OOKAMI PROJECT APPLICATION

Date: January 31, 2021

Project Title: 3D Simulations of Pure Deflagrations of Hybrid White Dwarf Progenitors

Usage: Testbed

Principal Investigator: Alan Calder

- University/Company/Institute: Stony Brook University/Institute for Advanced Computational Science
- Mailing address including country:
  Department of Physics and Astronomy
  Stony Brook University
  Stony Brook, NY 11794-3800
  United States
- Phone number: (631) 632-1176
- Email: alan.calder@stonybrook.edu

Names & Email of initial project users:

Catherine Feldman, catherine.feldman@stonybrook.edu
Eva Siegmann, eva.siegmann@stonybrook.edu
Alan Calder, alan.calder@stonybrook.edu

Usage Description:

Simulating a supernova explosion in 3D requires tremendous computing capabilities and resources, as well as state-of-the-art software. For this project, we will be using the FLASH code, a modular, extensible, highly parallelizable software system designed for multi-scale, multi-physics applications. It allows the combination of different physics modules (gravity, hydrodynamics, radiation transfer, etc.), which are solved on an adaptively refined mesh. Grid blocks are assigned to processors via a Morton space-filling curve, and distribution and communication are handled by MPI.

We will optimize the FLASH code for use on Ookami's A64fx processors. Different compilers and optimizations will be tested to see which combinations can take full advantage of the A64fx architecture. In particular, the code will be compiled with flags that show where use of SVE could lead to a significant speedup. MPI task mapping will be tuned to the most efficient distribution of memory that uses A64fx's NUMA feature to optimize both intra-node and inter-node communication. Additionally, FLASH has the ability to run on larger numbers of nodes as the grid refines and the problem size increases, and to control how much memory is used by each processor.

Once these steps are complete, we will know the optimal compilers and compiler flags, task mapping, memory per node, and number of cores to run our simulation on at different stages of its refinement. If this optimized code is correct and faster than that run on other architectures, a production project request will be submitted.

Computational Resources:

- Total node hours per year: 2-5 nodes * 1 hour/node * 300 runs = 600 - 1500 node hours
- Size (nodes) and duration (hours) for a typical batch job: 2-5 nodes for 1 hour
- Disk space (home, project, scratch):
  - Home: 20GB for analysis, visualization, and batch scripts
  - Project: 5 TB for important results from the 300 runs
  - Scratch: 20 TB for the code and output data for 300 runs

Personnel Resources:

None

Required software:

1. The FLASH code, which we will provide (http://flash.uchicago.edu, Modifications for CO and CONe WDs are available from http://astronomy.ua.edu/townsley/code)
2. HDF5
3. MPI
4. Python and the following packages (which will be installed on our user accounts) for data analysis and visualization
   - yt: https://yt-project.org/
   - h5py: https://www.h5py.org/
   - mpi4py: https://pypi.org/project/mpi4py/

If your research is supported by US federal agencies:

- Agency: US Department of Energy, National Science Foundation
- Grant number: DE-FG02-87ER40317; 1927880

Production projects:

Production projects should provide an additional 1-2 pages of documentation about how

1. the code has been tuned to perform well on A64FX (ideally including benchmark data comparing performance with other architectures such as x86 or GPUs)
2. it can make effective use of the key A64FX architectural features (notably SVE, the high-bandwidth memory, and NUMA characteristics)
3. it can accomplish the scientific objectives within the available 32 Gbyte memory per node