MaX DRIVING THE EXASCALE TRANSITION

Software engineering towards exascale: domain specific libraries, communication optimality, and machine learning

J. VandeVondele, A. Kozhevnikov, M. Kabic,
S. Jakobovits, S. Pintarelli, S. Frasch, M. Taillefumier
CSCS, ETH Zurich

EuroHPC summit, March 24th, 2021

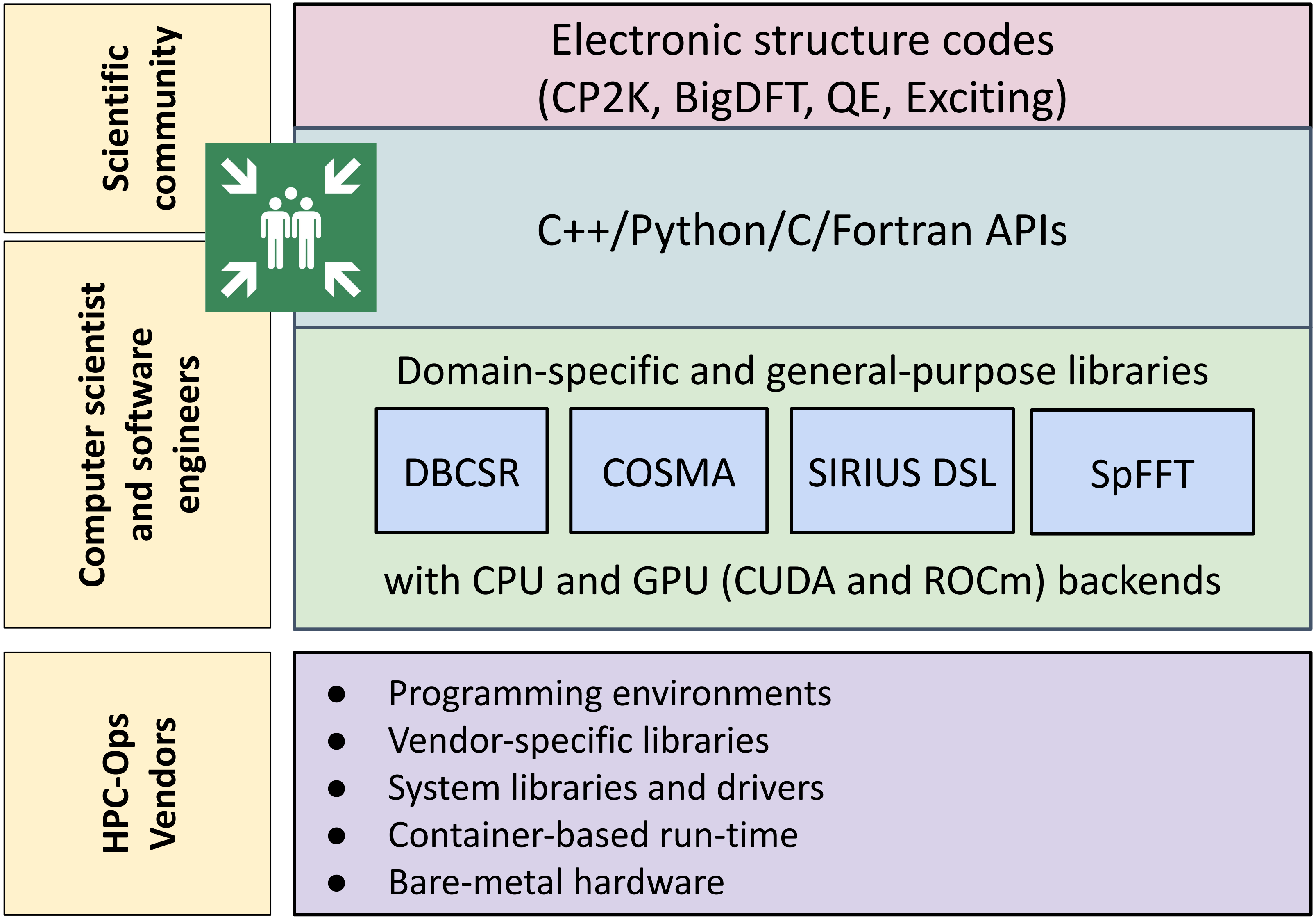
MaX "*Materials Design at the Exascale*", has received funding from the European Union's Horizon 2020 project call H2020-INFRAEDI-2018-1, grant agreement 824143

