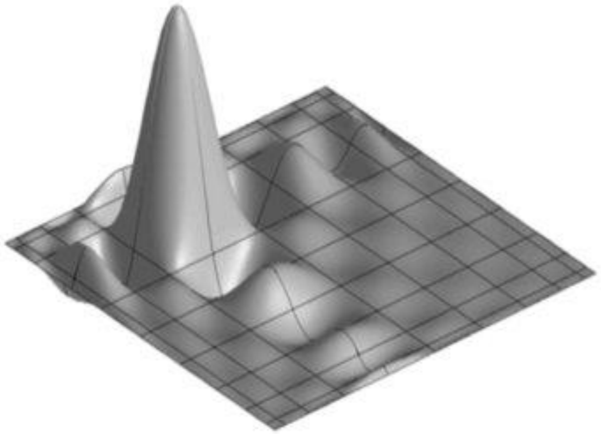
Center for Efficient Exascale Discretizations (CEED)

Tzanio Kolev (LLNL)
CEED PI

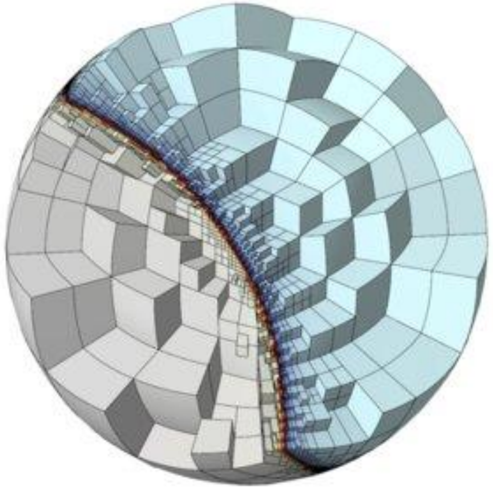
- PDE-based simulations on **unstructured grids**
- **high-order** and **spectral** finite elements

✓ *any order space on any order mesh* ✓ *curved meshes,*

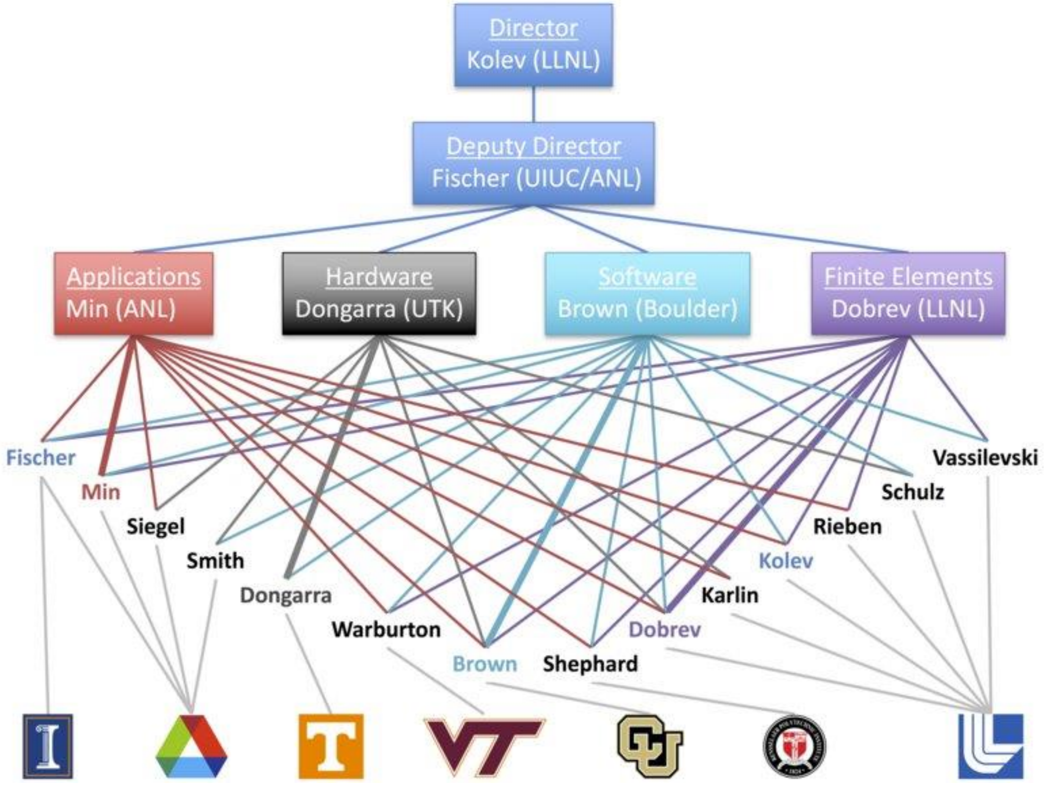
✓ *unstructured AMR* ✓ *optimized low-order support*



10th order basis function



non-conforming AMR, 2nd order mesh



2 Labs, 5 Universities, 30+ researchers

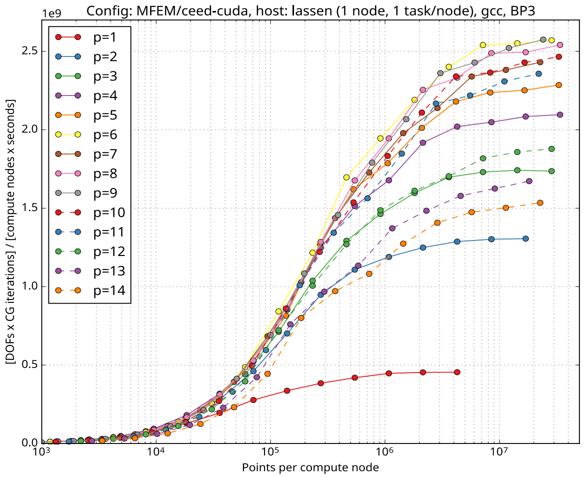
Why High-Order?

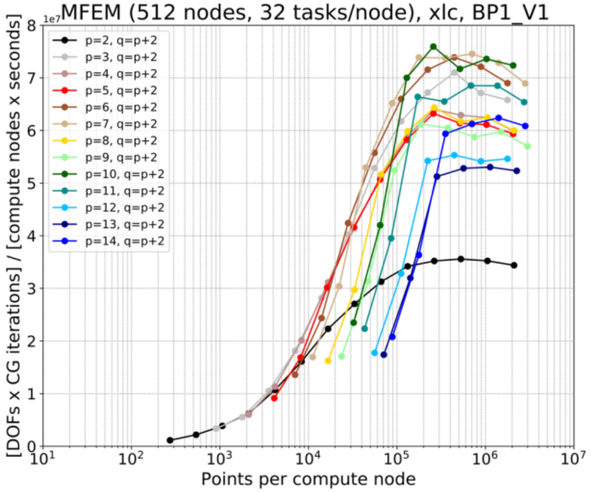
Better HPC

- ✓ *multiple levels of parallelism*
- ✓ *inter-device parallel sparse linear algebra*
- ✓ *on-device dense linear algebra*
- ✓ *batched tensor contractions*
- ✓ *FLOPs/bytes increase with order*

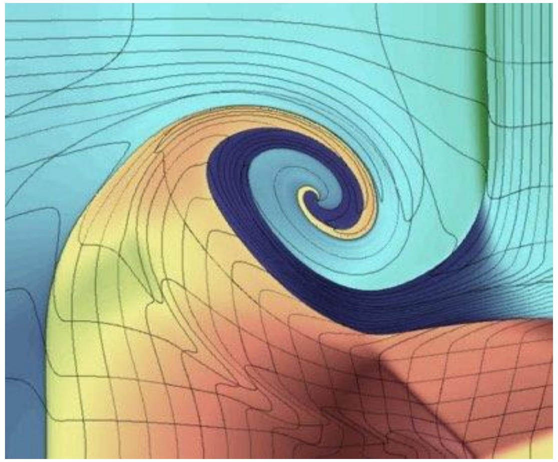
Better science

- ✓ *naturally support unstructured and curvilinear grids*
- ✓ *increased accuracy for smooth problems*
- ✓ *sub-element modeling for problems with shocks*
- ✓ *better symmetry, conservation, robustness, …*
- ✓ *many apps: compressible flow, AM, fusion, …*

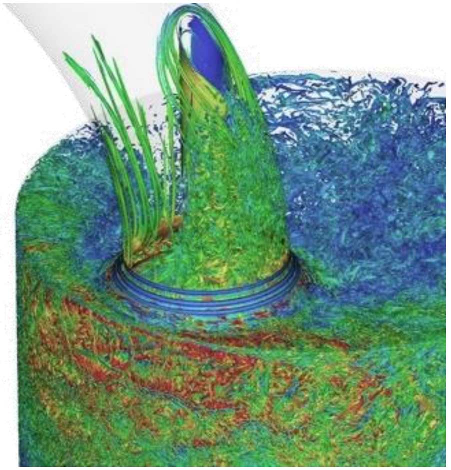




FEM benchmark problems on GPU (left) and CPU (right): high-order kernels have up to 5x better performance for same #dofs



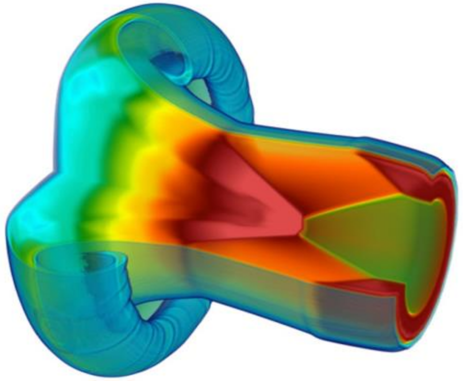
8th order Lagrangian shock hydro (MFEM)



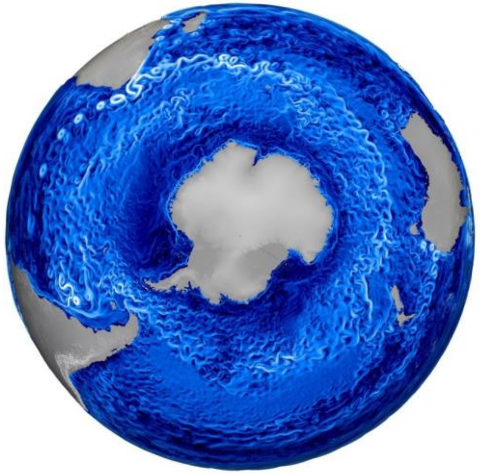
6th order DNS turbulence (Nek)

CEED Target Applications

CEED EXASCALE DISCRETIZATIONS

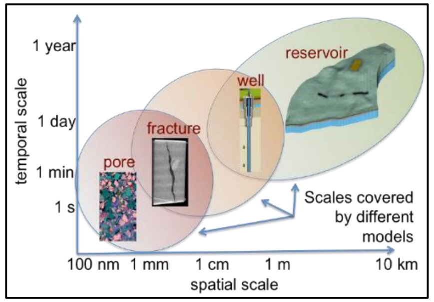


Compressible flow (MARBL)

