	✓	✓	✓
	IDR	✓	✓	✓
Preconditioners	(Block-)Jacobi	✓		✓
	ILU/IC		✓	✓
	Parallel ILU/IC	✓	✓	✓
	Parallel ILUT/ICT	✓	✓	✓
	Sparse Approximate Inverse	✓	✓	✓
Utilities	On-Device Matrix Assembly	✓	✓	✓
	MC64/RCM reordering	✓		
	Wrapping user data		✓	
	Logging		✓	
	PAPI counters		✓	

EXASCALE COMPUTING PROJECT

Extending to Intel GPUs







	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	✓	✓	✓	✓
Utilities	On-Device Matrix Assembly	✓	✓	✓	✓
	MC64/RCM reordering	✓			
	Wrapping user data	✓	✓	✓	✓
	Logging	✓	✓	✓	✓
	PAPI counters	✓	✓	✓	✓

Spack xSDK E4S Exa-PAPI

oneAPI

Industry Collaboration with bi-weekly meetings

EXASCALE COMPUTING PROJECT

Extending to Intel GPUs

Ginkgo

- Bi-Weekly technical meetings with Intel
- Long list of bug reports, feature requests, performance data discussions, documentation improvements …

… but also docker image contributions and bug fixes!

cuBLAS backend (and potentially other domains) fails with latest LLVM builds #223

New issue

Closed mmeterel opened this issue 22 days ago · 3 comments

mmeterel commented 22 days ago • edited — Contributor

Summary

As first observed in #219 many tests in cuBLAS backend is failing with latest LLVM builds.

Version

I have tried LLVM commit: 66361038b63caaae566fc9648f5da50b74222b83 and got the below tests failing (showing only a few of them)

```
 1 - BLAS/RT/Nrm2TestSuite/Nrm2Tests.RealSinglePrecision/Column_Major_TITAN_RTX (Failed)
 3 - BLAS/RT/Nrm2TestSuite/Nrm2Tests.RealDoublePrecision/Column_Major_TITAN_RTX (Failed)
 7 - BLAS/RT/Nrm2TestSuite/Nrm2Tests.ComplexDoublePrecision/Column_Major_TITAN_RTX (Failed)
17 - BLAS/RT/IamaxTestSuite/IamaxTests.RealSinglePrecision/Column_Major_TITAN_RTX (Failed)
19 - BLAS/RT/IamaxTestSuite/IamaxTests.RealDoublePrecision/Column_Major_TITAN_RTX (Failed)
23 - BLAS/RT/IamaxTestSuite/IamaxTests.ComplexDoublePrecision/Column_Major_TITAN_RTX (Failed)
27 - BLAS/RT/DotuTestSuite/DotuTests.ComplexDoublePrecision/Column_Major_TITAN_RTX (Failed)
35 - BLAS/RT/DotcTestSuite/DotcTests.ComplexSinglePrecision/Column_Major_TITAN_RTX (Failed)
67 - BLAS/RT/AsumTestSuite/AsumTests.ComplexSinglePrecision/Column_Major_TITAN_RTX (Failed)
81 - BLAS/RT/ScalTestSuite/ScalTests.RealSinglePrecision/Column_Major_TITAN_RTX (Failed)
85 - BLAS/RT/ScalTestSuite/ScalTests.ComplexSinglePrecision/Column_Major_TITAN_RTX (Failed)
```

Assignees: mmeterel
Labels: None yet
Projects: None yet
Milestone: No milestone
Development: No branches or pull requests
2 participants

tid % subgroup size >= 4 gives wrong division

(double) 1/a gives wrong result when the tid % subgroup siz
For example, when a = 1.07338829563753890
1/a should be 0.9316293125835232
if (local_id == assign_id) { a = double(1)/a; }
when assign_id < 4, Gen9 GPU still give the correct result
when assign_id >= 4, Gen9 GPU gives wrong 0.931629359
1.0000000506
CPU has more worse result

It is connected to optimizations (not reproducible with O0). F
fp-specuation=off do not improve results.
Ticket number: XDEPS-4031 ()

ginkgohub/oneapi:cuda11.6
DIGEST: sha256:0bc4c10d79a75b183ac1deafcda753365c6e1a94edc3046a9a0eb8ba2d7b9d94

OS/ARCH	COMPRESSED SIZE	LAST PUSHED
linux/amd64	6.63 GB	22 days ago by yhmtsai

IMAGE LAYERS

```
1  ADD file ... in /
2  CMD ["bash"]
3  ENV NVARCH=x86_64
```

Devcloud node issue

- sycl-ls/clinfo does not give any ou
  s001-n225, s011-n006
- no gpu on the nodes
  s001-n232, s001-n233, s011-n008
- github.com is not accessable on login-

Fix cuda/hip backend location #219

New issue

Merged mkrainiuk merged 2 commits into oneapi-src:develop from yhmtsai:fix_cuda_backend_location 20 days ago

Conversation 8 · Commits 2 · Checks 0 · Files changed 16 · +76 −0

yhmtsai commented on Aug 1 • edited — Contributor

Description

From intel/llvm#6407, it moves almost all headers from CL/sycl to sycl
I followed #199 way
make the header can use sycl/* if they exist and allow the old intel llvm.
I also update the CL/sycl.hpp which are not changed before.

All Submissions

- [ ] Do all unit tests pass locally? Attach a log. A: It is a compiling issue.
- [x] Have you formatted the code using clang-format?

Bug fixes

- [ ] Have you added relevant regression tests? A: It is a compiling issue.
- [x] Have you included information on how to reproduce the issue (either in a GitHub issue or in this PR)?

Reproduce:
compile the latest intel llvm and this repo, it will not be able to compile due to missing headers.

yhmtsai added 2 commits 2 months ago

- use the correct sycl path after intel/llvm#6407 — 8edef4a
- fix the missing sycl/sycl.hpp — 03a2f6f

mmeterel commented on Aug 1 — Contributor

@yhmtsai Thanks for the PR. Is the description from sycl/CL to CL correct? My understanding is all header files moved

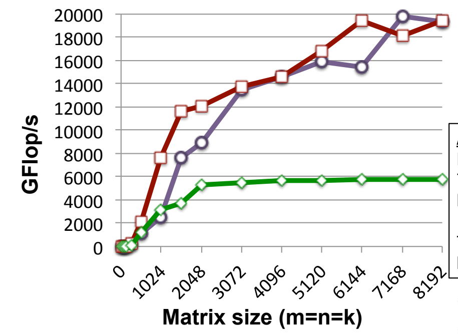
Reviewers: mkrainiuk, mmeterel
Assignees: No one assigned
Labels: None yet
Projects: None yet
Milestone: No milestone
Development: Successfully merging this pull request may close these issues. None yet
3 participants

From DPCPP AoT documentation, not clear:

- The options are also required at linking time? Unused in files without kernels?
- Any example of other projects integrating AoT in a CMake setup?