Overview

within MaX2, ETH Zurich delivers and promotes:

- The principle of separation of concerns
- general-purpose and domain specific libraries
  - NVIDIA GPU and AMD GPU hardware readiness
  - Communication-optimal algorithms
  - Machine-learning techniques for the performance
- Access to novel architectures and programming paradigms
- Knowledge transfer on the topics of HPC and HTC

Current software stack



Scientific community

Computer scientist and software engineers

HPC-Ops Vendors

Electronic structure codes
(CP2K, BigDFT, QE, Exciting)

C++/Python/C/Fortran APIs

Domain-specific and general-purpose libraries

DBCSR

COSMA

SIRIUS DSL

SpFFT

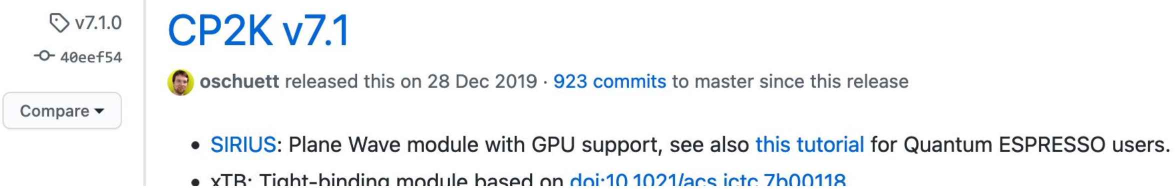
with CPU and GPU (CUDA and ROCm) backends

- Programming environments
- Vendor-specific libraries
- System libraries and drivers
- Container-based run-time
- Bare-metal hardware

SIRIUS DSL | Overview

SIRIUS is a domain specific library for electronic structure calculations, and is designed for GPU acceleration of popular community codes such as BigDFT, CP2K, Exciting, Elk and Quantum ESPRESSO

- MPI and OpenMP parallel
- Nvidia and AMD GPUs
- Fully featured



CP2K v7.1

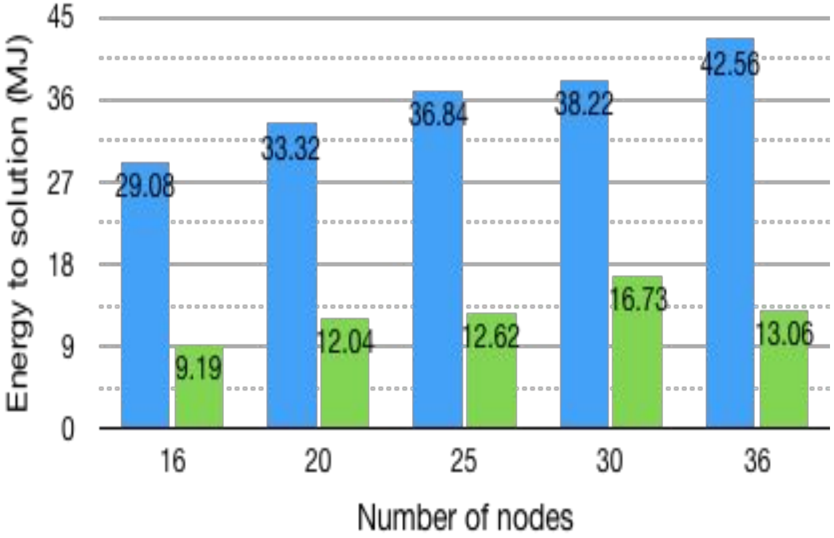
oschuett released this on 28 Dec 2019 · 923 commits to master since this release

- SIRIUS: Plane Wave module with GPU support, see also this tutorial for Quantum ESPRESSO users.
- xTB: Tight-binding module based on doi:10.1021/acs.jctc.7b00118

