OOKAMI PROJECT APPLICATION

Date:

Project Title:

Porting Octo-Tiger, an astrophysics program simulating the evolution of star systems based on the fast multipole method on adaptive Octrees

Usage:

- Testbed

Principal Investigator:

- University/Company/Institute: Center of Computation & Technology, Louisiana State University
- Mailing address including country: Center for Computation and Technology 1079 Digital Media Center Baton Rouge, LA 70803
- Phone number: 225-284-1710
- Email: pdiehl@cct.lsu.edu

Names & Email of initial project users:

- Hartmut Kaiser, Center of Computation and Technology, Louisiana State University, One the the core developers of HPX (hkaiser@cct.lsu.edu)
- Gregor Daiss, IPVS/SGS, University of Stuttgart, Main developer of the CUDA/AMD kernels in Kokkos (Gregor.Daiss@ipvs.uni-stuttgart.de)
- Dominic Marcelo, Center of Computation and Technology, Louisiana State University, The core developer of Octo-Tiger (dmarce504@gmail.com)
- Kevin Huck, Oregon Advanced Computing Institute for Science and Society, University of Oregon, One of the core developers of APEX the performance measurement tool integrated within HPX (khuck@cs.uoregon.edu)

Usage Description:

The major goal of this application is to port Octo-Tiger to the Arm64FX architecture. We have experience running Octo-Tiger on Power9 on ORNL's Summit and running x86 on CSCS's Piz Daint. We have some preliminary scaling runs for HPX on ARMv7/8 using a small in-house Raspberry Pi cluster. However, due to memory limitation, we could not run Octo-Tiger there. Additionally, we have ported and profiled HPX-enabled DCA++ (dynamical cluster approximation) on ThunderX2 and A64fx nodes on Wombat cluster at ORNL. We intend to publish an SC'22 main track paper with scaling results on this novel architecture. In addition, we would be interested to use the additional AMD hardware, since we recently added the support of AMD GPUs to Octo-Tiger.

Computational Resources:

- Total node hours per year: 10k (With this amount we could port Octo-Tiger to Summit and do the scaling runs to compare to the other architectures last year.) If necessary, 5k could be sufficient to run less scaling tests.
- Size (nodes) and duration (hours) for a typical batch job: Since we perform solely scaling runs, no job will run more than 30 minutes. We would run on up to 150 nodes for a very short time to produce scaling results.
- Disk space (home, project, scratch): 25 GB home; project: 0.5 TB Compilation of the code in release and debug; scratch: 0.5 TB for only scaling runs, we do not need to store large simulations files. All files will be shared between the team in project/scratch.

Personnel Resources:

From our previous experience of porting Octo-Tiger to NERSC's Cori and ORNL's Summit, we only needed assistance using their ticket system.

Required software:

Octo-Tiger compiles with gcc/nvcc or clang/clang-nvcc and depends on: the C++ standard library for parallelism and concurrency (HPX), Silo, HDF5, boost, hwloc, jemalloc/tcmalloc and cmake. For communication, libfabric and MPI are supported. As accelerator devices, CUDA and AMD GPU are supported using Kokkos as abstraction layers.

If your research is supported by US federal agencies:

- Agency: NSF
- Grant number(s): 1814967