Intel Compiler (Fortran/C/C++/L0) - Intel Discrete GPU Accelerator - Joint Laboratory for System Evaluation (anl.gov)
hang_atomic_on_local
Ticket number: CMPLRLLVM-36572 (works in PVC, but still fails on ATS node)
related to driver not compiler self

Performance of DPC++ MAGMA SGEMM on Intel GPUs

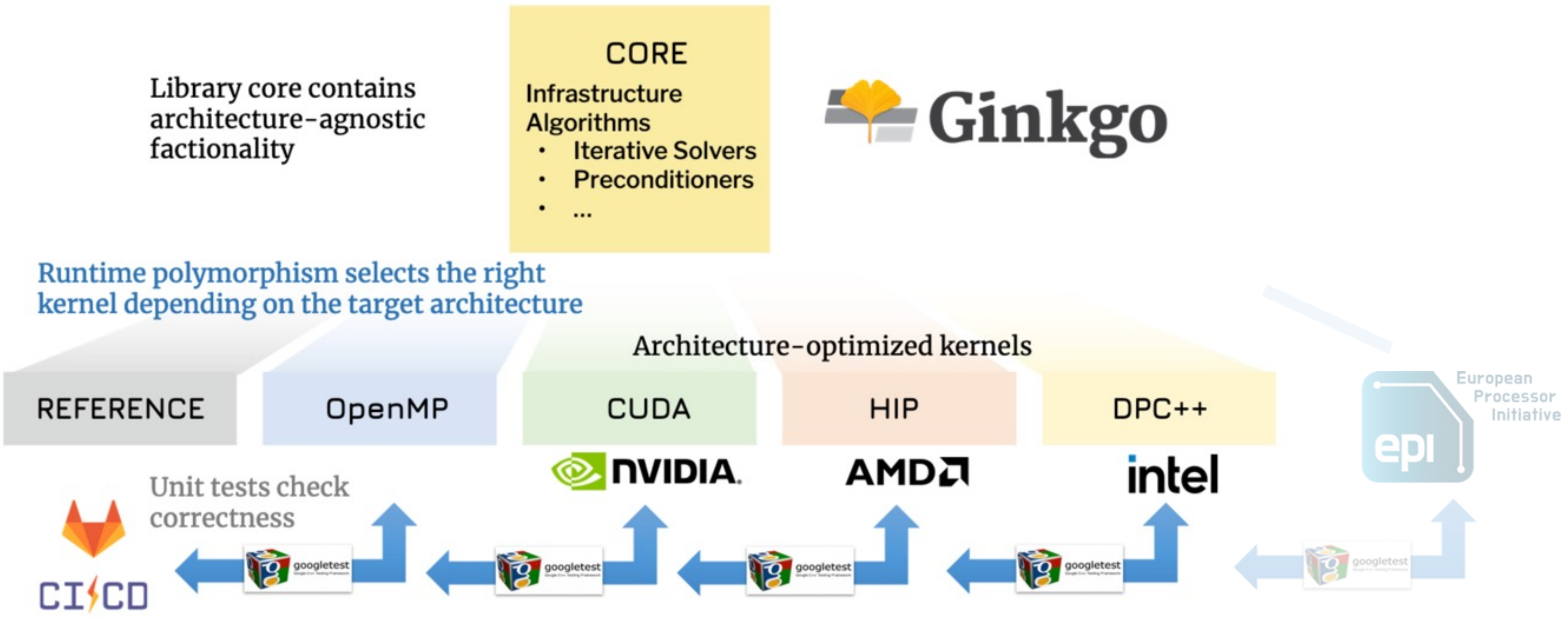


Arcticus at ALCF
Intel Xe-HP GPU (Arctic Sound 2x)
7,680 x2 Cores @900 MHz
FP32 peak 13,820 x2 GFlop/s

Transitional system to Aurora
Exascale supercomputer at ANL

ECP EXASCALE COMPUTING PROJECT

Portability as central design principle



Focus efforts as lightweight tool in ECP to address challenges

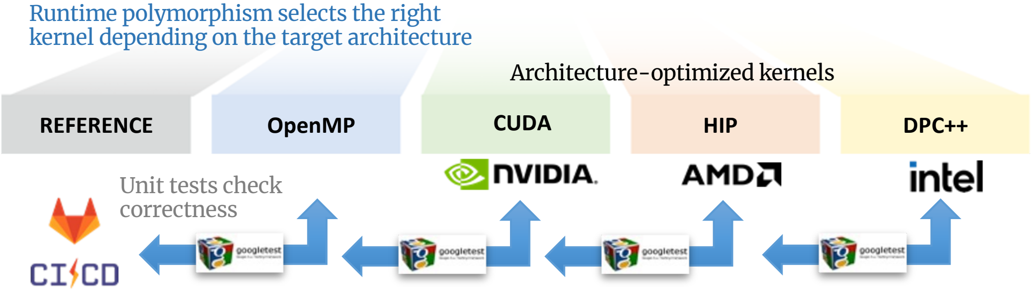
Focus efforts
- Mixed precision
- batched


- *Address recent hardware trends (tensor cores, etc.)*
- *Address hardware requirements*

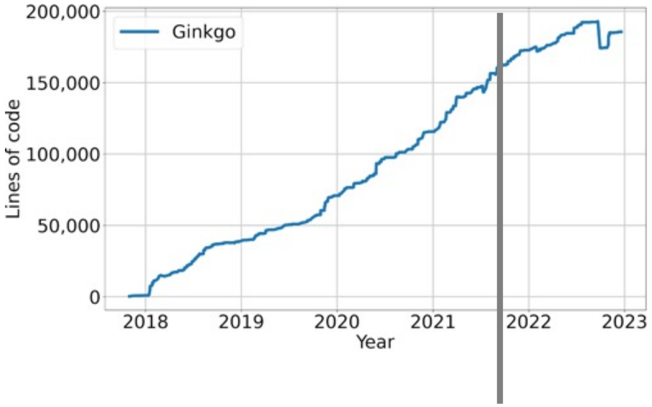
Library core contains architecture-agnostic factionality

CORE
Infrastructure
Algorithms
- Iterative Solvers
- Preconditioners
- …

Ginkgo



	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	✓	✓	✓	✓
Utilities	On-Device Matrix Assembly	✓	✓	✓	✓
	MC64/RCM reordering	✓			
	Wrapping user data	✓ (all)			
	Logging	✓ (all)			
	PAPI counters	✓ (all)			



Spack xSDK E4S Exa-PAPI oneAPI

Industry Collaboration with bi-weekly meetings

EXASCALE COMPUTING PROJECT

Mixed precision focus effort

Focus efforts
- Mixed precision
- batched


- *Address recent hardware trends (tensor cores, etc.)*
- *Address hardware requirements*

Advances in Mixed Precision Algorithms: 2021 Edition

by the ECP Multiprecision Effort Team (Lead: Hartwig Anzt)

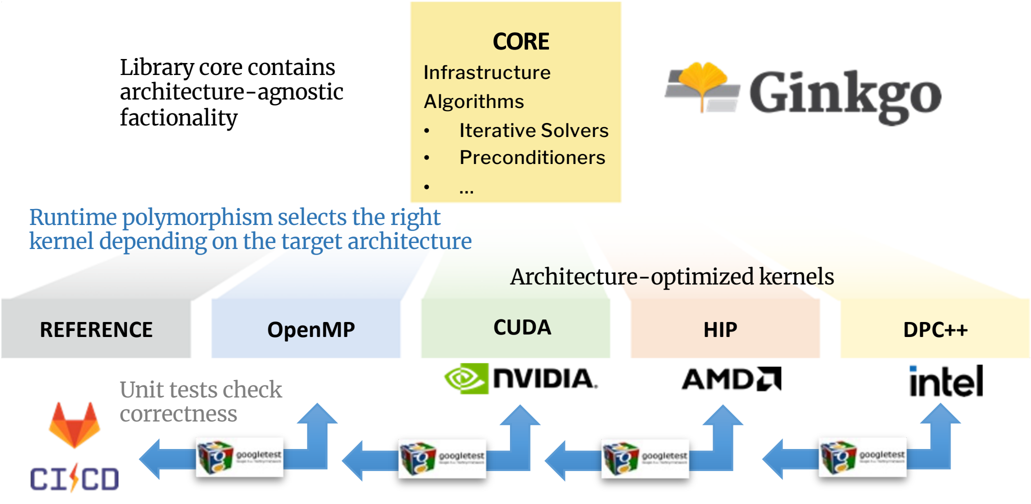
Ahmad Abdelfattah, Hartwig Anzt, Alan Ayala, Erik G. Boman, Erin Carson, Sebastien Cayrols, Terry Cojean, Jack Dongarra, Rob Falgout, Mark Gates, Thomas Grützmacher, Nicholas J. Higham, Scott E. Kruger, Sherry Li, Neil Lindquist, Yang Liu, Jennifer Loe, Piotr Luszczek, Pratik Nayak, Daniel Osei-Kuffuor, Sri Pranesh, Sivasankaran Rajamanickam, Tobias Ribizel, Barry Smith, Kasia Swirydowicz, Stephen Thomas, Stanimire Tomov, Yaohung M. Tsai, Ichi Yamazaki, Urike Meier Yang