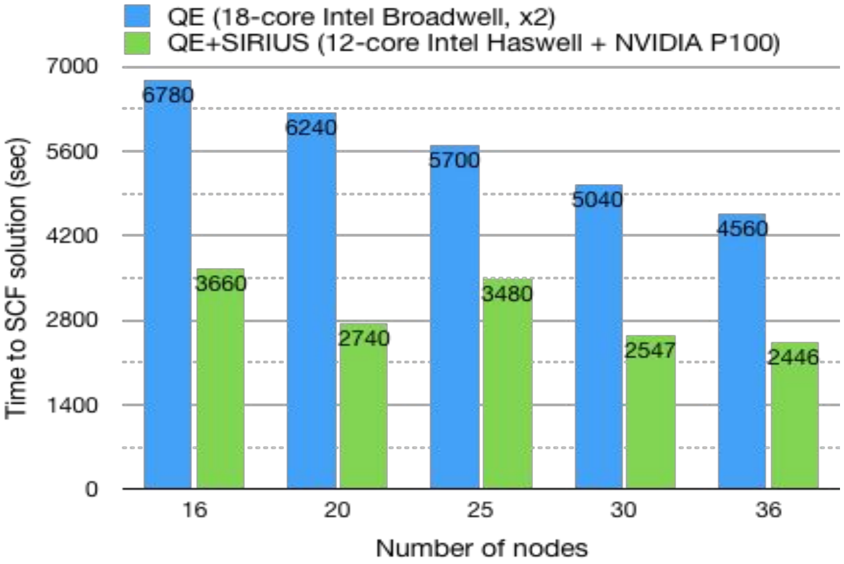
Features		
Pseudopotential	Full-potential	Common
● NC/US/PAW pseudopotentials ● Collinear and non-collinear magnetism ● Hubbard U correction ● Spin-orbit coupling ● Stress tensor ● Atomic forces ● Verification of S-operator matrix ● Iterative Davidson and exact diagonalization solvers	● L(A)PW+lo method with arbitrary number of local orbitals ● Collinear and non-collinear magnetism with second variational approach ● Iterative Davidson and exact diagonalization solvers ● Spin-orbit coupling ● Atomic forces	● Python frontend ● Symmetrization of lattice-periodic functions and on-site matrices (using symmetries from *spglib*) ● Generation of **k**-point mesh using *spglib* ● Run-time control of the eigenvalue solvers (Lapack / MAGMA / ScaLAPACK / ELPA) ● Run-time control of the BLAS provider (CPU BLAS / cuBlas / cuBlasXt)

SIRIUS DSL | GPU (NVIDA and AMD) backend for Quantum ESPRESSO

ENERGY TO SOLUTION



TIME TO SOLUTION

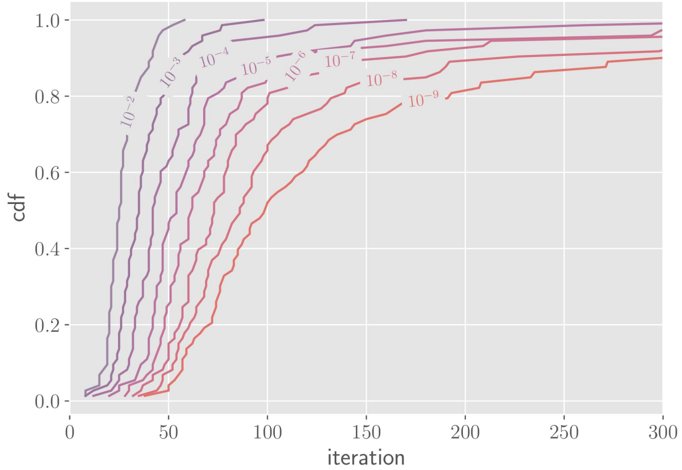


A single GPU node of Piz Daint outperforms a multicore node, in both time and energy to solution by 200% - 300%

443-atom unit cell of IrC

Enables high throughput computing (HTC) with robust methods

- The standard SCF cycle often fails to converge
- Direct minimization algorithms provide a robust alternative method
- Enable Exascale-level HTC calculations based on the AiiDA framework.