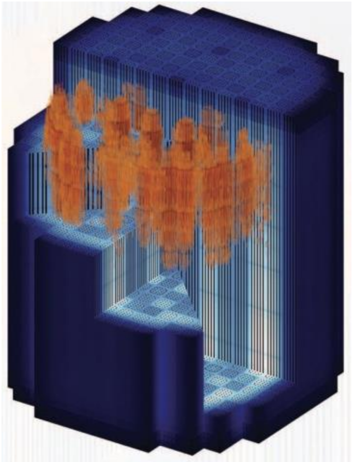


Climate (E3SM)

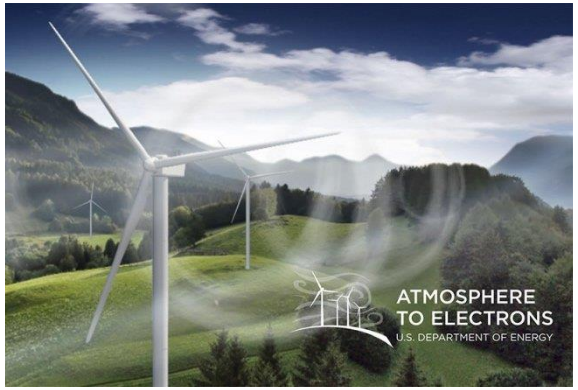
Urban systems (Urban)



Subsurface (GEOS)

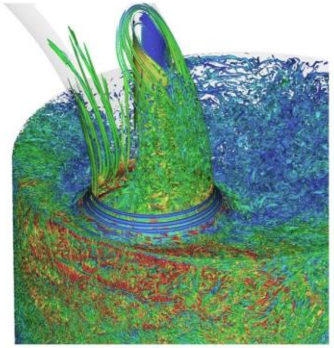


Modular Nuclear Reactors (ExaSMR)

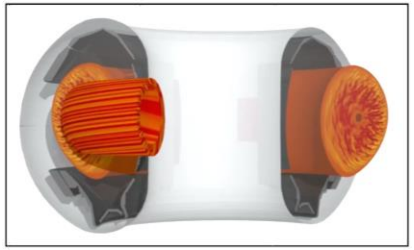


Wind Energy (ExaWind)

Additive Manufacturing (ExaAM)



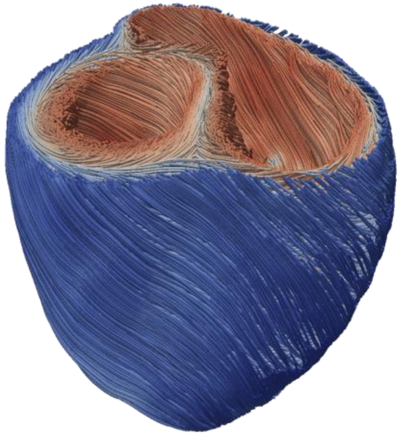
Combustion (Nek5000)



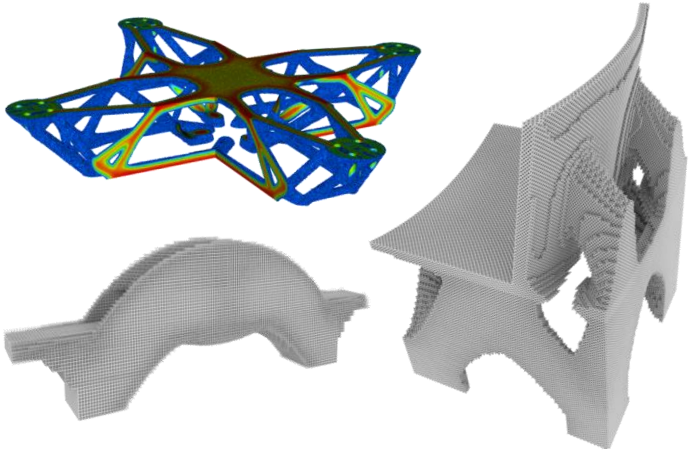
Magnetic Fusion (WDMApp)

We are interested in working with other applications!

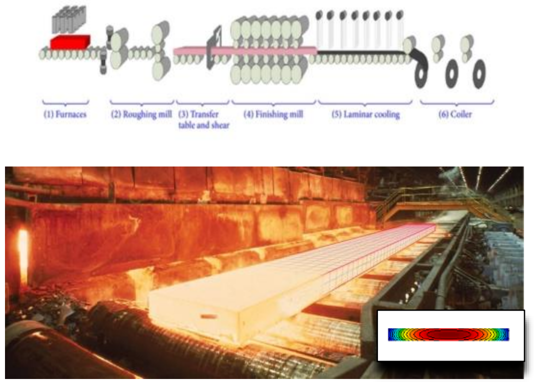
Additional Application Collaborations



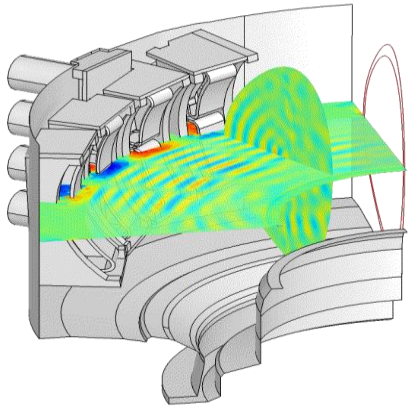
Heart Modelling (Cardioid)



Topology optimization for additive manufacturing

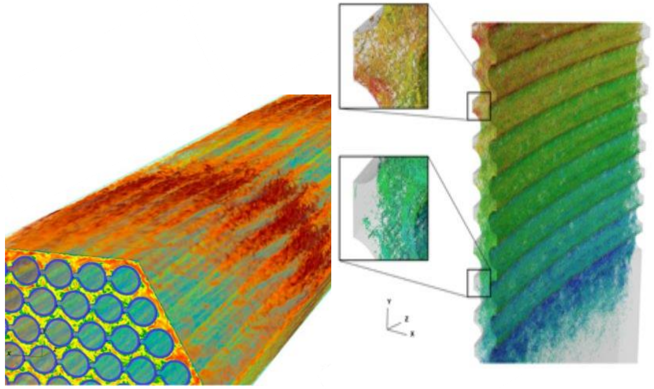


Hot strip mill slab modeling (U. S. Steel)

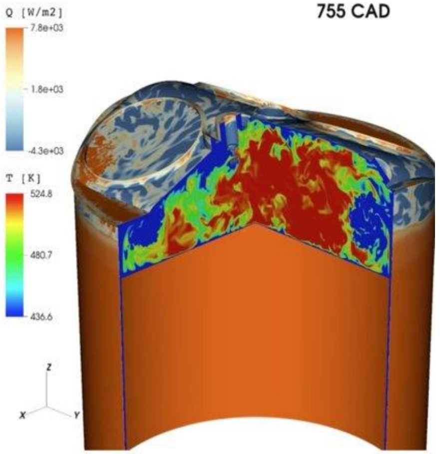


Core-Edge tokamak EM wave propagation (SciDAC)

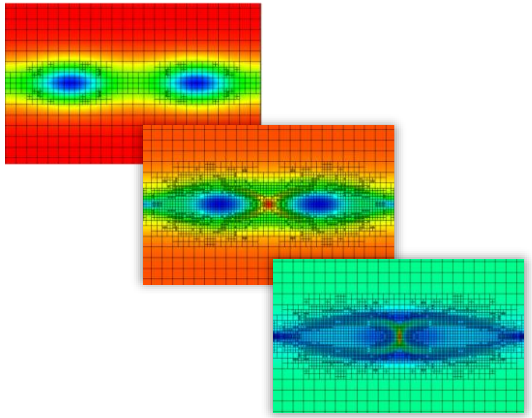
Electric aircraft design



Reactor simulation (DOE NEAMS)



Engine simulation (DOE VTO)



Adaptive MHD island coalescence (SciDAC)

CEED Discretization Libraries

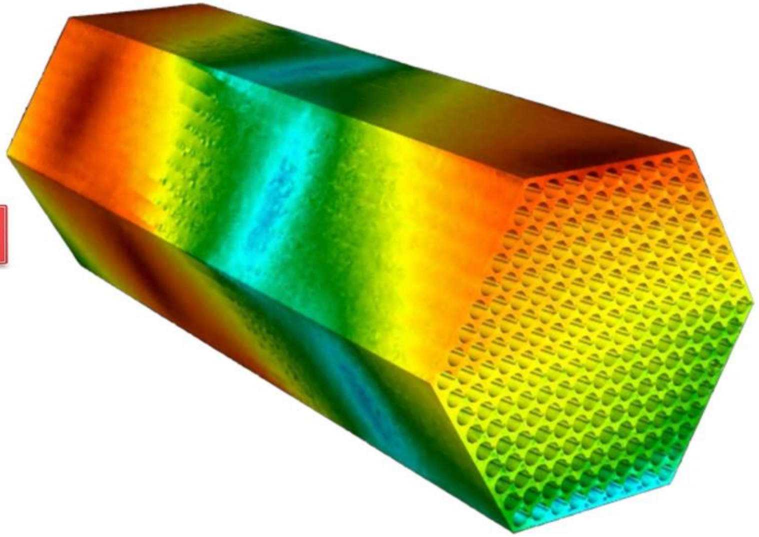
CEED EXASCALE DISCRETIZATIONS

✓ *better exploit the hardware to deliver significant performance gain over conventional methods*

✓ *based on MFEM/Nek, low & high-level APIs*

Nek5000

NekRS new

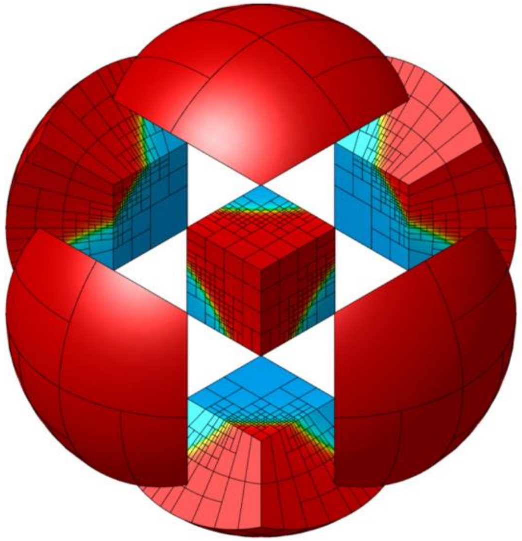


nek5000.mcs.anl.gov

High-performance spectral elements

MFEM

v4.1 new



mfem.org

Scalable high-order finite elements

CEED Miniapps

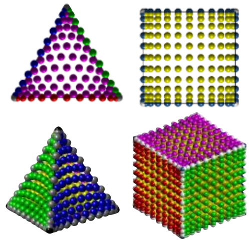
ceed.exascaleproject.org/miniapps

- **Nekbone / NekBench** (Nek5000) *new*
  - A lightweight subset of Nek5000; solves a standard Poisson equations; weak-scaled to 6 million MPI ranks. Currently support OpenACC/CUDA-based GPU variants.
- **libParanumal**
  - An experimental testbed for multi-level parallel implementations of high-order finite element computations; under development.
- **Laghos** (MARBL) *new*
  - A proxy for the Lagrangian component of MARBL; solves the time-dependent Euler equation of compressible gas; makes use of unstructured moving meshes.
- **Remhos** (MARBL) *new*
  - A proxy the Eulerian component of the MARBL; solves the pure advection equations that are used to perform monotonic and conservative discontinuous field interpolation (remap) in ALE methods.
- **ExaConstit** (ExaAM)
  - A miniapp for the ExaAM project based on MFEM evaluating constitutive material properties at continuum scale.
- **HPGMG-FE**