Available access | Research article | First published online March 19, 2021

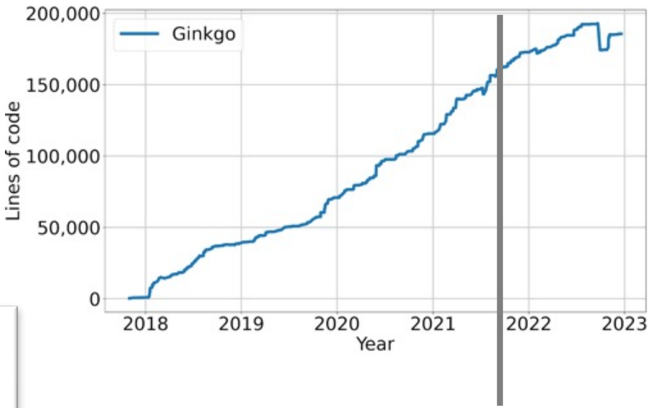
A survey of numerical linear algebra methods utilizing mixed-precision arithmetic

Ahmad Abdelfattah, Hartwig Anzt, [...], and Ulrike Meier Yang +18 View all authors and affiliations

Volume 35, Issue 4 | https://doi.org/10.1177/10943420211003313



	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	✓	✓	✓	✓
Utilities	On-Device Matrix Assembly	✓	✓	✓	✓
	MC64/RCM reordering	✓			
	Wrapping user data	✓ (all)			
	Logging	✓ (all)			
	PAPI counters	✓ (all)			



Industry Collaboration with bi-weekly meetings

EXASCALE COMPUTING PROJECT

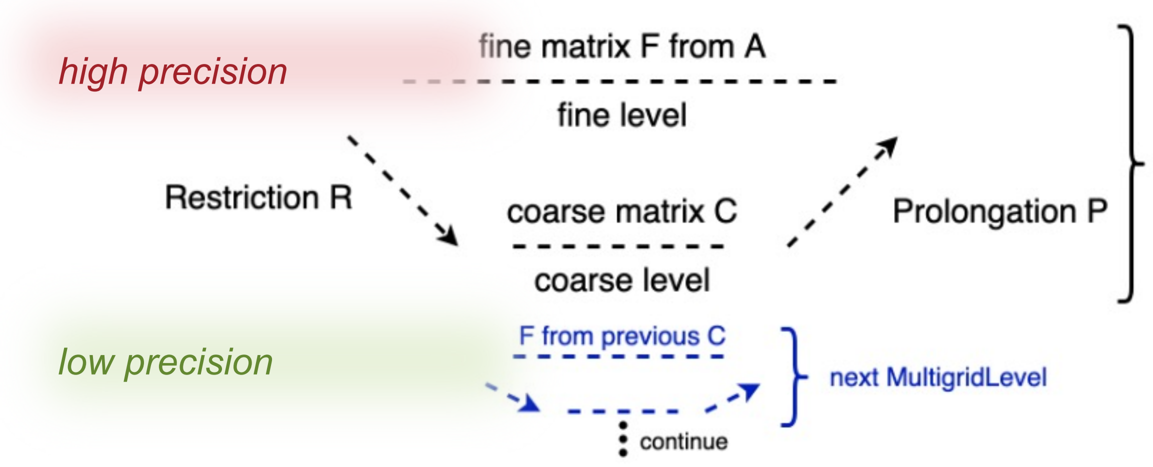
Mixed precision AMG on GPUs

- **Preconditioning iterative solvers**
  - Idea: Approximate inverse of system matrix to make the system "easier to solve": $P^{-1} \approx A^{-1}$

    and solve $Ax = b \quad \Leftrightarrow \quad P^{-1}Ax = P^{-1}b \quad \Leftrightarrow \quad \tilde{A}x = \tilde{b}$

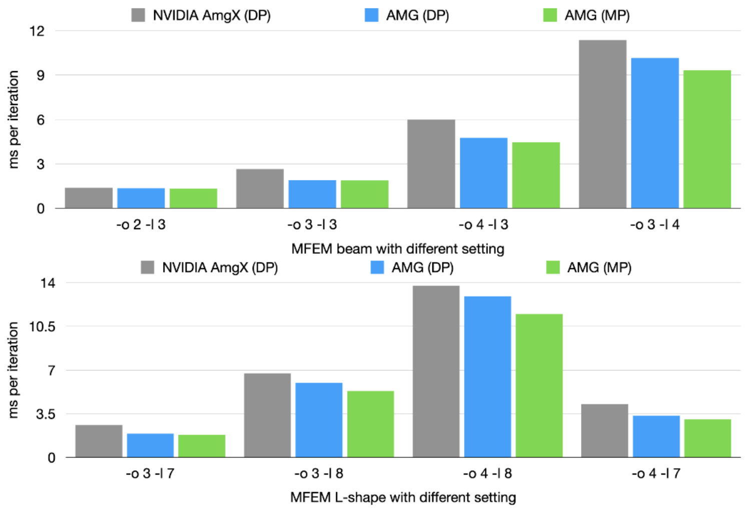
- **Mixed Precision Multigrid Preconditioner**

```
multigrid::build()
    .with_max_levels(10u) // equal to NVIDIA/AMGX 11 max levels
    .with_min_coarse_row(64u)
    .with_pre_smoother(sm, sm_f)
    .with_mg_level(pgm, pgm_f)
    .with_level_selector(
        [](const size_type level, const LinOp*) -> size_type {
            // Only the first level is generated by MultigridLevel(double).
            // The subsequent levels are generated by MultigridLevel(float)
            return level >= 1 ? 1 : 0;
        })
    .with_coarest_solver(coarest_solver_f)
```



Ginkgo

Mike Tsai



EXASCALE COMPUTING PROJECT

Stephen F. McCormick, Joseph Benzaken, Rasmus Tamstorf: Algebraic error analysis for mixed-precision multigrid solvers, https://arxiv.org/abs/2007.06614

Mixed precision AMG on GPUs

Focus efforts
- Mixed precision
- batched

ICL UTK @ICL_UTK · Sep 13 ···

Congratulations to Yu-Hsiang Mike Tsai from @KITKarlsruhe, in collaboration with ICL's Natalie Beams and @HartwigAnzt! Their paper "Mixed Precision Algebraic Multigrid on GPUs" took home a best paper award at PPAM2022.
ppam.edu.pl

Library core contains architecture-agnostic factionality

CORE
Infrastructure
Algorithms
- Iterative Solvers
- Preconditioners
- …

Ginkgo

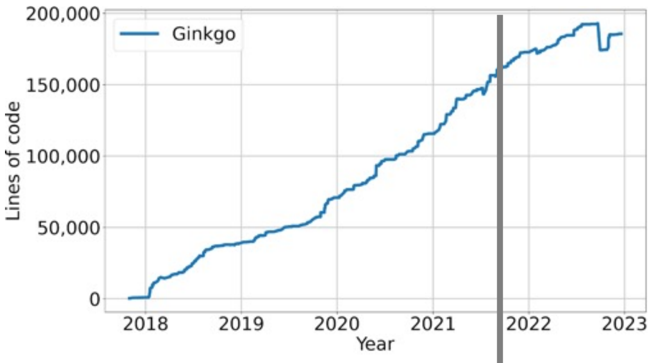
Runtime polymorphism selects the right kernel depending on the target architecture