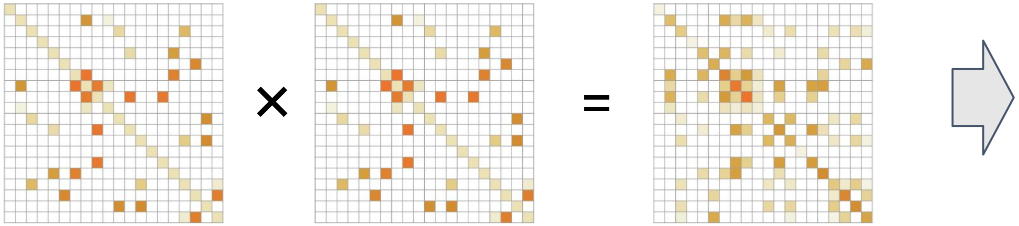


100 metallic (magnetic) systems failing to converge with the SCF cycle. Recent tests carried out by Marnik Bercx from EPFL yielded a 98% success rate with tolerance 10^{-10} (NC pseudos). Results for the ultrasoft case are in preparation.

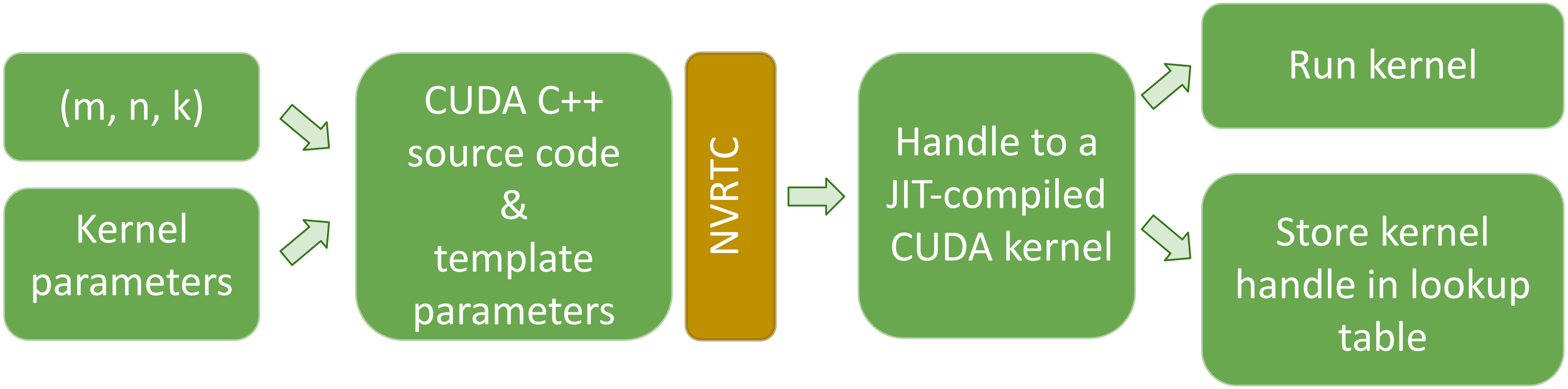
- Robust optimization is implemented in the SIRIUS python module or in C++ library
- Performance portability (Nvidia or AMD gpus) via Kokkos and GPU BLAS/LAPACK libraries
- In SIRIUS enabled QuantumESPRESSO

DBCSR library | Overview

DBCSR is a highly optimized library designed to perform sparse matrix-matrix multiplication at scale.



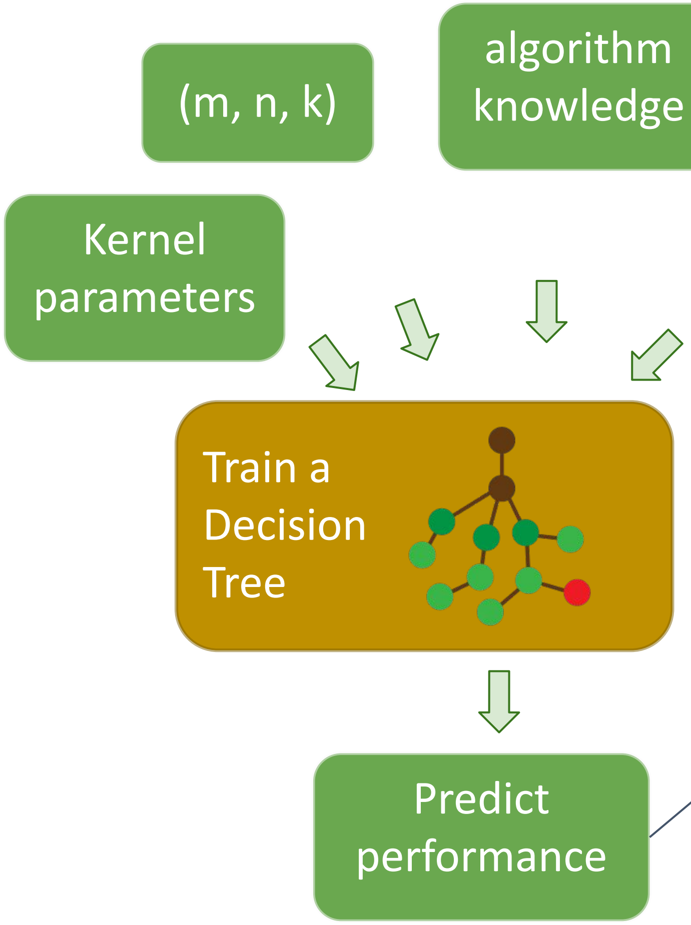
broken down into small dense blocks, run on GPU