✓useful for vendor engagements, collaboration ✓ part of CORAL-2, ECP, ASC proxies

libParanumal

github.com/paranumal/libparanumal

ECP Apps	Flow-apps	Time Stepping	Spatial Discretization	Algorithmic Acceleration	Core	External Libraries
NekRS	Incompressible Navier-Stokes	Implicit-explicit	Discontinuous Galerkin	Heterogeneous hybrid multigrid	Iterative solvers	AMG hypre
MFEM	Oseen	Semi-Lagrangian	Continuous Galerkin	Krylov Recycling	OCCA kernels	Portability OCCA
libCEED	Stokes	JFNK	Curvilinear elements	RHS projection methods	Parallel mesh wrangling	Partitioning parRSB
	Compressible Navier-Stokes	Explicit	High-order Elements 	Adaptive time stepping	Gather-scatter ops	GatherScatter gslib
	Galerkin-Boltzmann	Semi-analytic		Spatial adaptivity	Async halo exchanges	Comms MPI

- ✓ libParanumal = ***reference*** library of highly optimized GPU implementations of high-order finite element and discontinuous Galerkin based PDE solvers.
- ✓ Developed by Tim Warburton's group at Virginia Tech

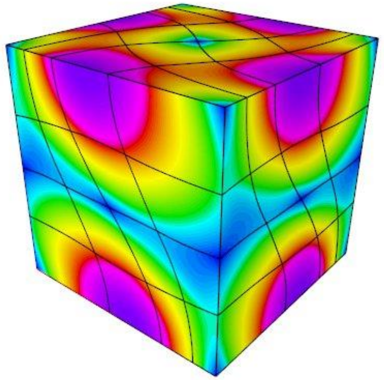
Laghos

github.com/ceed/laghos

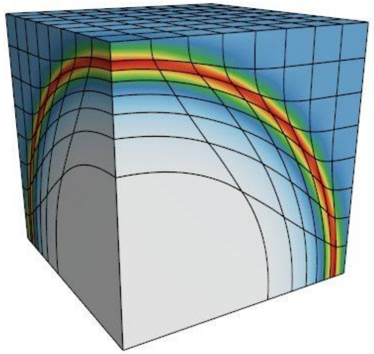
- ✓ Miniapp for HO compressible shock hydrodynamics
  - moving (high-order) curved meshes
  - explicit time integration
- ✓ Proxy for LLNL's Blast code
- ✓ Based on MFEM
  - quick to prototype, C++
  - inherits/motivates performance improvements
- ✓ **Laghos-3.0: device support based on MFEM-4.0**
- ✓ Procurement benchmark for CTS-2 at LLNL
- ✓ Used for El Capitan's node architecture decision
- ✓ **Strong collaboration with NVIDIA**
  - Summit on Summit meetings
  - Optimizing the Laghos-2.0 CUDA version

$$\frac{d\rho}{dt} = -\rho \nabla \cdot v, \qquad \frac{dx}{dt} = v$$

$$\rho \frac{dv}{dt} = \nabla \cdot \sigma, \qquad \rho \frac{de}{dt} = \sigma : \nabla v$$



3D Taylor-Green smooth test problem

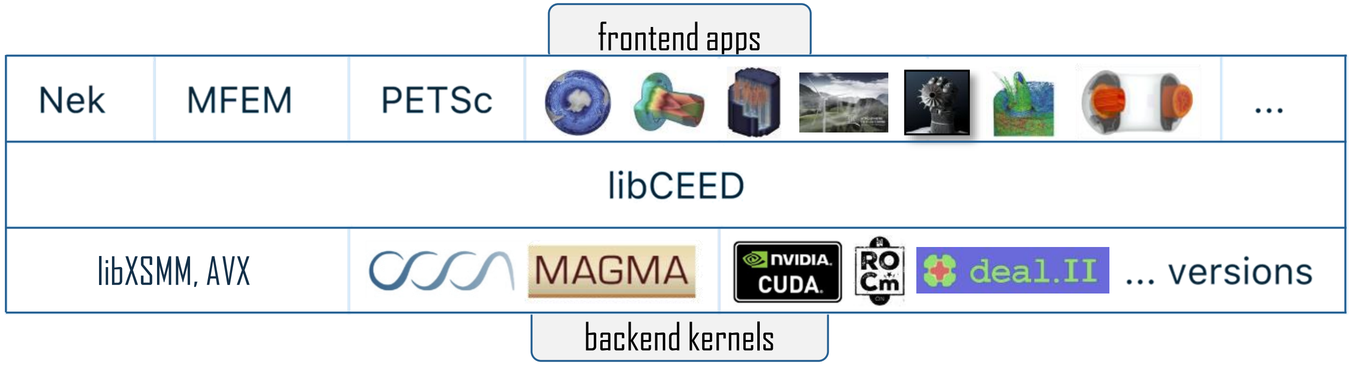


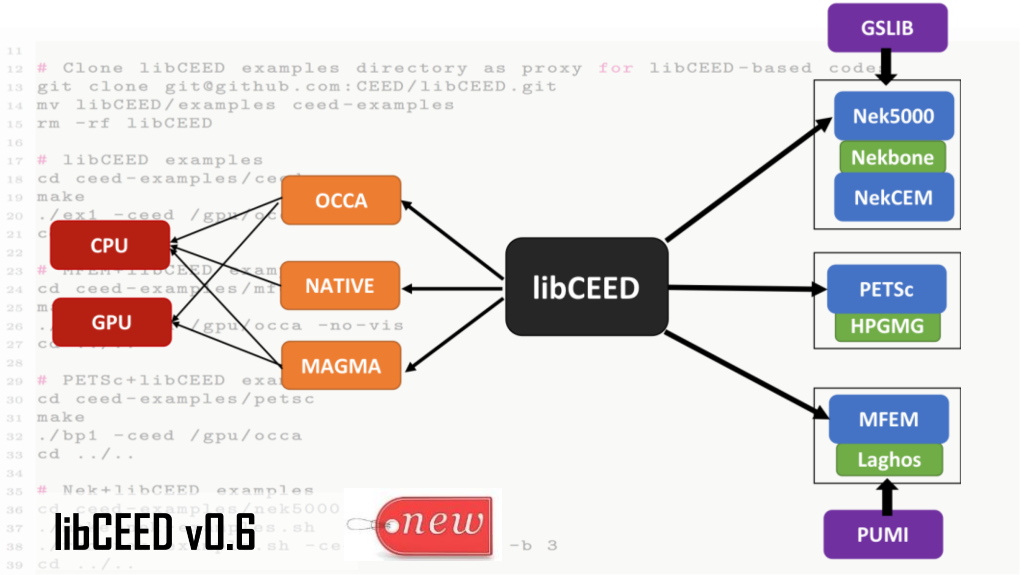
3D Sedov blast shock test problem

libCEED: CEED's Low-Level API Library

lightweight, portable & performant high-order operator evaluation

github.com/ceed/libceed





- ✔ API between *frontend apps* and *backend kernels*
- ✔ *Efficient operator description* (not global matrix)
- ✔ Clients use any backend as a run-time option
- ✔ Backend can be added as plugins without recompiling
- ✔ Backends compete for best performance, latency vs throughput, optimize for order/device, use JIT, …

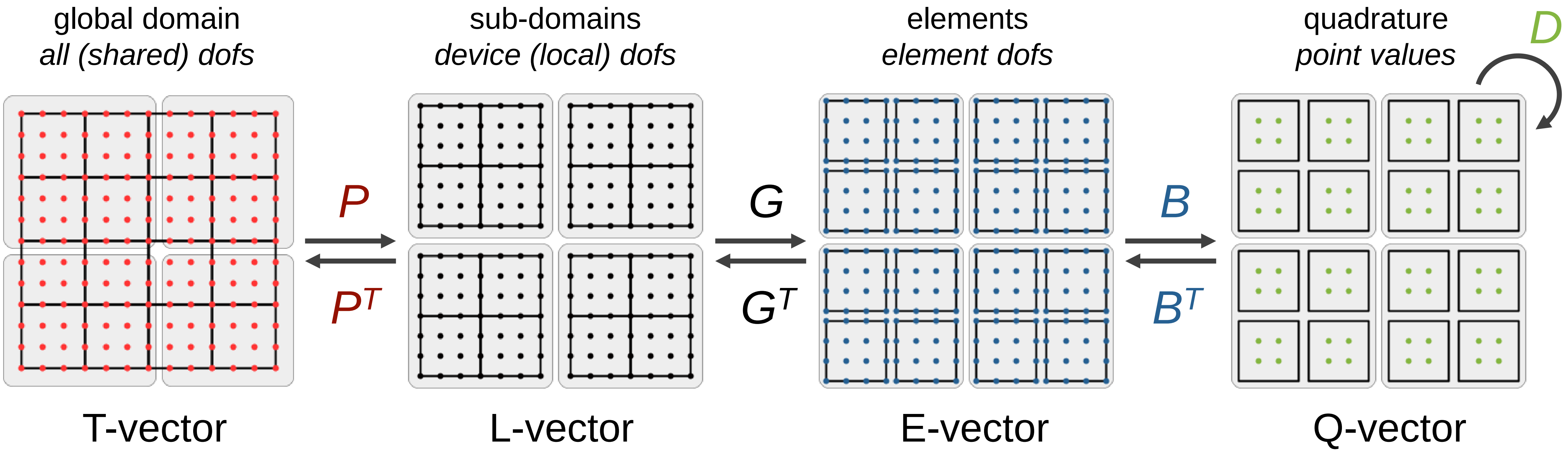
✔ Extensible backends

- **CPU**: reference, vectorized, libXSMM
- **CUDA** using NVRTC cuda-gen
- **OCCA** (JIT): CPU, OpenMP, OpenCL, CUDA
- **MAGMA**

FEM Operator Decomposition

FEM operator assembly/evaluation can be split into parallel, mesh, basis, and geometry/physics parts:

$$A = P^T G^T B^T D B G P$$



- ✓ *partial assembly* = store only *D*, evaluate *B* (tensor-product structure)
- ✓ better representation than *A*: *optimal memory, near-optimal FLOPs*
- ✓ purely algebraic
- ✓ applicable to many apps

CEED benchmarks

github.com/ceed/benchmarks

✓ CEED's *bake-off problems* (BPs) are high-order kernels/benchmarks designed to test and compare the performance of high-order codes.