Architecture-optimized kernels

REFERENCE	OpenMP	CUDA	HIP	DPC++
		NVIDIA	AMD	intel

Unit tests check correctness

CI/CD



Spack · xSDK · E4S · Exa-PAPI · oneAPI

Industry Collaboration with bi-weekly meetings

	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	✓	✓	✓	✓
AMG	AMG preconditioner	✓	✓	✓	✓
	AMG solver	✓	✓	✓	✓
	Parallel Graph Match	✓	✓	✓	✓
Utilities	On-Device Matrix Assembly	✓	✓	✓	✓
	MC64/RCM reordering	✓			
	Wrapping user data	✓			
	Logging	✓			
	PAPI counters	✓			

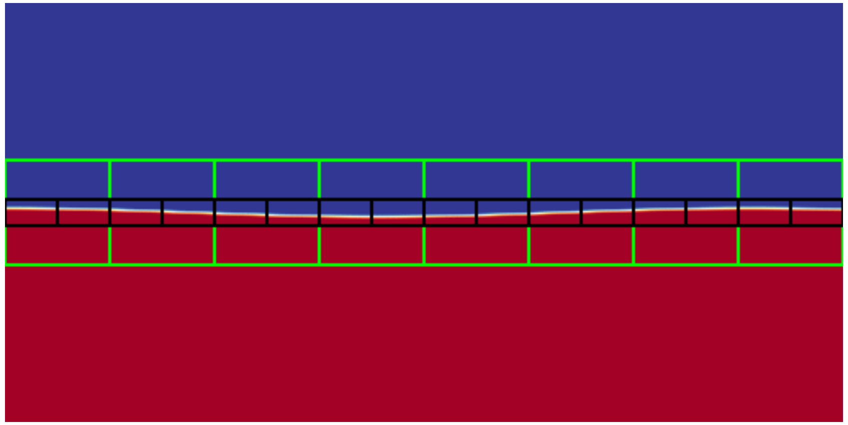
EXASCALE COMPUTING PROJECT

Batched focus effort – Combustion Simulations

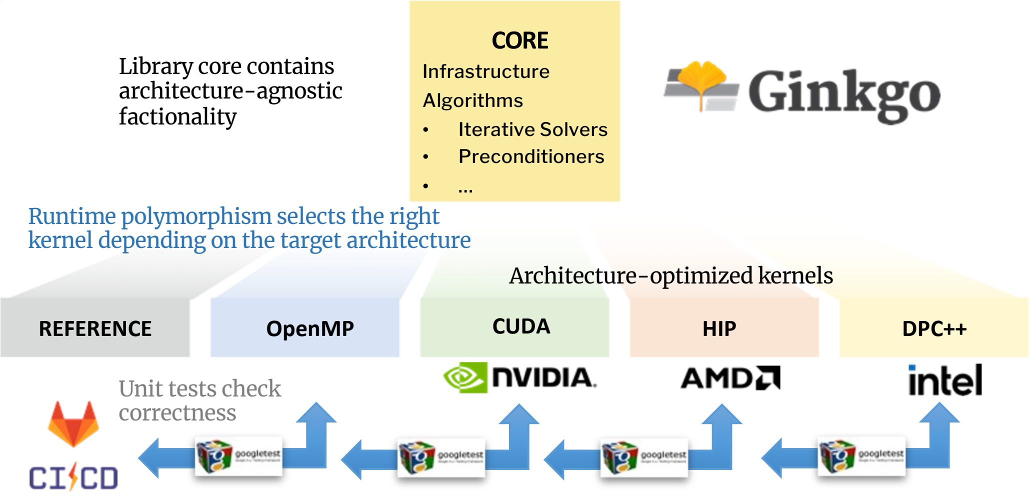
Batched iterative solvers for SUNDIALS / PeleLM

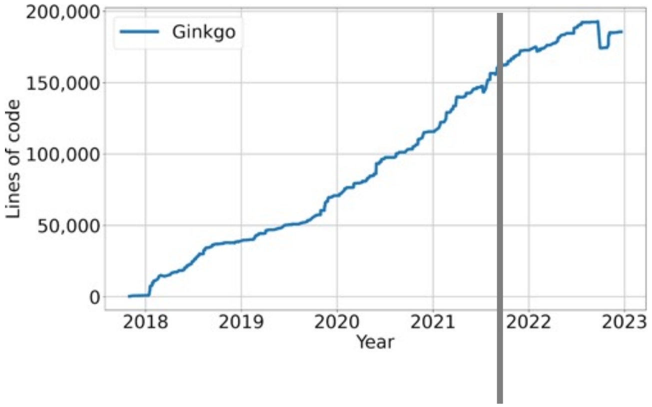
PeleLM is a parallel, adaptive mesh refinement (AMR) code that solves the reacting Navier-Stokes equations in the low Mach number regime. The core libraries for managing the subcycling AMR grids and communication are found in the AMReX source code.

https://amrex-combustion.github.io/PeleLM/overview.html



Problem	Size	Non-zeros (A)	Non-zeros (L+U)
dodecane_lu	54	2,332 (80%)	2,754 (94%)
drm19	22	438 (90%)	442 (91%)
gri12	33	978 (90%)	1,018 (93%)
gri30	54	2,560 (88%)	2,860 (98%)
isooctane	144	6,135 (30%)	20,307 (98%)
lidryer	10	91 (91%)	91 (91%)





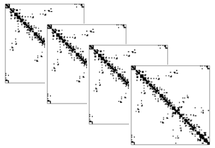
Spack · xSDK · E4S · Exa-PAPI · oneAPI

Industry Collaboration with bi-weekly meetings

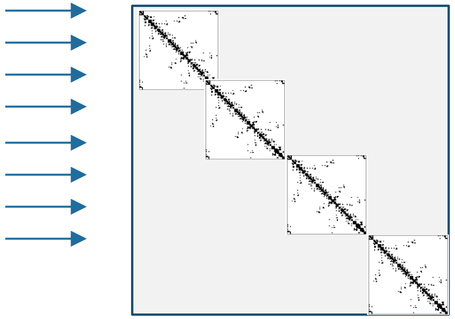
	Functionality	OMP	CUDA	HIP	DPC++
Basic	SpMV	✓	✓	✓	✓
	SpMM	✓	✓	✓	✓
	SpGeMM	✓	✓	✓	✓
Krylov solvers	BiCG	✓	✓	✓	✓
	BiCGSTAB	✓	✓	✓	✓
	CG	✓	✓	✓	✓
	CGS	✓	✓	✓	✓
	GMRES	✓	✓	✓	✓
	IDR	✓	✓	✓	✓
Preconditioners	(Block-)Jacobi	✓	✓	✓	✓
	ILU/IC		✓	✓	✓
	Parallel ILU/IC	✓	✓	✓	✓
	Parallel ILUT/ICT	✓	✓	✓	✓
	Sparse Approximate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

EXASCALE COMPUTING PROJECT

Batched focus effort – Combustion Simulations

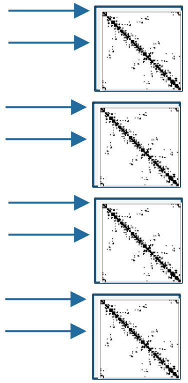


- Many sparse problems of medium size have to be solved concurrently.
  - ~ 50 – 2,000 unknowns, < 50% dense;
  - All sparse systems may share the same sparsity pattern;
  - An approximate solution may be acceptable (e.g., inside a non-linear solver);



- One solution is to arrange the individual systems on the main diagonal of one large system.
  - Convergence determined by the "hardest" problem;
  - No reuse of sparsity pattern information;
  - Global synchronization points;

- Better approach: design batched iterative solve functionality that solves all problems concurrently.
  - Problem-dependent convergence accounted for;
  - No global synchronization;
  - Reuse of sparsity pattern information;