(m, n, k)
Kernel parameters
CUDA C++ source code & template parameters
NVRTC
Handle to a JIT-compiled CUDA kernel
Run kernel
Store kernel handle in lookup table

10’000s of kernels each with 10’000s of possible parameters
Predict optimal implementations of all kernels using ML!

DBCSR library | Machine Learning for optimal compute kernel parameters

Finds near-optimal kernels within 2% of best possible performance



(m, n, k)

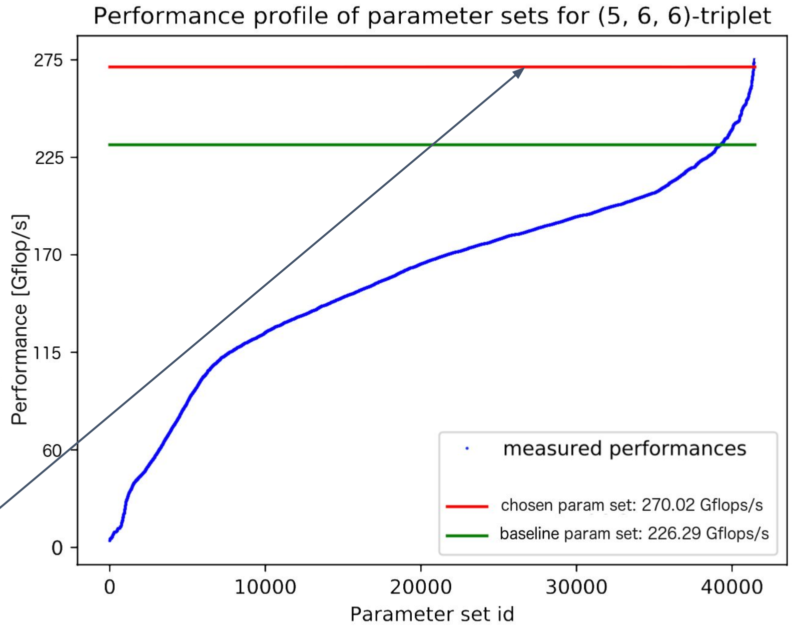
algorithm knowledge

Kernel parameters

hardware knowledge

Train a Decision Tree

Predict performance



Performance profile of parameter sets for (5, 6, 6)-triplet

CUDA / ROCm and OpenCL backends are implemented!

COSMA library | Motivation

Achieve communication-optimality in matrix-matrix multiplication, 50 years of algorithmic progress!