BP1: Solve {Mu=f}, where {M} is the mass matrix, q=p+2

BP2: Solve the vector system {Mui=fi} with {M} from BP1, q=p+2

BP3: Solve {Au=f}, where {A} is the Poisson operator, q=p+2

BP4: Solve the vector system {Aui=fi} with {A} from BP3, q=p+2

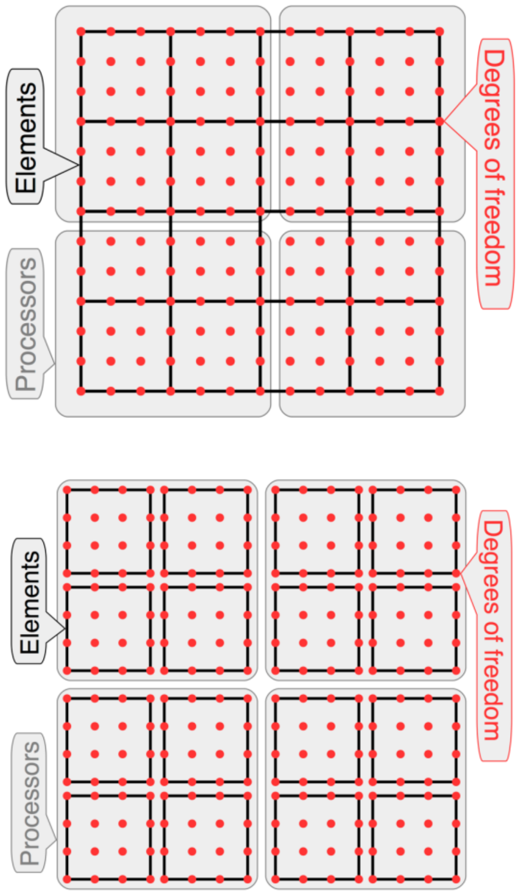
BP5: Solve {Au=f}, where {A} is the Poisson operator, q=p+1

BP6: Solve the vector system {Aui=fi} with {A} from BP3, q=p+1

✓ Mixture of compute-intensive + nearest-neighbor communication + vector reductions

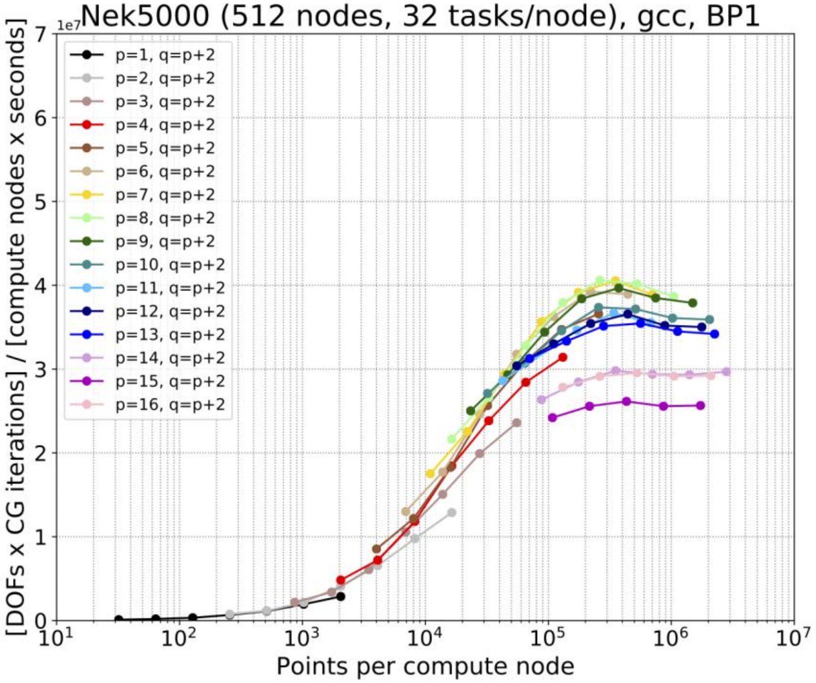
✓ Compared Nek, MFEM, deal.ii on BG/Q, KNLs, GPUs.

✓ Goal is to learn from each other, benefit all HO applications

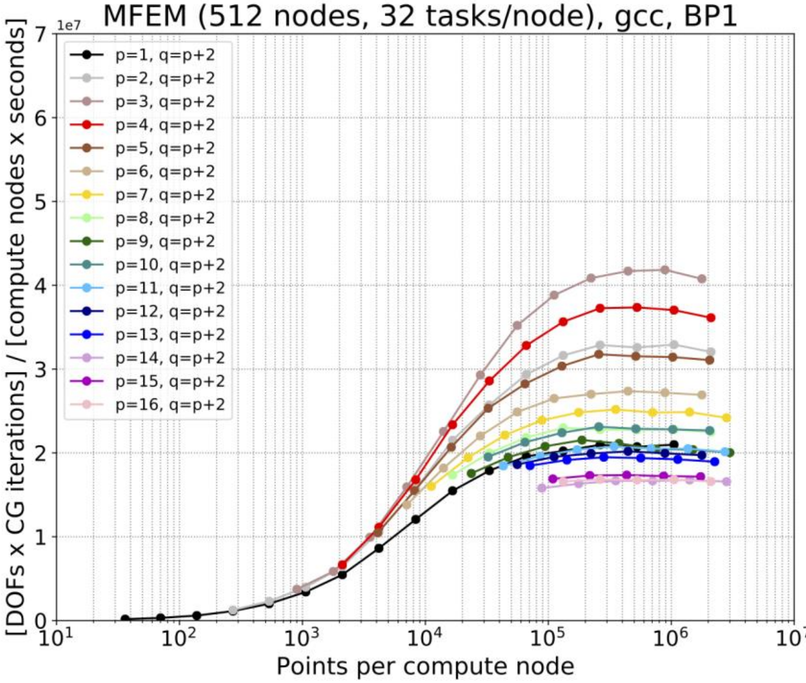


BP terminology: T- and E-vectors of HO dofs

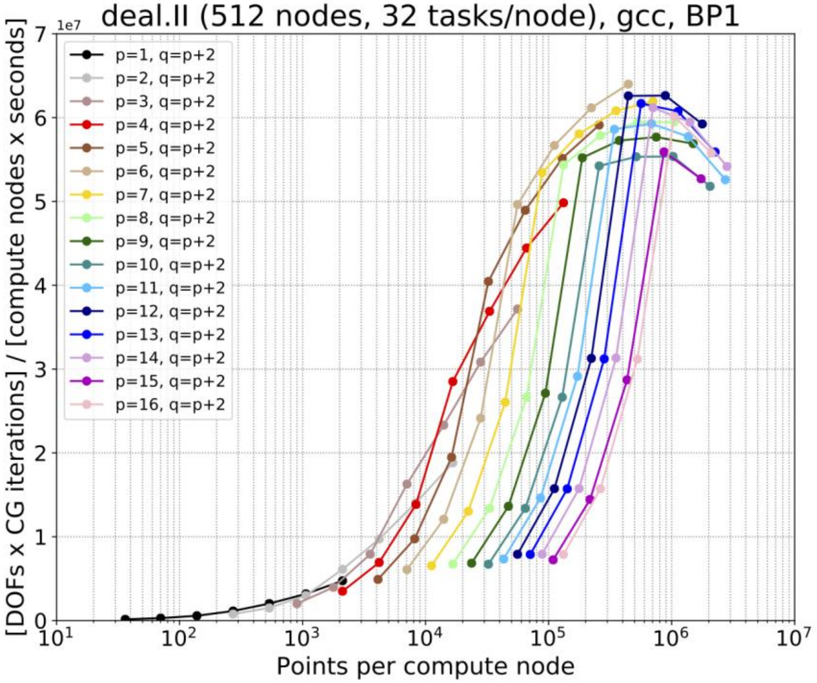
CEED Bake-off Problem 1 on CPU



(a) BP1 Nek5000



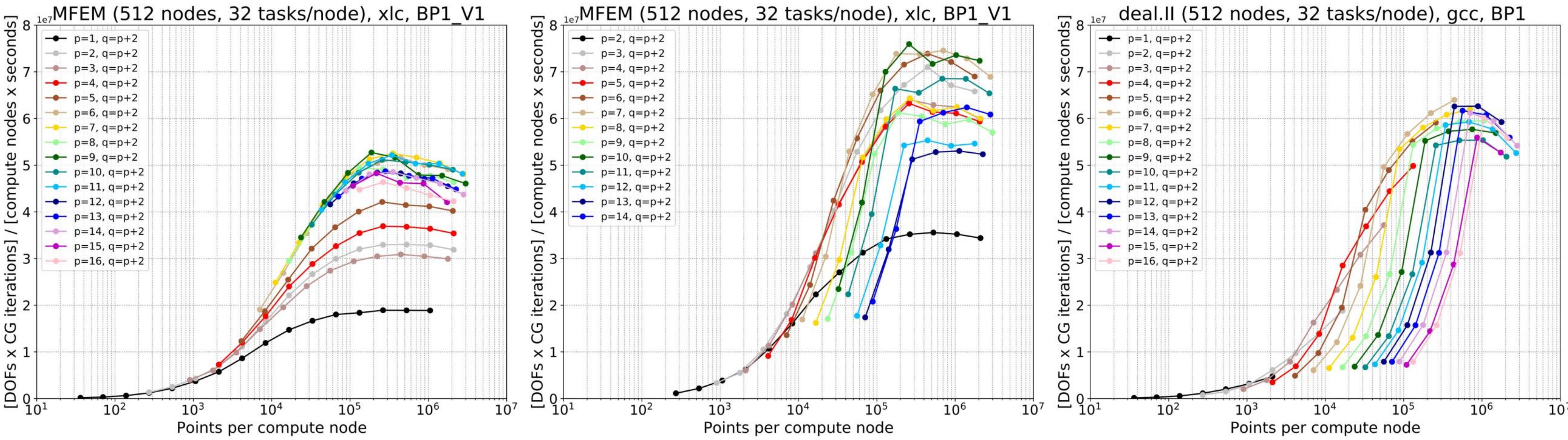
(b) BP1 MFEM



(c) BP1 deal.II

- ✔ All runs done on BG/Q (for repeatability), 16384 MPI ranks. Order p = 1, ...,16; quad. points q = p + 2.
- ✔ BP1 results of Nek (left), MFEM (center), and deal.ii (right) on all using gcc compiler BG/Q.
- ✔ Paper: "Scalability of High-Performance PDE Solvers"
- ✔ deal.ii results from Martin Kronbichler: `github.com/kronbichler/ceed_benchmarks_dealii`