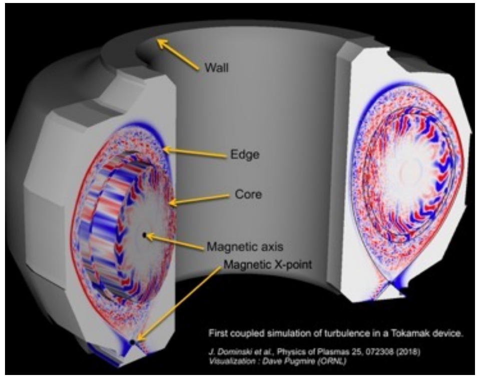


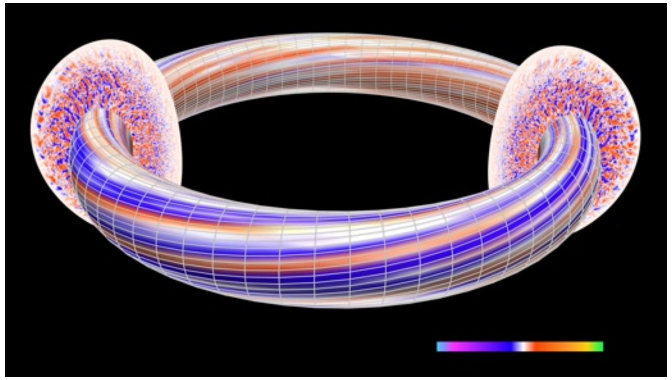
EXASCALE COMPUTING PROJECT

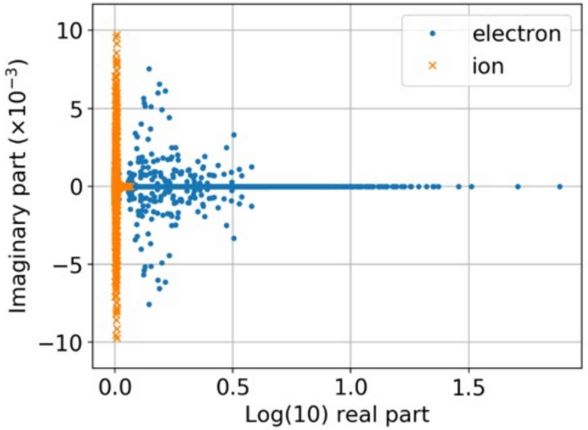
Batched focus effort – Fusion Plasma Simulations

XGC is a gyrokinetic particle-in-cell code, which specializes in the simulation of the edge region of magnetically confined thermonuclear fusion plasma. The simulation domain can include the magnetic separatrix, magnetic axis and the biased material wall. XGC can run in total-delta-f, and conventional delta-f mode. The ion species are always gyrokinetic except for ETG simulation. Electrons can be adiabatic, massless fluid, driftkinetic, or gyrokinetic.

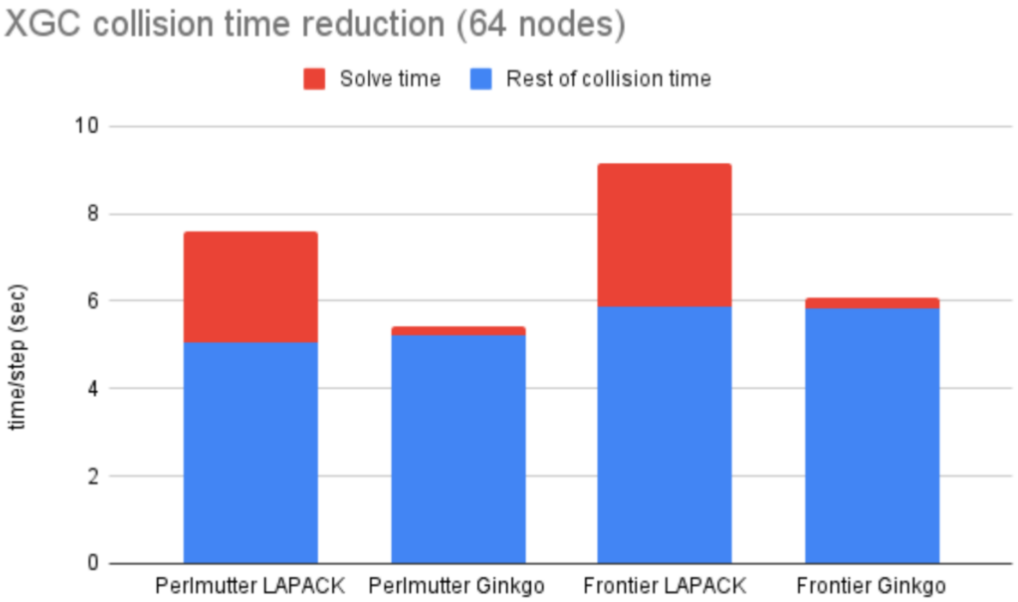
Source: https://xgc.pppl.gov/html/general_info.html





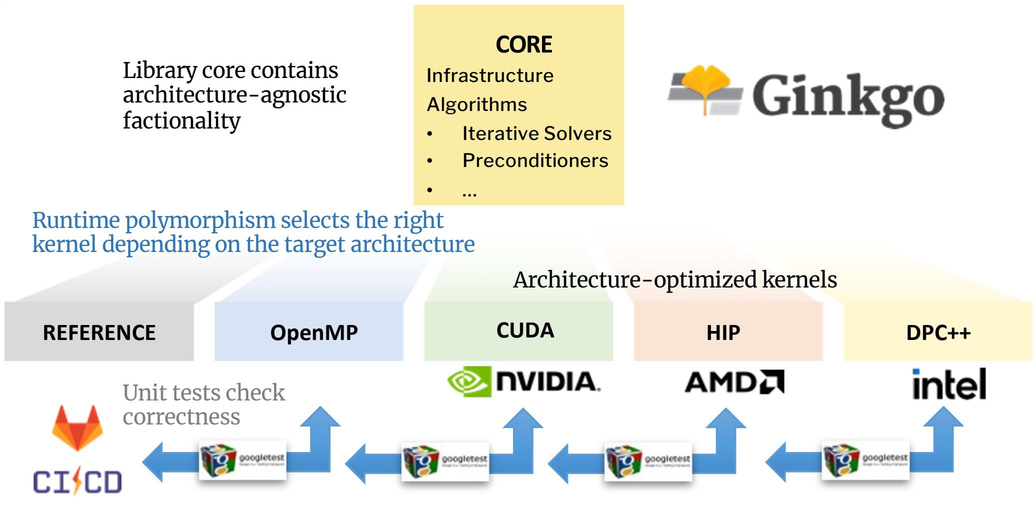


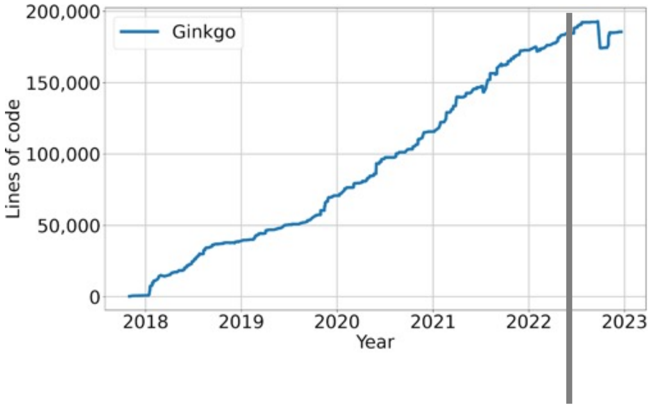
- Two species
- Ions easy to solve
- Electrons hard to solve
- Banded matrix structure
- Non-symmetric, need BiCGSTAB
- n = ~1,000
- nz = ~9,000