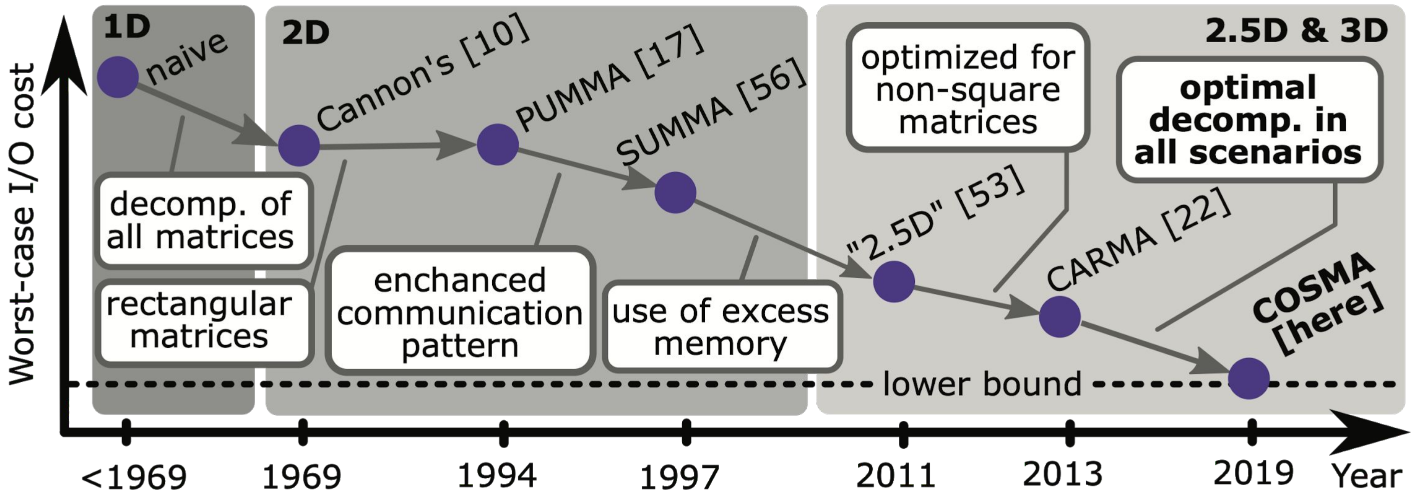
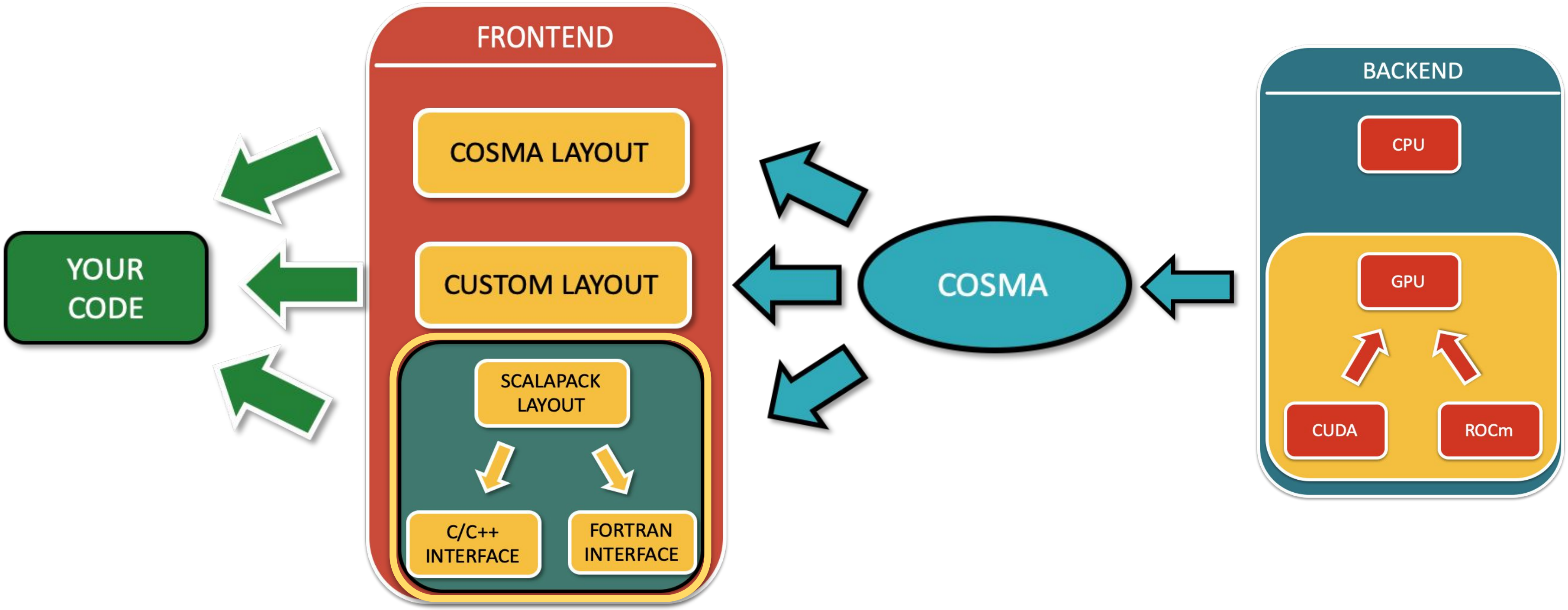


Figure 2: Illustratory evolution of MMM algorithms reaching the I/O lower bound.

