DOE Office of Science INCITE Project:
Extreme-scale Simulation of Supernovae and Magnetars from Realistic Progenitors
2018 Q2 Report

Principal Investigator:
Sean M. Couch
Michigan State University

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

July 1, 2018

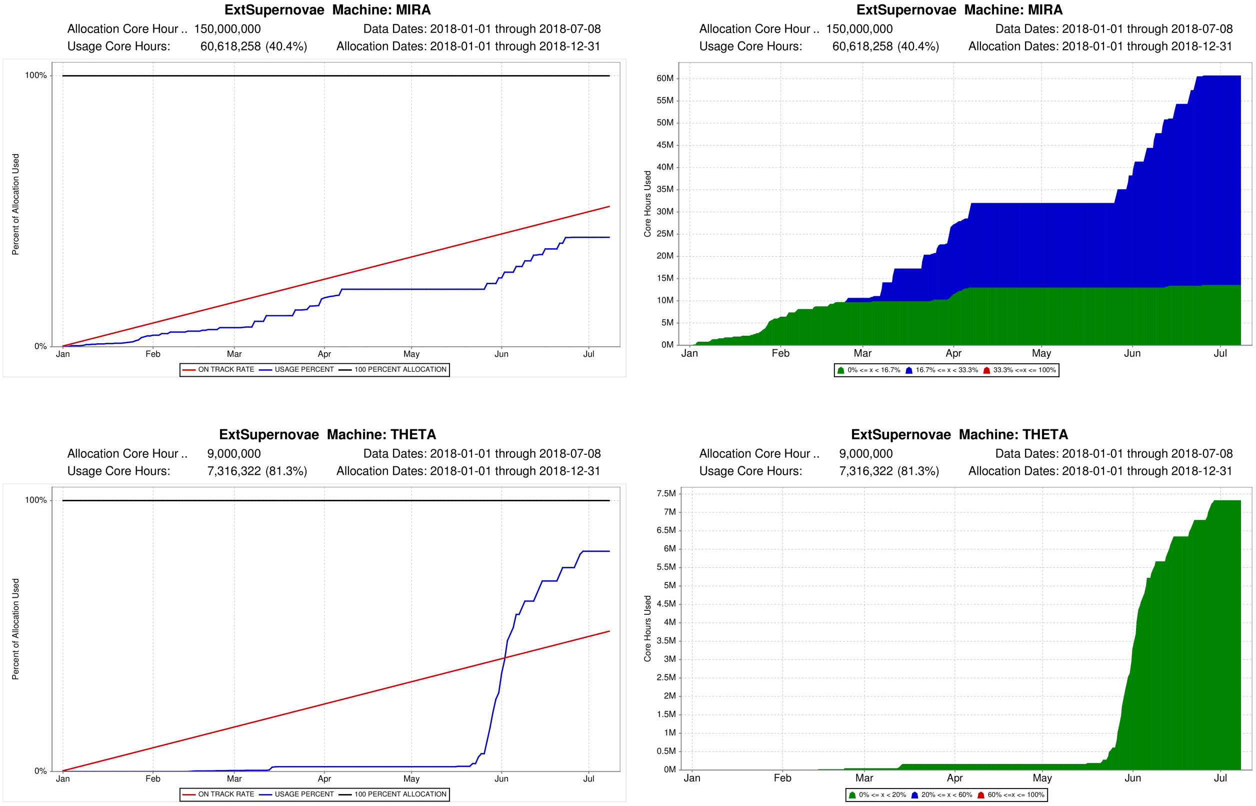

Figure 1: Allocation usage.

1 Project Usage

Thus far, we have expended 60.6M core-hours on Mira, 40.4% of our 2018 allocation (see Figure 1). We are slightly behind the ideal usage curve, but in Q3 we will commence production 3D progenitor simulations, adding a second capability-scale job to our workflow. Overall, we anticipate catching up quickly to the on-track usage curve. The vast majority of our usage on Mira has been at the capability level.

We have so far expended 7.3M core-hours on Theta, 81.3% of our 2018 allocation. At beginning of our allocation, we had technical issues with the Intel fortran compiler on Theta when enabling OpenMP. After a few weeks of investigation, we decided to switch to GNU fortran compiler. After that, we moved one high-resolution production simulation from Mira to Theta and tuned our application on Theta to continue that simulation. A weak scaling test of restarting that simulation is shown in Figure 2. Figure 2 suggests that the performance is getting worse when using more than 300 nodes on Theta. However, we noticed that after the maintenance on May 21, 2018, the wall-clock limit on Theta is limited to 6 hours with > 256 nodes and 9 hours with > 384 nodes. Since our simulations require more than 50 of restarts, this limitation made us difficult to monitor our simulation. Therefore, we developed a simply automation job scheduling script to re-submit our job on Theta.