EXASCALE COMPUTING PROJECT

Adding Batched Functionality



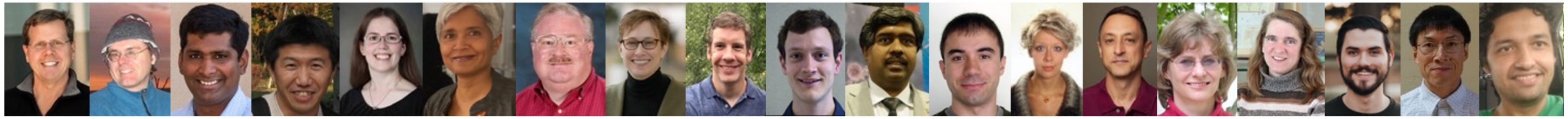


Spack xSDK E4S Exa-PAPI oneAPI sundials XGC

Industry Collaboration with bi-weekly meetings

	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	✓	✓	✓	✓
Utilities	On-Device Matrix Assembly	✓	✓	✓	✓
	MC64/RCM reordering	✓			
	Wrapping user data	✓			
	Logging	✓			
	PAPI counters	✓			

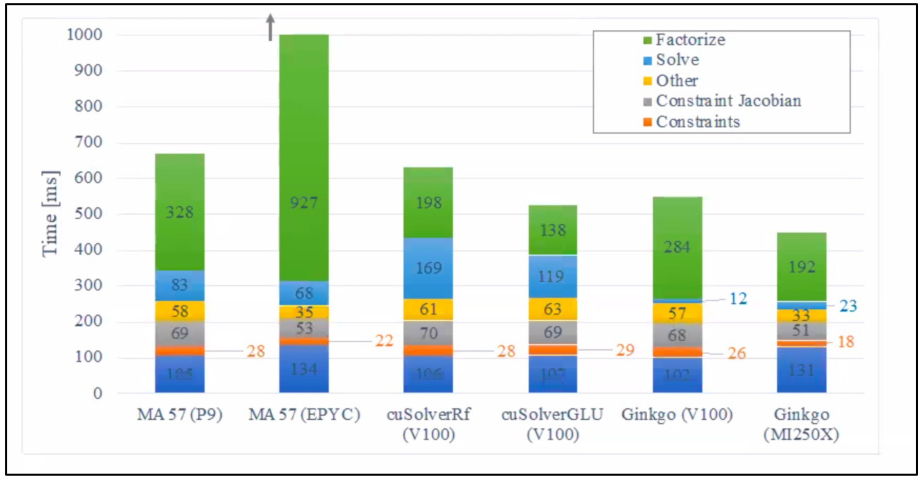
EXASCALE COMPUTING PROJECT



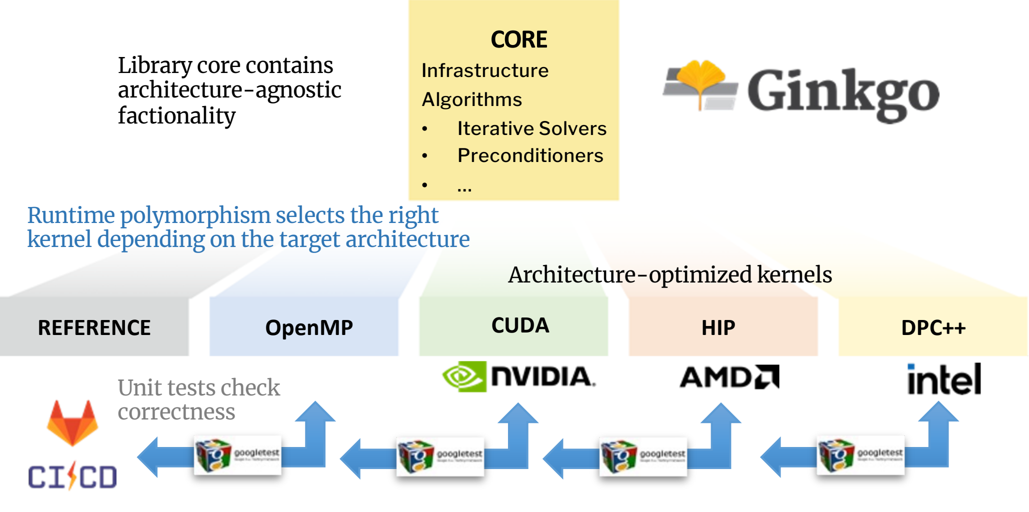
Sparse direct solvers for power grid simulations

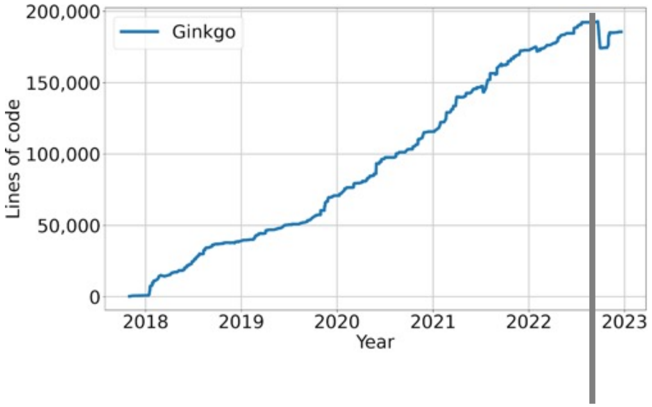
EXASGD

© Slaven Peles



- Power Grid Simulations
- All GPU solvers outperform CPU sovers
- Ginkgo first GPU-resident solver





Industry Collaboration with bi-weekly meetings

EXASGD

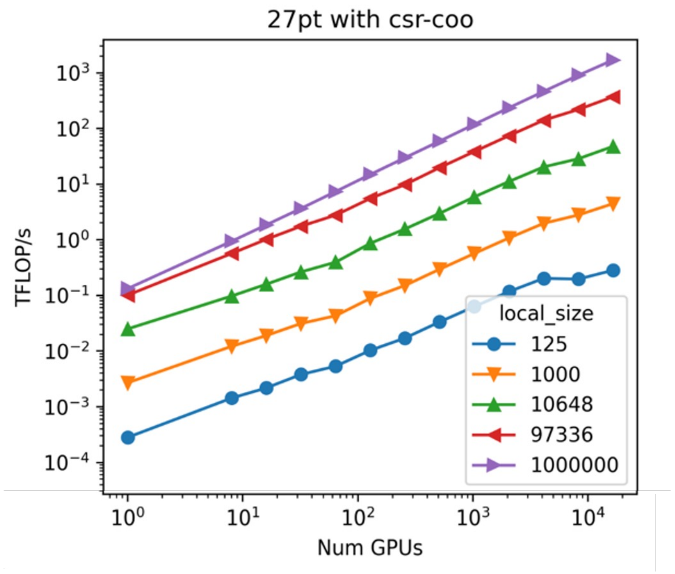
	Functionality	OMP	CUDA	HIP	DPC++
Basic	SpMV	✓	✓	✓	✓
	SpMM	✓	✓	✓	✓
	SpGeMM	✓	✓	✓	✓
Krylov solvers	BiCG	✓	✓	✓	✓
	BiCGSTAB	✓	✓	✓	✓
	CG	✓	✓	✓	✓
	CGS	✓	✓	✓	✓
	GMRES	✓	✓	✓	✓
	IDR	✓	✓	✓	✓
Preconditioners	(Block-)Jacobi	✓	✓	✓	✓
	ILU/IC		✓	✓	✓
	Parallel ILU/IC	✓	✓	✓	✓
	Parallel ILUT/ICT	✓	✓	✓	✓
	Sparse Approximate 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	✓			
	Logging	✓			
	PAPI counters	✓			

