OOKAMI PROJECT APPLICATION

Date: 2021-12-28

Project Title: libCEED: fast algebra for finite elements

Usage:

- Testbed

Principal Investigator: Jed Brown

- University/Company/Institute: University of Colorado Boulder
- Mailing address including country: Department of Computer Science; 430 UCB; Boulder, CO 80309; USA
- Phone number: 773 234 5332 (cell), 303 492 1592 (office)
- Email: jed.brown@colorado.edu

Names & Email of initial project users:

- Leila Ghaffari, leila.ghaffari@colorado.edu
- Jeremy Thompson, jeremy.thompson@colorado.edu
- Jed Brown, jed.brown@colorado.edu

Usage Description:

libCEED (`https://libceed.readthedocs.io`) is a library offering portable performance for the algebra in finite element methods, especially high order methods. This involves sparse gather/scatter operations, tensor contractions, and vectorized linear and nonlinear operations that define the PDE and boundary conditions that the user wants to solve. libCEED is usable by MFEM, PETSc, and several applications (including via PSAAP and SciDAC). It is implemented in C and offers bindings in Python, Julia, Rust, and Fortran.

libCEED has AVX-specific backends (intrinsics and via libxsmm) as well as CUDA and HIP backends using just-in-time compilation. We also have a crude SVE implementation developed by Leila and Jeremy during the hackathon last month. We would like to further optimize this backend and explore the impact of fixed-length versus agnostic SVE instructions. We will evaluate this impact using benchmarks (CEED BPs) and in applications that rely on libCEED.

Computational Resources:

- Total node hours per year: less than 15000, likely around 1500
- Size (nodes) and duration (hours) for a typical batch job: 1 node for 10 minutes (typical) up to 128 nodes for 10 minutes (scaling study)
- Disk space (home, project, scratch): 100 GB

Personnel Resources (assistance in porting/tuning, or training for your users):

Our users participated in the ARM Hackathon in November and are thus familiar with the system.

Required software:

- C compiler that supports SVE
- MPI
- BLAS/BLIS (can install)
- PETSc (will install development version)
- libCEED (our primary development; will install)

If your research is supported by US federal agencies:

- Agency: DOE
- Grant number(s): Exascale Computing Project (17-SC-20-SC)
