DOE Office of Science INCITE Project:
Extreme-scale Simulation of Supernovae and Magnetars from Realistic Progenitors
2019 Q3 Report

Principal Investigator:
Sean M. Couch
Michigan State University

Co-Investigators:
Andrew Christlieb (Michigan State University)
Evan O'Connor (Stockholm University)
Kuo-Chuan Pan (National Tsinghua University)
Luke Roberts (Michigan State University)
MacKenzie Warren (Michigan State University)

October 1, 2019

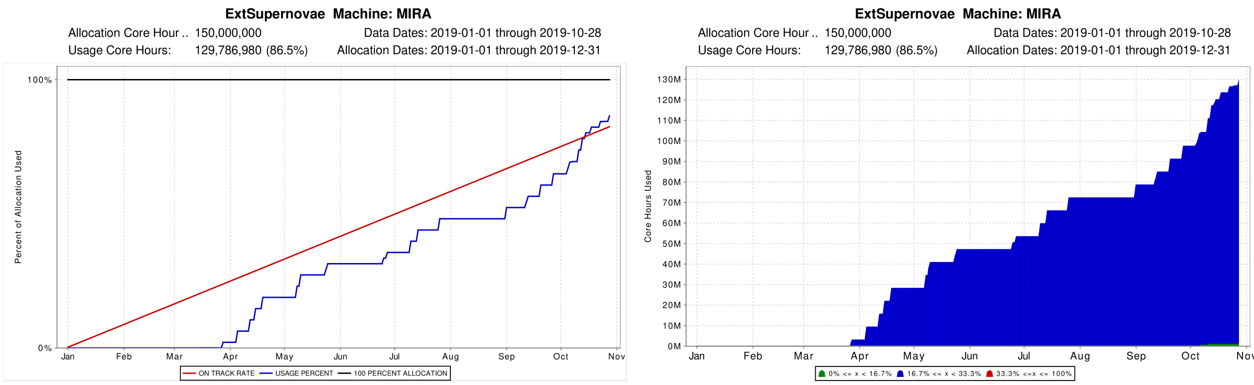

Figure 1: Allocation usage.

1 Project Usage

So far in 2019 we expended 129.8M core-hours on Mira out of our total 2019 allocation of 150M core-hours (86.5% usage). This is just slightly ahead of the linear usage curve. Our burn rate in Q3 was substantially larger than for Q2. We are now running two of our primary simulation milestones in the Capability queue and will start a second Capability-scale simulation in the next week or two. Figure 1 we show our current usage and categorized hours on Mira.

After completely code tuning, and overcoming an issues with constructing a new set of initial conditions, we have begun production simulations on Theta in Q3. In just a couple weeks of production, we have expended 2.2M core-hours on Theta (12.5% of total allocation) and the simulation has already grown to the Capability scale. Based on our experience last year running on Theta, we do not anticipate any difficulty in expending our entire allocation before the end of the calendar year.

2 Report on Project Milestones

Our milestones for Year 2, and corresponding progress, were:

1. Long time simulations of MHD CCSNe - These simulations have been restarted from simulations carried over from 2018 and are running in the Capability queue. Substantial progress has been made on these simulations in Q3 and they are nearing completion.

2. High-resolution simulation of MHD dynamos in the proto-neutron star - So far in Q1/Q2 we have analyzed simulations from 2018 that will serves as the initial conditions for this high-resolution simulation. We have started this simulation in Q3. Given the extreme resolution of this simulation, it will run in the Capability queue from the outset.

3. MHD simulation of CCSN progenitors - These simulations will be started in Q2. We are tuning our progenitor application to make better use of OpenMP threading.

4. CCSN simulation with 3D progenitors - This simulation is now running at Capability scale on Theta.

5. Implement microphysics from TEAMS SciDAC collaboration and neutrino-electron scattering (NES) - the TEAMS microphysics package is not yet ready for production simulations. We have during Q1 finished an implementation of NES and are now using it in production on Theta.

3 Project Productivity

3.1 Primary

Publications

- "Simulating Turbulence-aided Neutrino-driven Core-collapse Supernova Explosions in One Dimension", Couch, S. M., Warren, M. L., O'Connor, E. P. 2019, *arXiv e-prints*, arXiv:1902.01340

- "Features of Accretion Phase Gravitational Wave Emission from Two-dimensional Rotating Core-Collapse Supernovae", Pajkos, M. A., Couch, S. M., Pan, K., O'Connor, E. P. 2019, *Astrophysical Journal*, 878, 13

Presentations

- "Gravitational Waves from Core-collapse Supernovae," S.M. Couch, LIGO SN Group Seminar, March 2019

- "Multidimensionl Supernova Progenitors," S.M. Couch, SciDAC TEAMS Collaboration Meeting, May 2019

- "Predicting Supernova Neutrino Signals," M. Warren, Supernova Early Warning System (SNEWS) 2.0 Workshop, June 2019

- "High-order MHD for Supernovae," S.M. Couch, ASTRONUM 2019, July 2019

4 Center Feedback

Our catalyst, Adrian Pope, has been extremely helpful. He is now helping us tune our code for Theta.

5 Code Description and Characterization

`FLASH` is a highly capable, fully modular, extensible, community code that is widely used in astrophysics, cosmology, fluid dynamics, and plasma physics, and other fields. The capabilities of the FLASH code include adaptive mesh refinement (AMR), several self-gravity solvers, an advection-diffusion-reaction (ADR) flame model, an accurate and detailed treatment of nuclear burning, and a sophisticated two-moment neutrino transport scheme based on an explicit hyperbolic solver. The neutrino interactions are included through the open-source neutrino interaction library `NuLib`. We have enhanced the performance of the two-moment neutrino transport scheme significantly as well as upgraded the transport to now include full velocity and gravitational red-shift dependence in the evolution equations.

`FLASH` is written in modern Fortran, with some utility functions written in C, and a build system written in Python. It requires MPI library support, and either HDF5 or P-NetCDF for I/O. Additional mathematical software, such as `Hypre`, may be required to configure `FLASH` for particular simulations.

Algorithm classes used within `FLASH` include Sparse Linear Algebra solvers, FFT, active and passive particles, structured grids, and AMR.