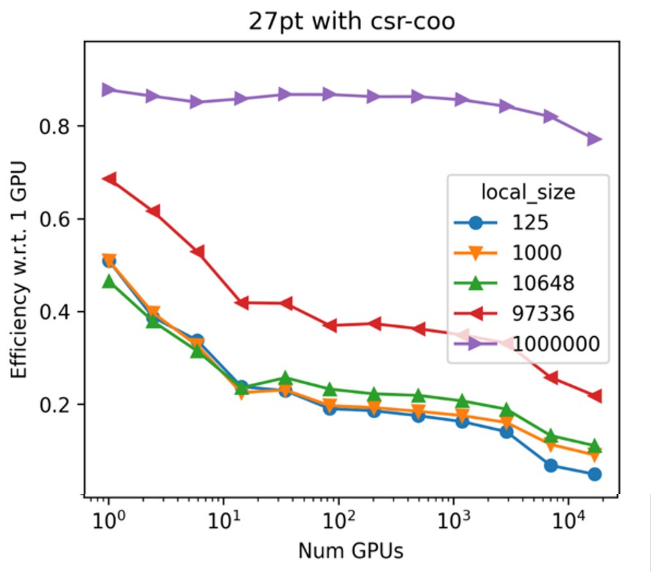
EXASCALE COMPUTING PROJECT

Distributed runs on Frontier (Cray + AMD MI250 GPUs)

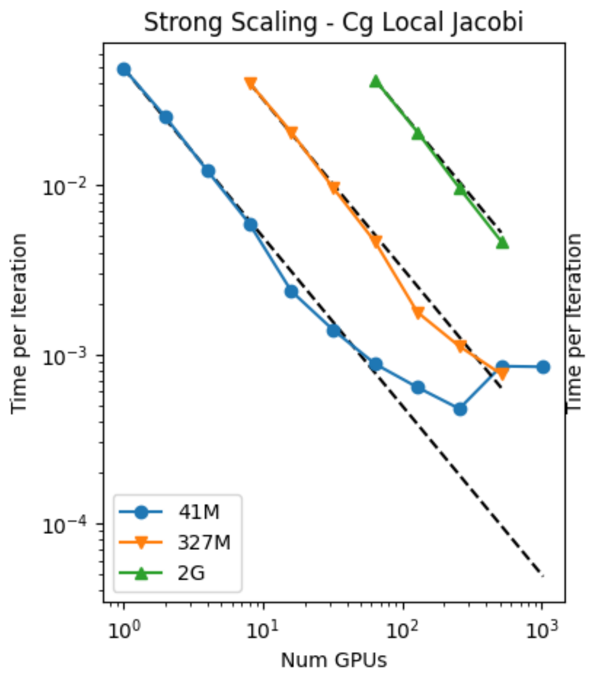
Weak scaling: problem size increases with parallel resources

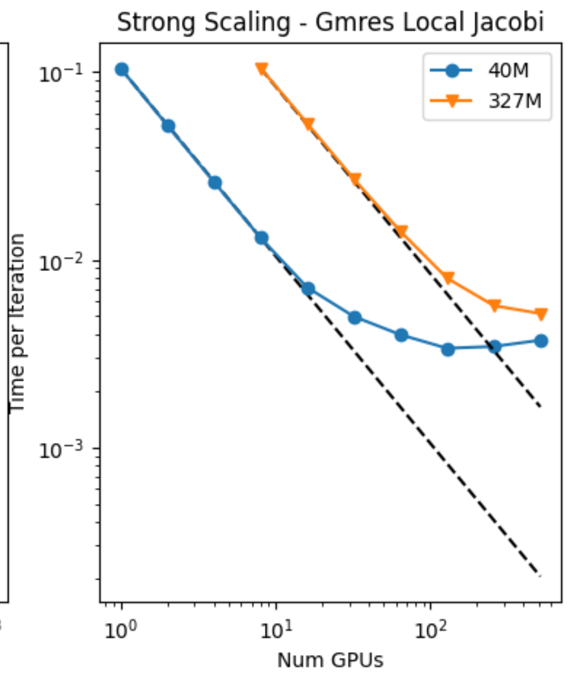
Weak scaling up to 16k GCDs (8k GPUs)