COSMA is communication-optimal for all combinations of matrix dimensions, number of processors and memory sizes, without the need for any parameter tuning.

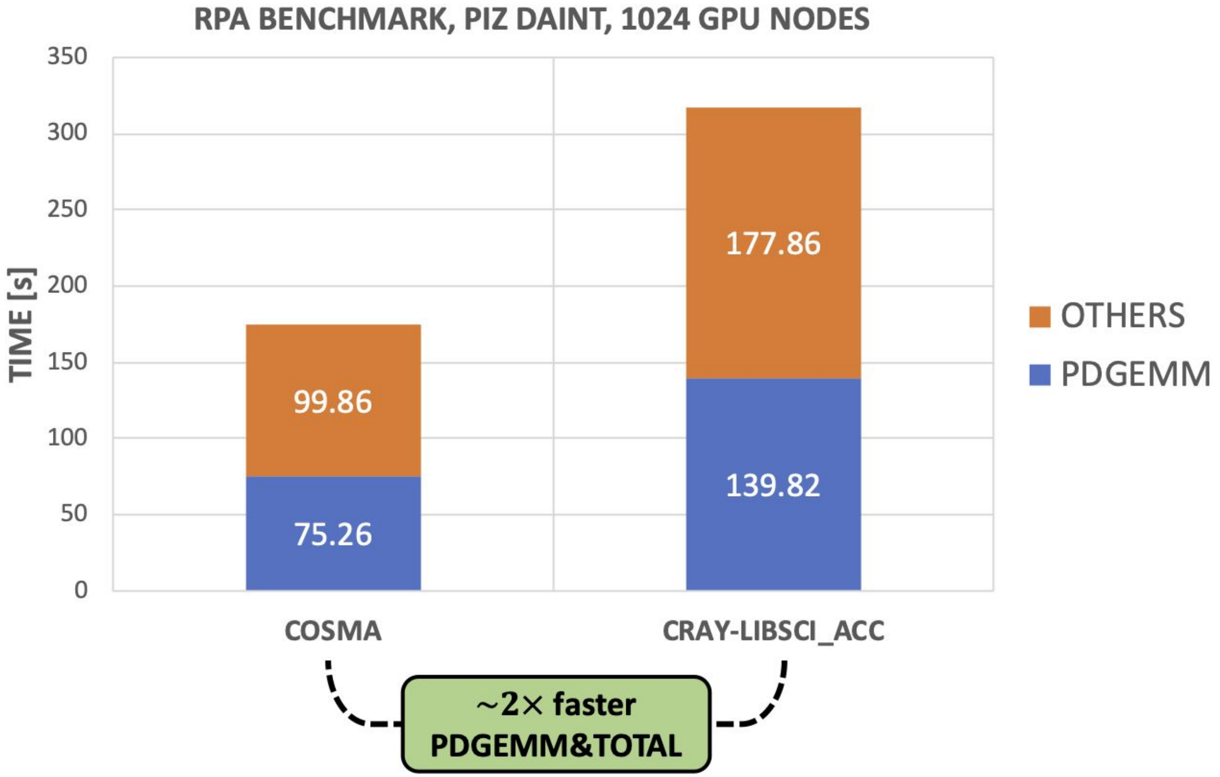
COSMA library | Overview

- MPI and OpenMP parallel
- Nvidia and AMD GPUs via CUDA and ROCm
- Drop-in replacement for ScaLAPACK, no re-coding needed



COSMA library | Performance

On 1024 nodes, the speedup becomes even very significant, outperforming vendor specific solutions significantly



~2× faster PDGEMM&TOTAL

COSMA library is fully integrated in the CP2K toolchain and used in production in order to accelerate large tall-and-skinny matrix multiplications arising in RPA calculations.

DRIVING
THE EXASCALE
TRANSITION

Thanks you! Questions?

MaX “*Materials Design at the Exascale*”, has received funding from the European Union’s Horizon 2020 project call H2020-INFRAEDI-2018-1, grant agreement 824143