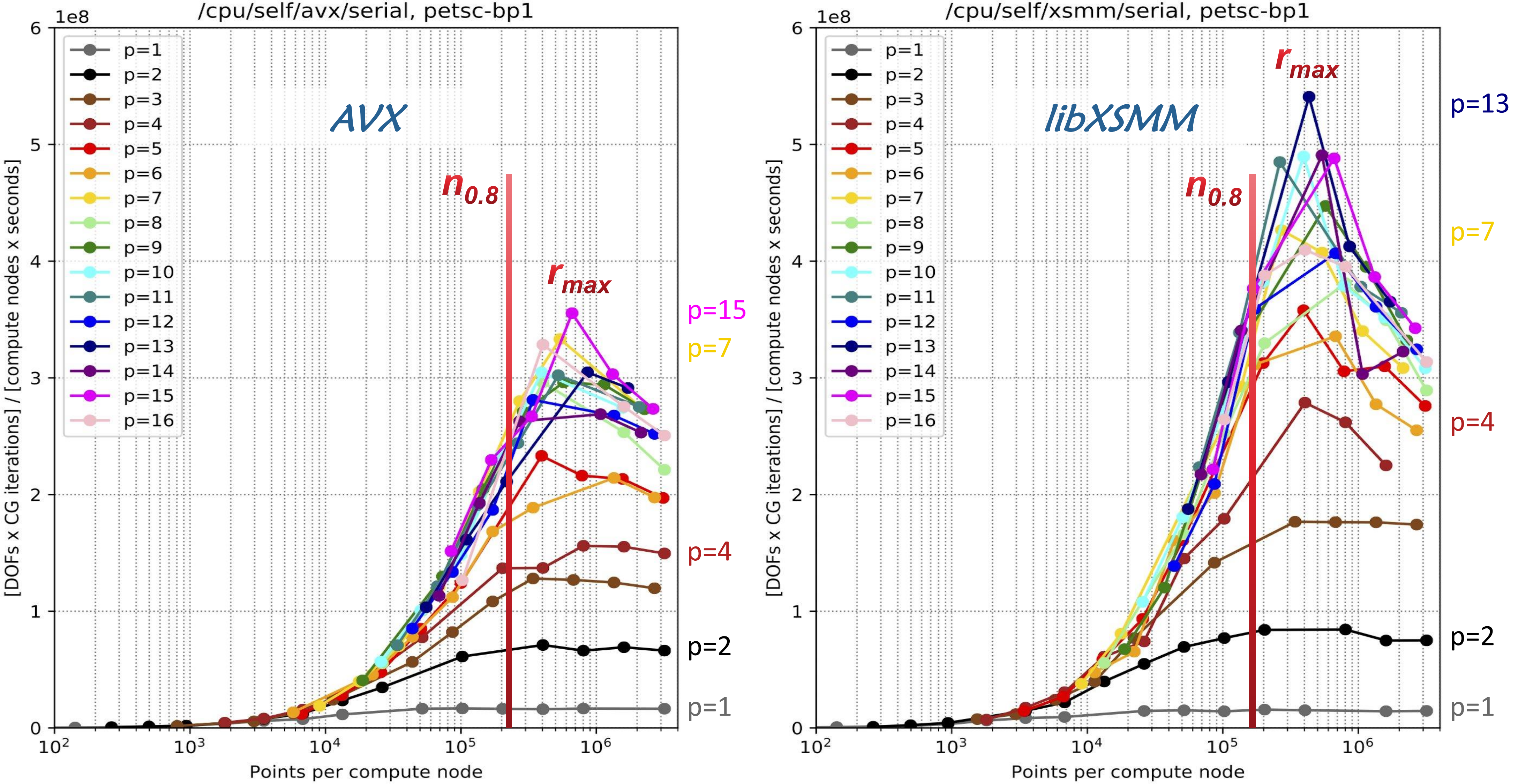
CEED Bake-off Problem 1 on CPU



(a) BP1 MFEM-before **(b)** BP1 MFEM-after **(c)** BP1 deal.II

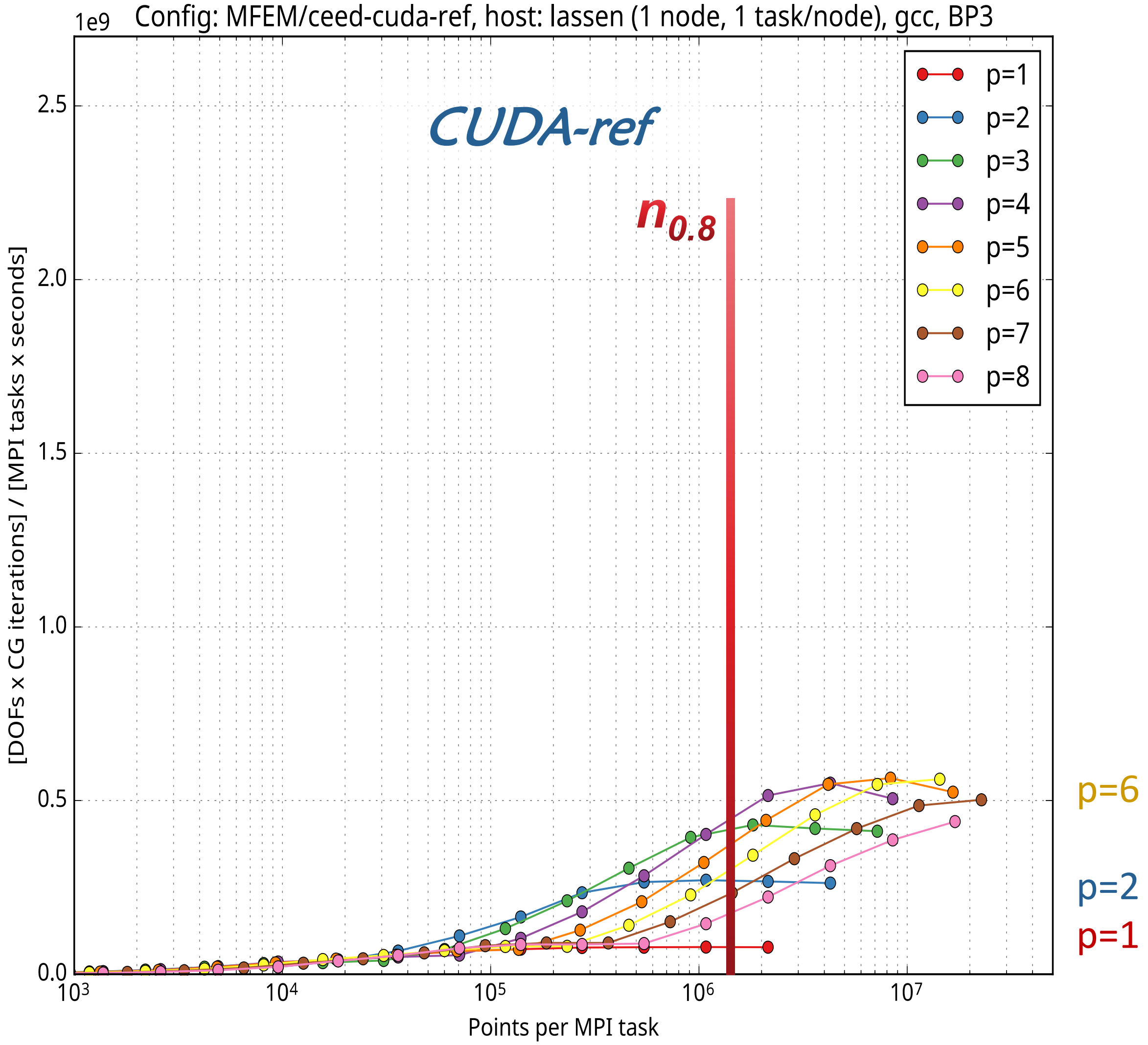
- ✓ All runs done on BG/Q (for repeatability), 16384 MPI ranks. Order $p = 1, \ldots, 16$; quad. points $q = p + 2$.
- ✓ BP1 results of MFEM+xlc (left), MFEM+xlc+intrinsics (center), and deal.ii + gcc (right) on BG/Q.
- ✓ Paper: “Scalability of High-Performance PDE Solvers”
- ✓ Cooperation/collaboration is what makes the bake-offs rewarding.

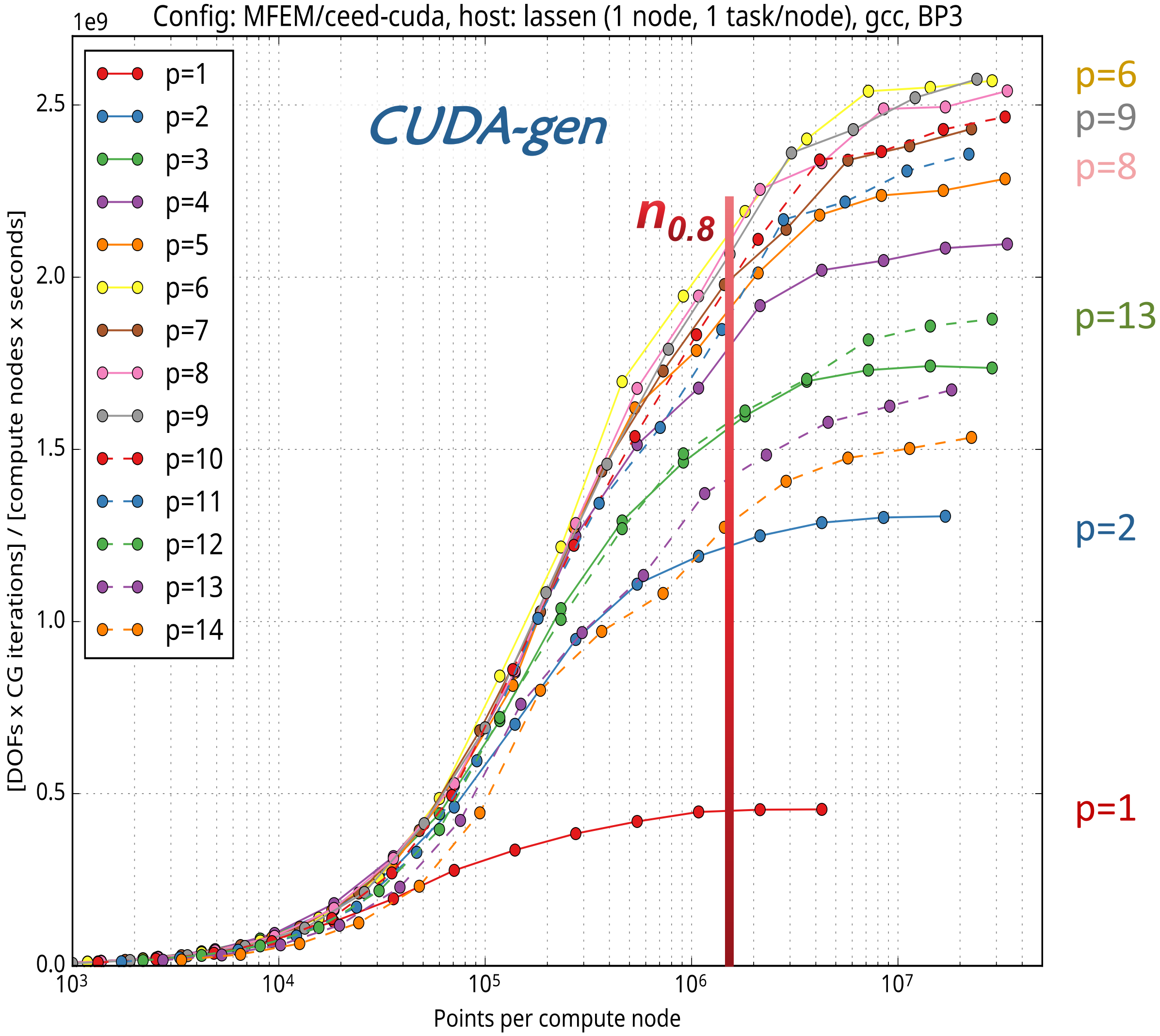
BP1/CPU: AVX + libXSMM backends



PETSc-BP1, 3D, Intel Xeon E5-2680 v3, 24 cores/node (Haswell)