Recently, we have encountered another technical issue, where memory fragmentation happened randomly. This might be a memory leak or simply insufficient memory but we don't observe this

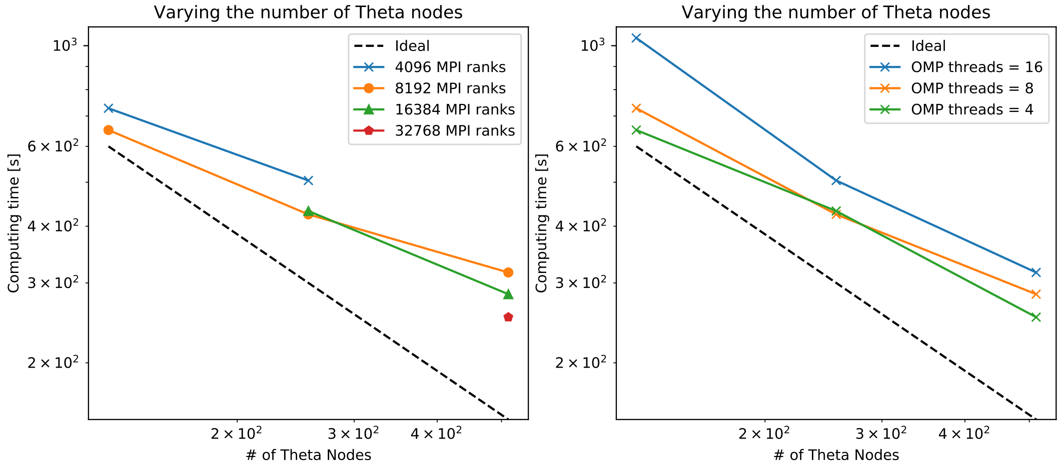

Figure 2: Weak scaling test on Theta. different colors represent different combinations of MPI and OpenMP tasks.

issue on Mira. We are still investigating this issue and this issue break our job scheduling script. The production run is still running but the progress become slower due to this issue. Currently, we have spend 7M core-hours on Theta, which is 78% of our allocation.

2 Report on Project Milestones

Our milestones for Year 1, and corresponding progress, are:

1. *3D simulations of magnetorotational core-collapse supernovae* – These simulations have been started and are running at capability on Mira. The simulations have progressed sufficiently that we are beginning to see the impact of including rotation and magnetic fields. These simulations are on track to meet our scientific goals for 2018. The simulations on Theta involved rapidly-rotating initial conditions and have moved very quickly, thanks to our greater computational efficiency on Theta and higher queue throughput.

2. *3D simulations of iron core collapse in massive stars* – In Q1 and Q2, we have optimized our simulation setup for Mira and have run preliminary simulations in reduced geometry to determine the optimal parameters and initial conditions. We plan to commence production runs in Q3.

3. *High-resolution simulations of magnetorotational turbulence* – These simulations will be started in Q3.

4. *Develop SIMpliPy workflow tool* – Work on this continues. We have two summer undergraduate REU students in the group that are working on this tool.

5. *Implement marching cubes for EOS and opacities* – This will begin Q3.

3 Project Productivity

3.1 Primary

Publications

- “Turbulence in Core-Collapse Supernovae”, Radice, D., Abdikamalov, E., Ott, C. D., Moesta, P., Couch, S. M., Roberts, L. F. 2018, *Journal of Physics G*, 45, 053003 (3 citations)
- “Equation of State Dependent Dynamics and Multi-messenger Signals from Stellar-mass Black Hole Formation”, Pan, K., Liebendörfer, M., Couch, S. M., Thielemann, F. 2018, *The Astrophysical Journal*, 857, 13 (4 citations)

Presentations

- “Understanding Massive Stellar Death: Predictive Simulation of Core-collapse Supernovae,” S.M. Couch, CCAPP Seminar, Ohio State University, Columbus, OH, April 2018
- “Understanding Massive Stellar Death: Predictive Simulation of Core-collapse Supernovae,” S.M. Couch, Physics and Astronomy Colloquium, Louisiana State University, Baton Rouge, LA, March 2018
- “Understanding Massive Stellar Death: Predictive Simulation of Core-collapse Supernovae,” S.M. Couch, Physics and Astronomy Colloquium, University of Alabama, Tuscaloosa, AL, February 2018

3.1.1 Secondary

- Co-I and postdoc Kuo-Chuan Pan will start a faculty position at National Tsing Hua University in Taiwan this summer.
- Co-I and postdoc MacKenzie Warren has won a prestigious NSF Postdoctoral Fellowship.

4 Center Feedback

Our catalyst, Adrian Pope, has been extremely helpful. We have been experience occasional, seemingly random, I/O errors when reading large checkpoint files at startup. He is helping us debug this, but the error is difficult to reproduce reliably. This has slightly slowed progress, but not ground it to a halt.

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`. During Year 2 of this allocation we enhanced the performance of the two-moment neutrino transport scheme significantly as well upgrade 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.