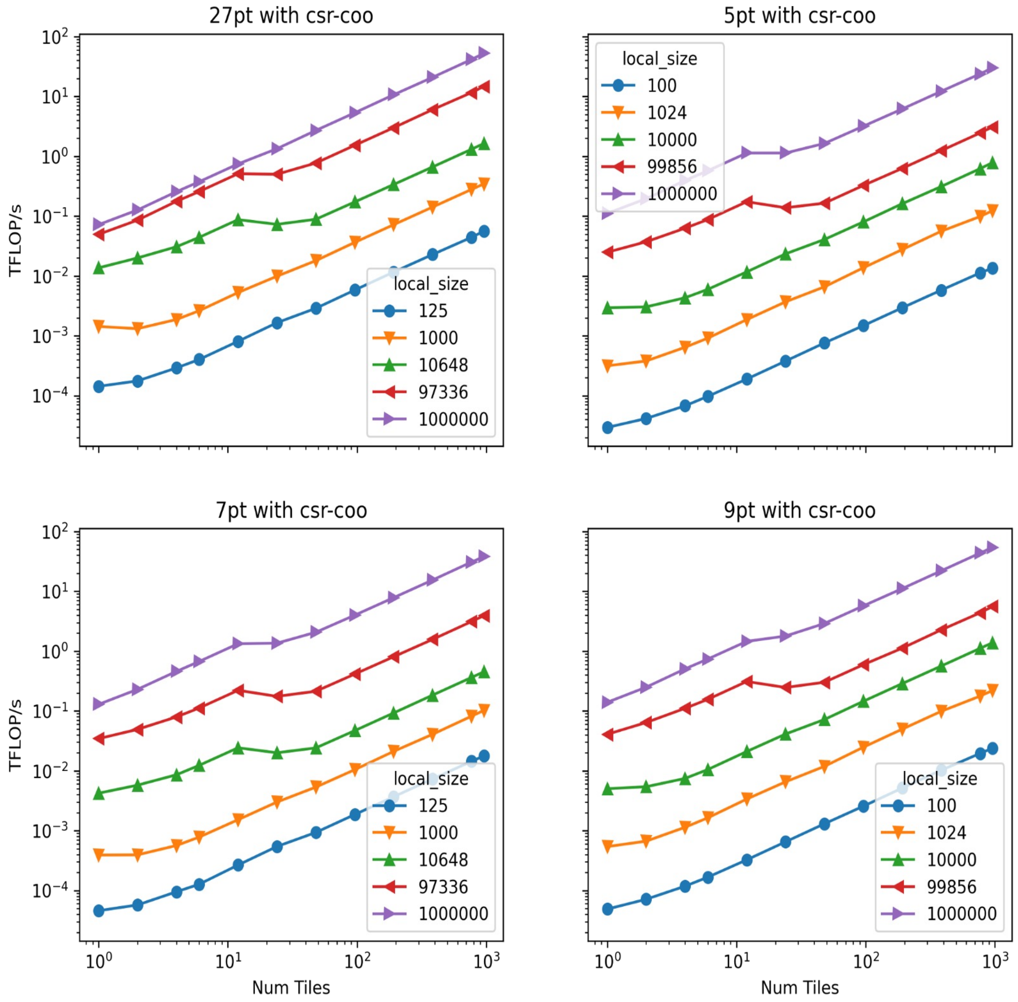
Strong scaling: problem size constant

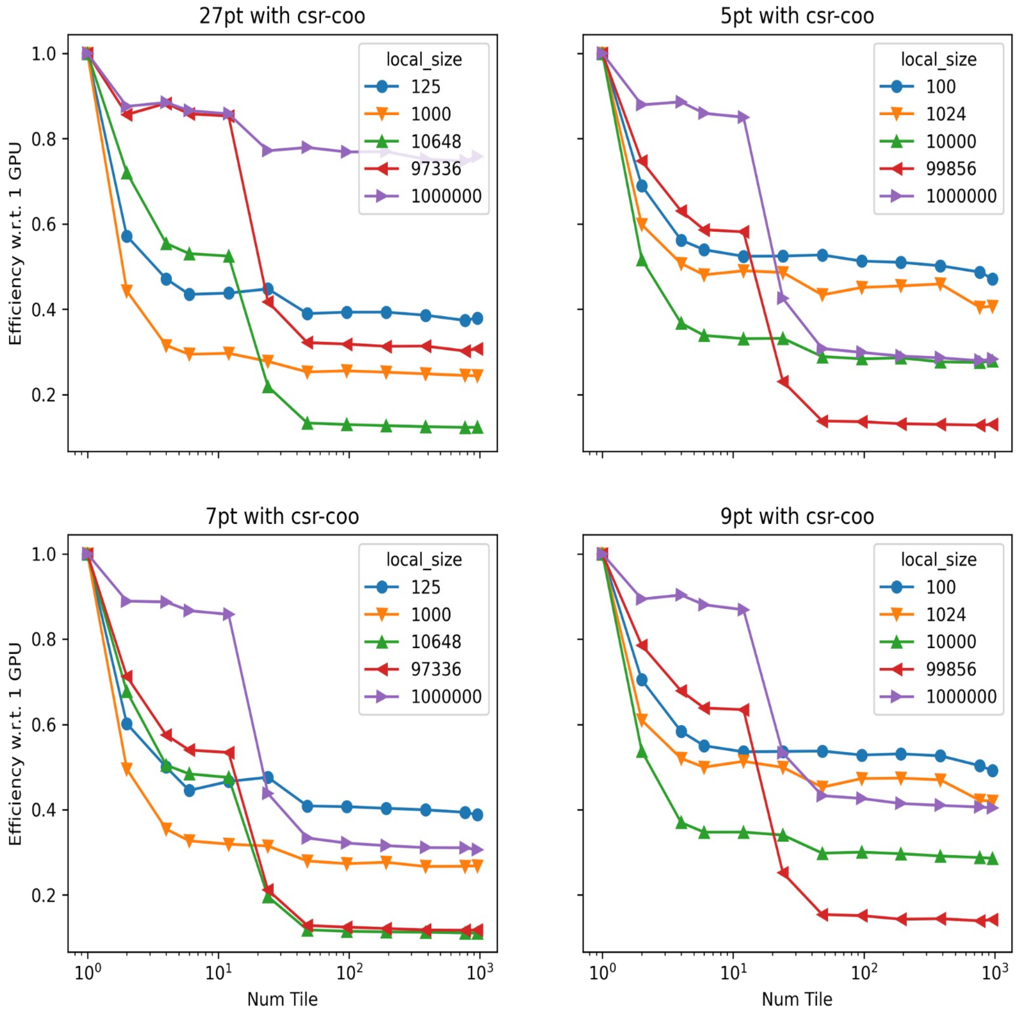




Distributed runs on Sunspot (Intel PVCA GPUs)

Weak scaling: problem size increases with parallel resources

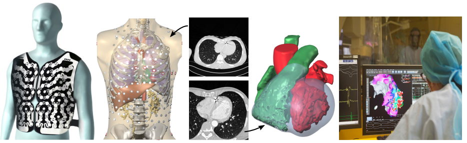




“Now” – Near completion of ECP

- Sustainable software design ready for the addition of new backends.

- EuroHPC Project MICROCARD uses Ginkgo



MICROCARD

https://www.microcard.eu

- BMBF PDExa project uses Ginkgo

deal.II

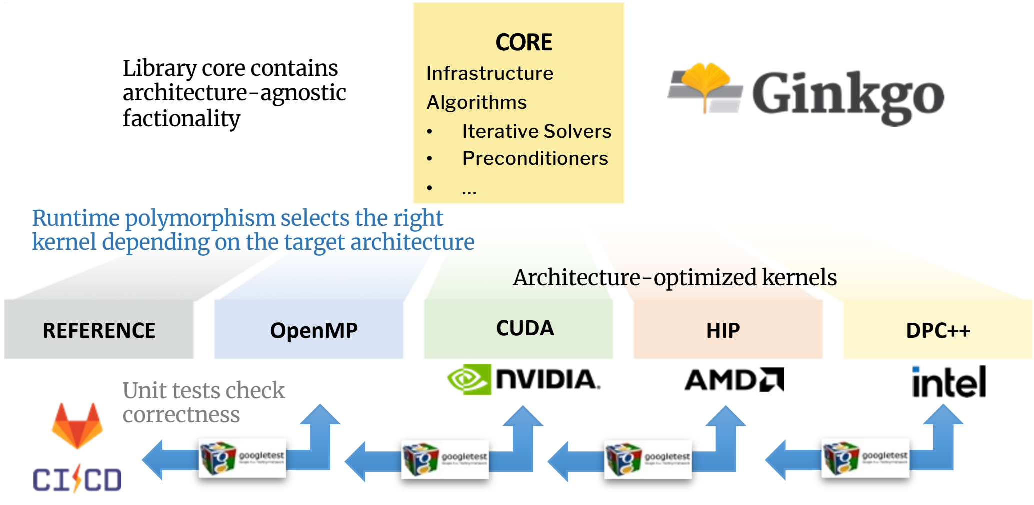
- BMBF ExaSIM project uses Ginkgo

exasim

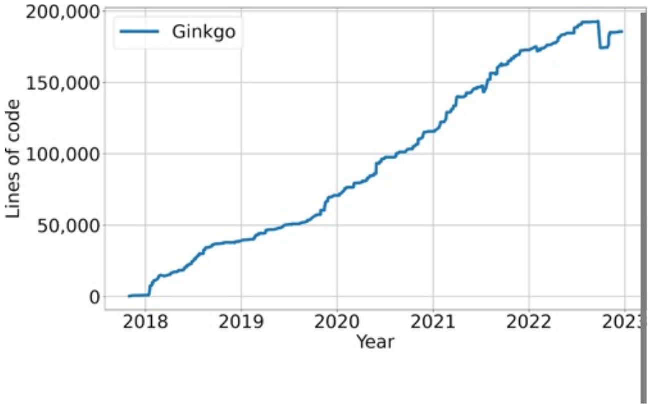
OpenFOAM

The Open Source CFD Toolbox

https://exasim-project.com



	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 (spans OMP–DPC++)		
	Symbolic LU	✓	✓	✓	✓
	Numeric LU	✓	✓	✓	
	Sparse TRSV	✓	✓	✓	
Utilities	On-Device Matrix Assembly	✓	✓	✓	✓
	MC64/RCM reordering	✓			
	Wrapping user data		✓ (spans OMP–DPC++)		
	Logging		✓ (spans OMP–DPC++)		
	PAPI counters		✓ (spans OMP–DPC++)		



Industry Collaboration with bi-weekly meetings

EXASGD

EXASCALE COMPUTING PROJECT

Lessons learnt from the Ginkgo development process

- **ECP earmarking roughly half the budget to Software & App development is a game changer.**
  - **Central component for the success of ECP.**
  - This concept needs to – and does become - the blueprint for other nations and projects.

- **Workforce recruitment and workforce retention are the key to success in software development.**
  - Money does not write software. RSEs do. **We need to create attractive career plans.**
  - We need to make research software development attractive to students. **Academic recognition.**

- **Anticipating the future in hardware development accelerates the porting process.**
  - **Blueprints** and **early access systems** both useful.
  - **Interaction with industry** is mutually beneficial.

- **Management, tools, and strategic initiatives, interaction and collegial behavior are important.**
  - Jira/Notion/[…] milestones and deliverables give projects and collaborative interactions a structure and timeline.
  - **Strategic focus groups, conferences,** and **meetings** bring experts together and **create collaboration**.
  - **Listen to the application needs. Value input and acknowledge collaborators.**

EXASCALE COMPUTING PROJECT