BP3/GPU: CUDA-ref + CUDA-gen backends

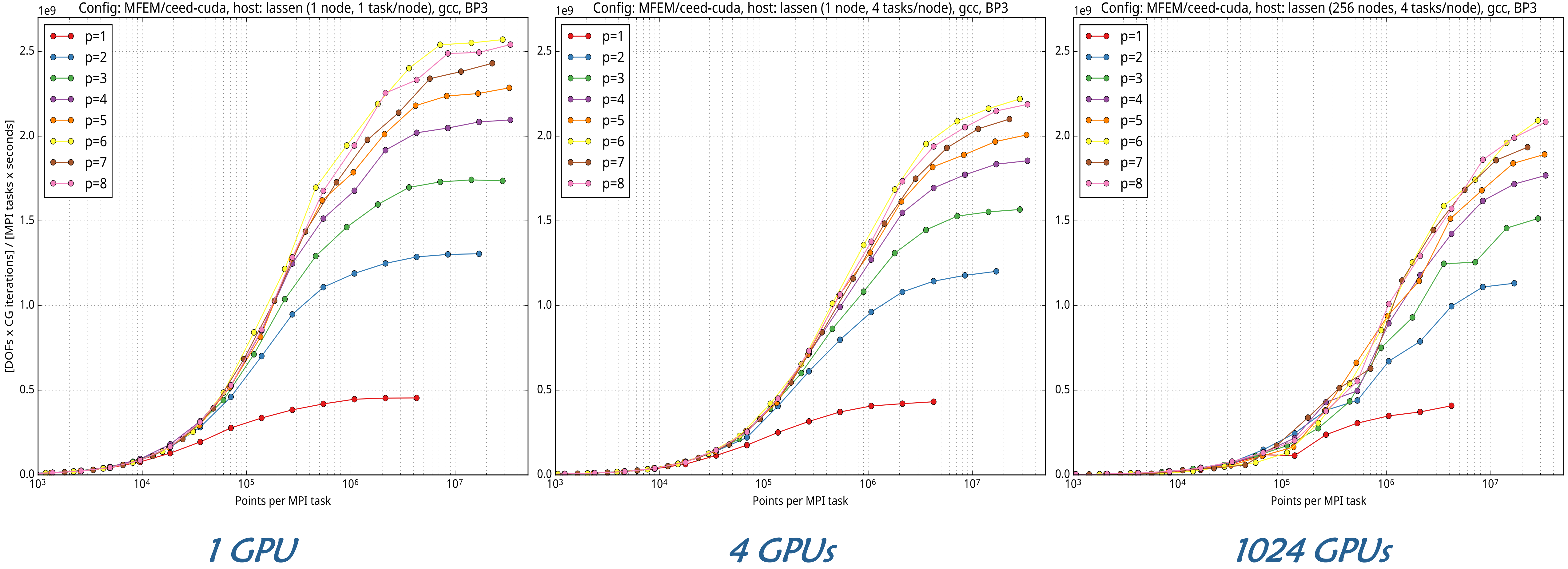




MFEM-BP3, 3D, Lassen 4 x V100 GPUs / node, $n_{0.8}$ about 7.5x larger than Haswell

MFEM performance on multiple GPUs

*MFEM-BP3, **3D**, Lassen 4 x V100 GPUs / node*

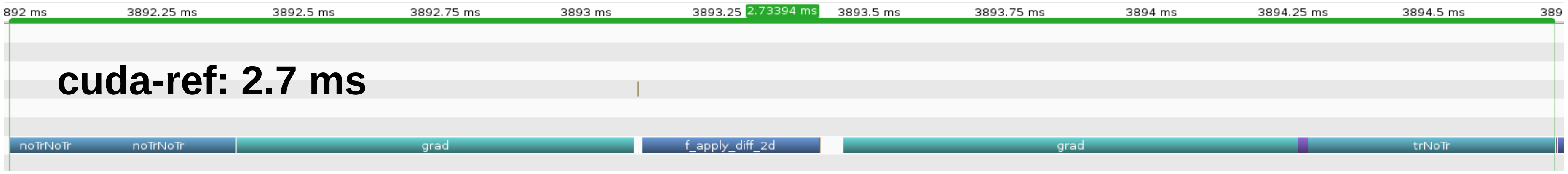


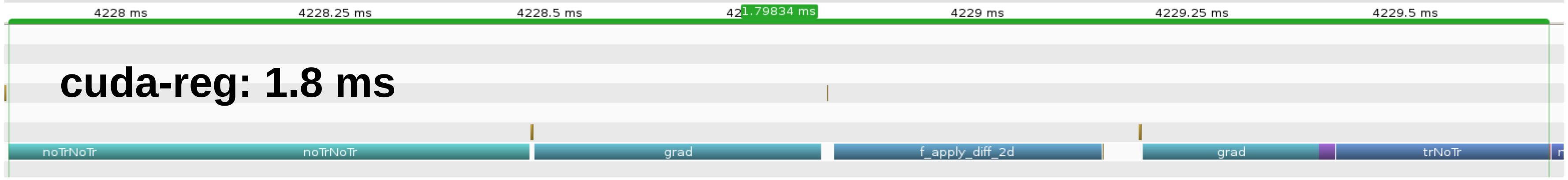
Optimized kernels for MPI buffer packing/unpacking on the GPU
GPU-aware MPI ready

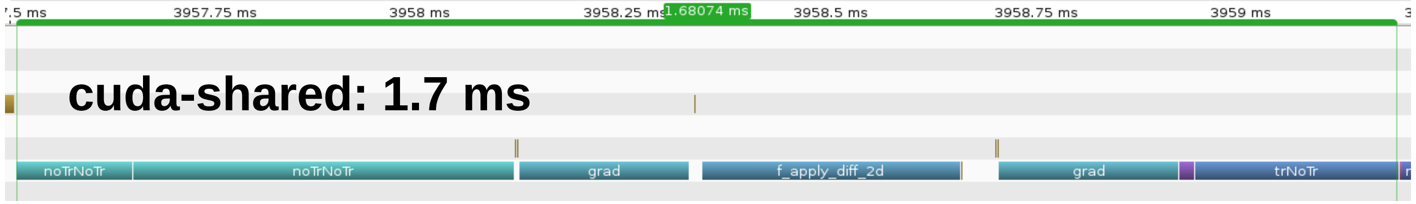
Best total performance: **2.1 TDOF/s**
Largest size: 34 billion

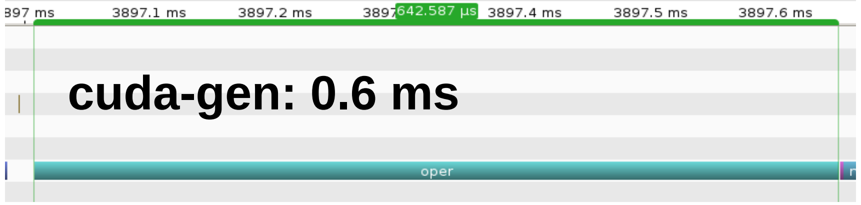
GPU backend performance evolution

	cuda-ref →	cuda-reg →	cuda-shared →	cuda-gen
Automatic host-device memory sync.	✓	✓	✓	✓
Register blocking	✗	✓	✓	✓
Shared memory	✓	✗	✓	✓
Kernels fusion	✗	✗	✗	✓

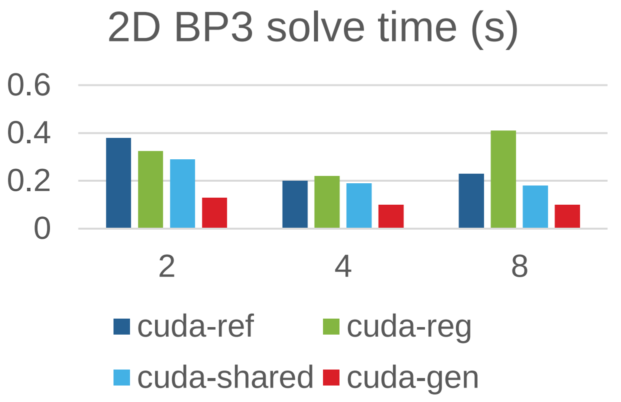


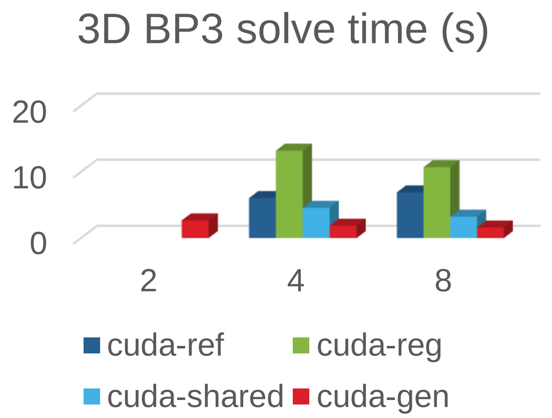






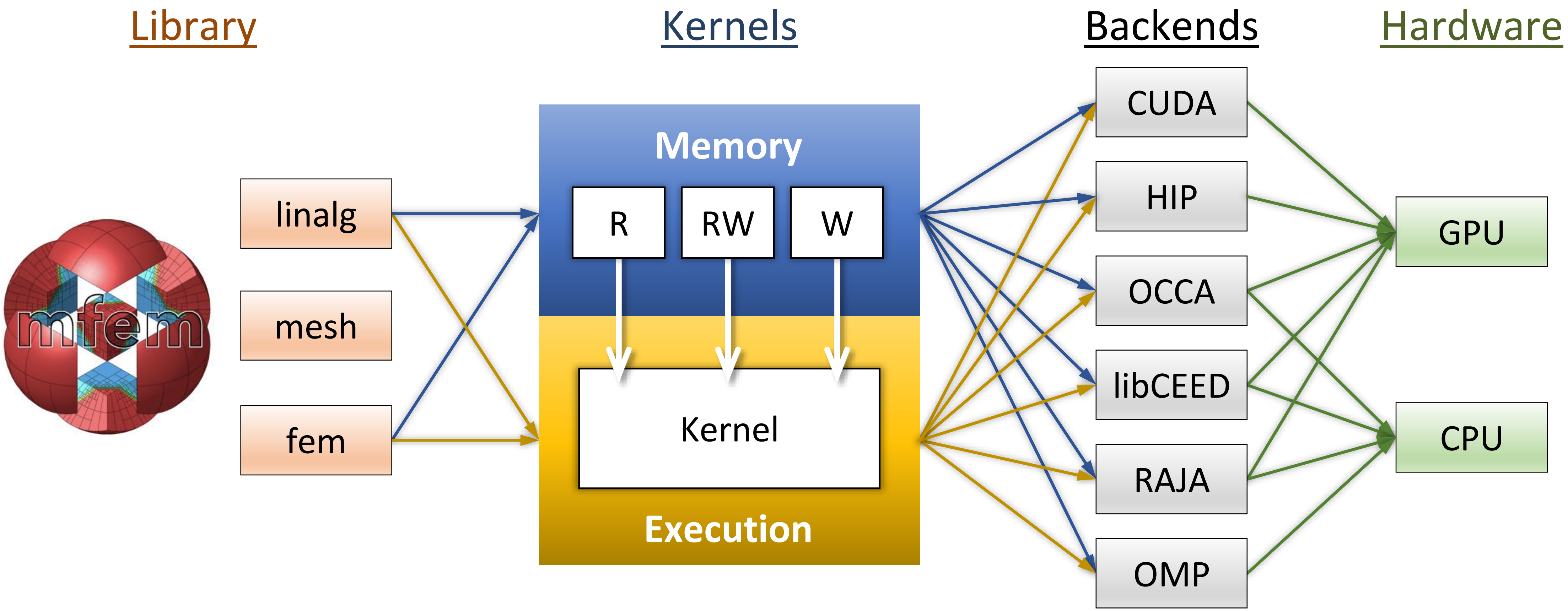
→ ***Internal collaboration + bake-off based on the BPs has been critical!***





Device support in MFEM-4.0

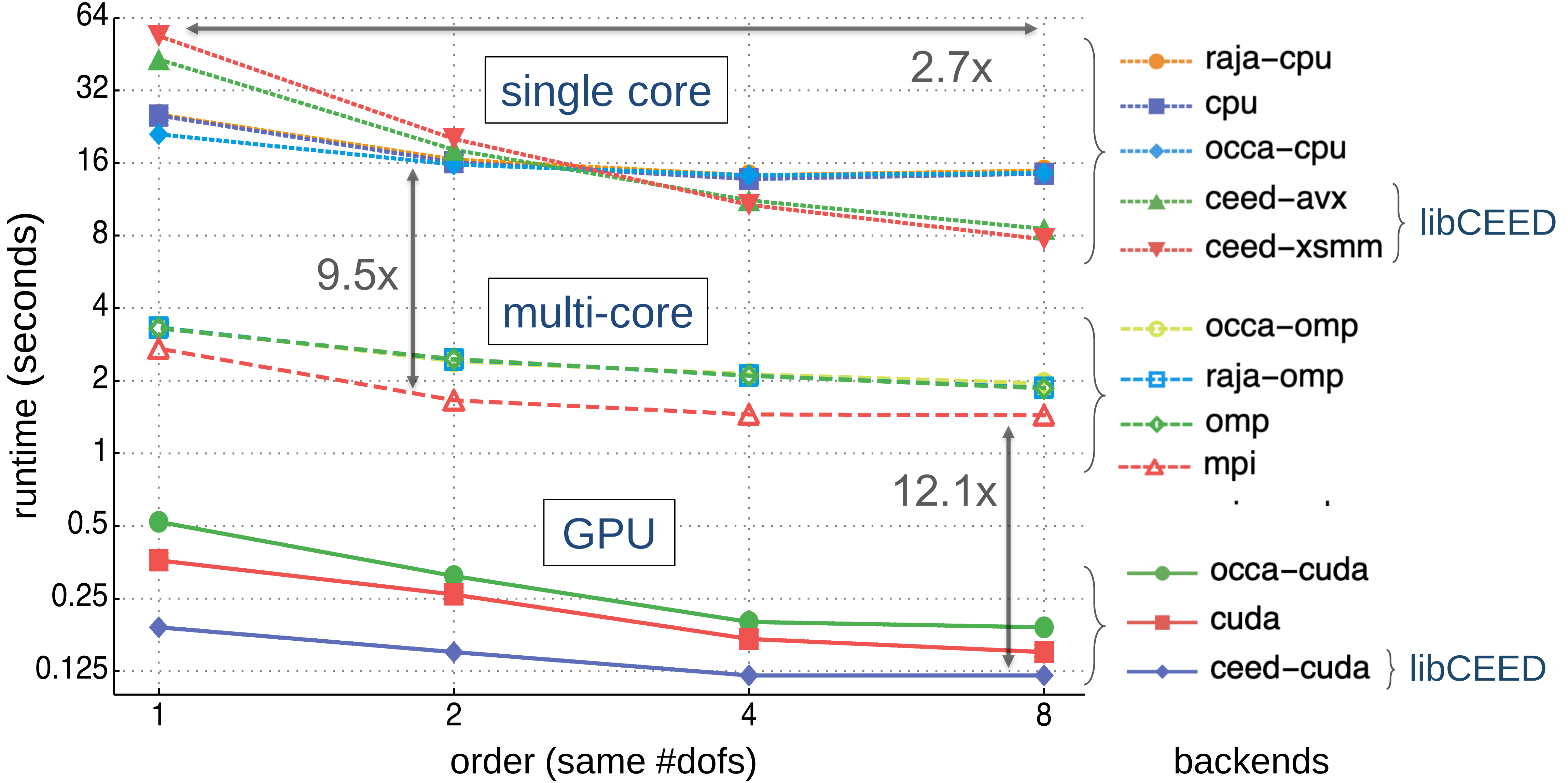
***MFEM-4.0** adds initial GPU support in many linear algebra and finite element operations*



- ✓ Kernels can be specified via loop-body lambda-capture, or raw CUDA, OCCA; many have single source
- ✓ ***Backends are runtime selectable, can be mixed***
- ✓ ***Recent additions: support for AMD/HIP***

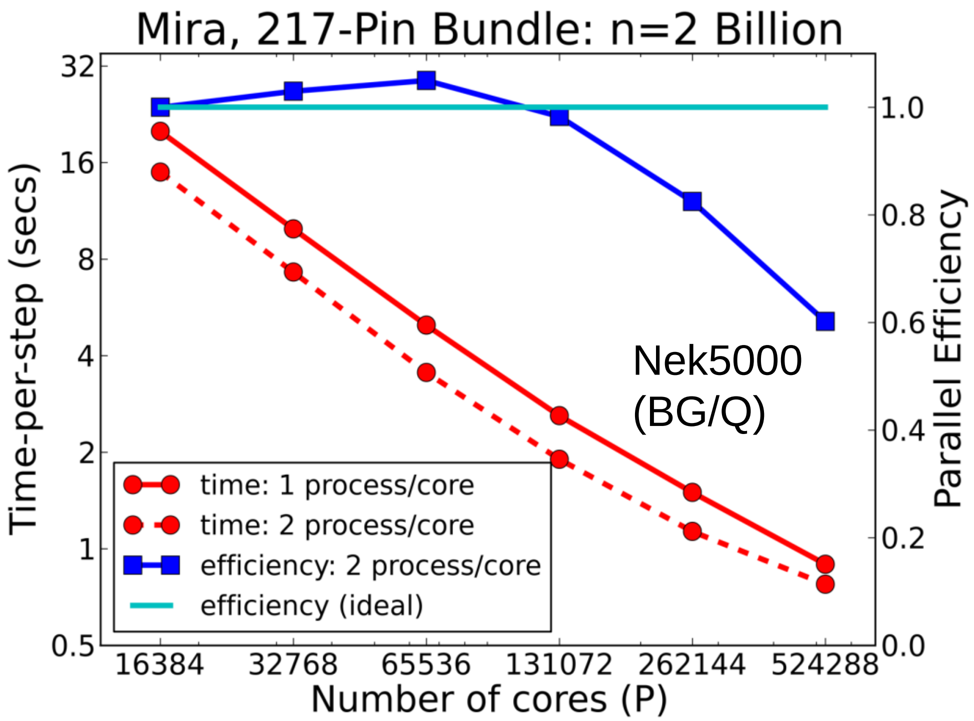
Initial results with MFEM-4.0

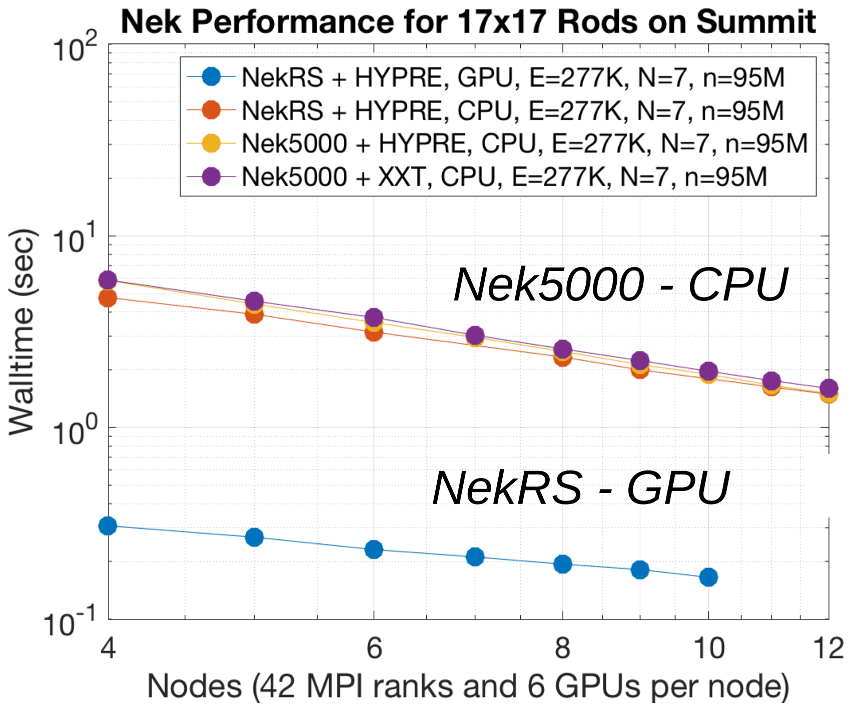
*Example 1, 200 CG-PA iterations, **2D**, 1.3M dofs, GV100 + 16 core Skylake (Linux)*

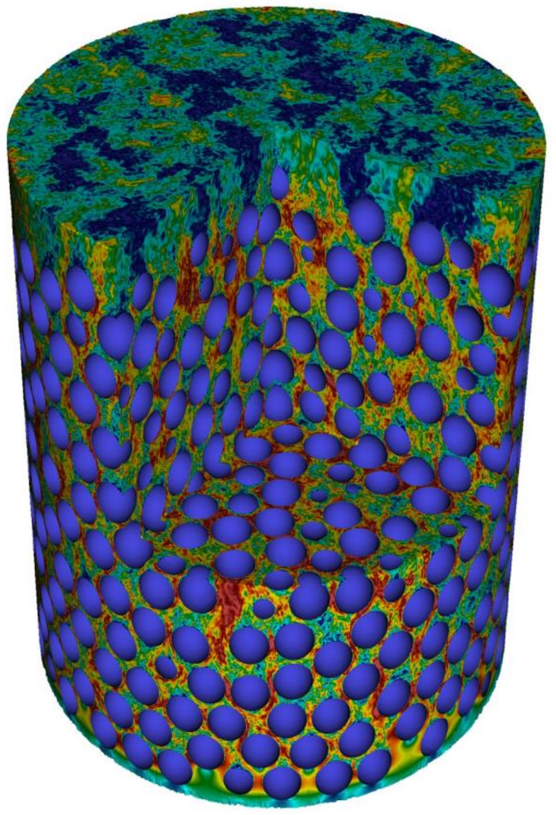


NekRS: GPU-oriented version of Nek5000

- **Nek5000** strong scales (2000 pts/rank) to > 1 million cores **(Gordon Bell winner)**
- **NekRS** based on **OCCA** + **libParanumal** **(Warburton et al., V. Tech.)**
  - Using GPUs, NekRS is about 10-15x faster than all-CPU code on Summit
  - Main kernels get about 2 TFLOPS on Nvidia V100 (FP64)
  - Principal kernels have been tested on AMD and Intel Gen9

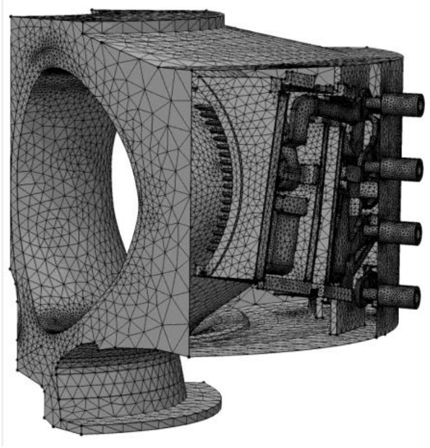




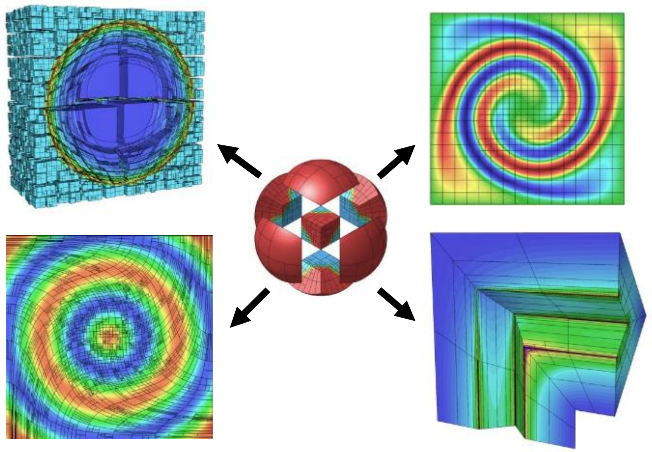


NekRS** turbulence simulation for pebble-bed reactor using **66 GPUs on Summit

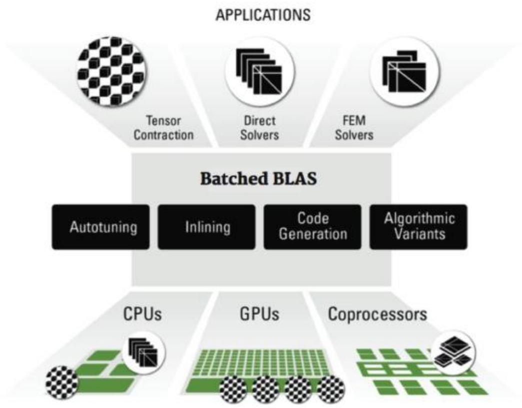
High-Order Software Ecosystem



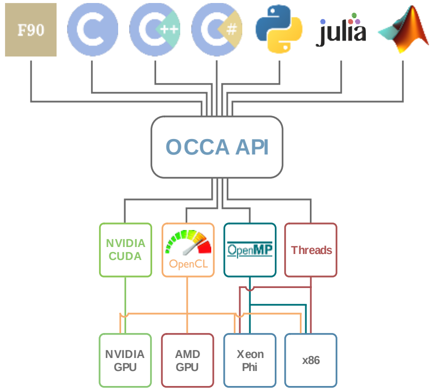
High-order Meshes



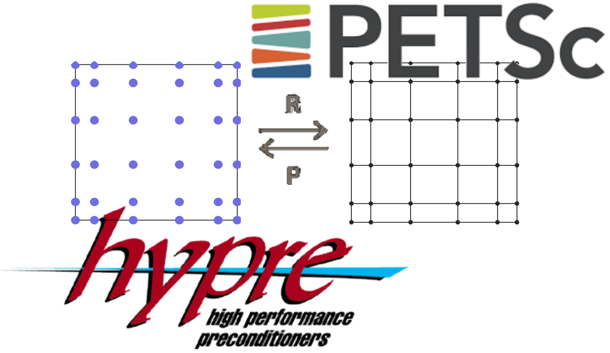
Unstructured AMR



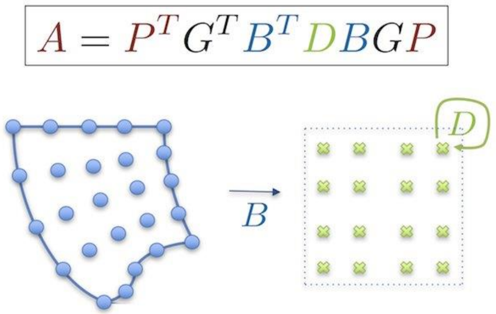
Tensor contractions



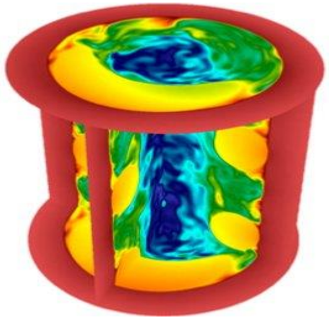
Performance portability



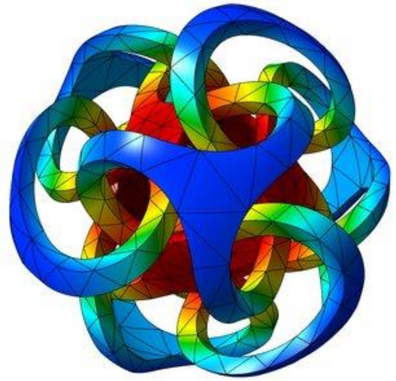
Scalable matrix-free solvers



High-Order Operator Format



General Interpolation

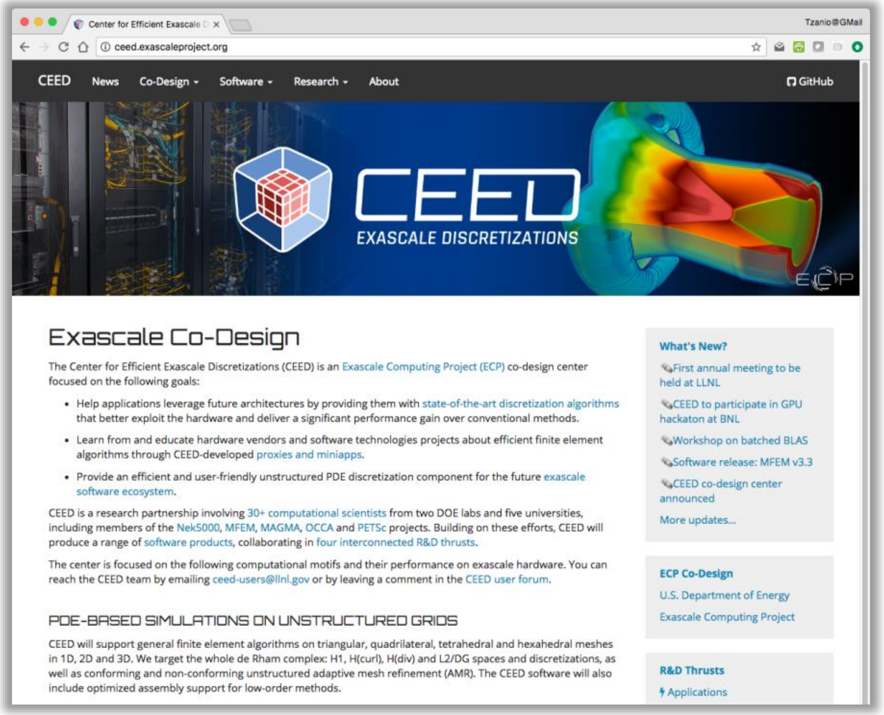


High-Order Visualization

More info at: http://ceed.exascaleproject.org/fe

CEED Resources

- **Website:**
  http://ceed.exascaleproject.org
- **Software:**
  https://github.com/ceed
- **Publications:**
  https://ceed.exascaleproject.org/pubs
- **ECP:**
  https://confluence.exascaleproject.org/display/CEED
- **Email:**
  ceed-support@llnl.gov



- **Applications**
  - **Misun Min**
  - Elia Merzari
  - Vladimir Tomov
  - Robert Rieben
- **Sofware**
  - **Jed Brown**
  - **Tim Warburton**
  - **Mark Shephard**
  - David Medina
- **Hardware**
  - **Stan Tomov**
  - Jean-Sylvain Camier
  - Ian Karlin
  - Scott Parker
- **Finite Elements**
  - **Veselin Dobrev**
  - **Paul Fischer**
  - Tzanio Kolev
  - Panayot Vassilevski

Lawrence Livermore National Laboratory

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.
LLNL-PRES-790865