Scaling Information for
"Four Projects in Astrophysical Magnetohydrodynamics"

David Collins, PI

The four projects presented in this proposal will be using the code Enzo (Bryan et al. 2014; Collins et al. 2010). Enzo is an open source adaptive mesh refinement (AMR) code that has been used in hundreds of astrophysical works. These studies include the formation of the first stars (Abel et al. 2002), clusters of galaxies (Xu et al. 2011) and the large scale structure of the universe. It employs several hydrodynamics solvers, including the piecewise parabolic method (PPM, Colella & Woodward 1984), two implementations of magnetohydrodynamics (MHD), self-gravity, and Lagrangian particles that can be used for collisionless dark matter, stars, dust, and passive tracers. One of the primary advantages of Enzo over other codes is its use of structured AMR, which allows it to add resolution elements adaptively as dictated by the problem. A variety of refinement criterion are available. The present studies will use the divergence-preserving MHD module (Collins et al. 2010). For the patch solver we use the second order MHD method of Li et al. (2008) and the constrained transport method of Gardiner & Stone (2005) to preserve the divergence-free constraint ($\nabla \cdot \mathbf{B} = 0$) to machine precision. For the AMR, the divergence-free reconstruction of Balsara (2001) is used to interface magnetic fields with the adaptive mesh. For chemistry and radiative cooling used in the *cores* project, Grackle is used (Smith et al. 2017).

To measure the behavior of the solvers, we ran a weak scaling test with the main physics packages for the four projects. The four projects are of two varieties: the *foregrounds* and *turbulence* projects employ driven turbulence and the hydro/MHD solver, while the *cores* and *galaxies* projects additionally employ the gravity solver. Thus we run two scaling studies, one with just the hydro and one with hydro, gravity, and AMR. For both studies, a constant amount of work, 128^3 zones per task, was used for each node. Scaling was done from 8 through 4096 processors, with 64 threads per node (when possible). For the AMR, one level covering 1/8 of the box by volume was used. The packages in question do not depend heavily on the regime of physics in question, so uniform gas was taken in each case. The results can be seen in Figure 1. Here we plot $\zeta = \frac{zone\ updates}{core\ second}$ vs. number of mpi tasks. For ideal scaling, this will be independent of the number of processes.

The blue curve, applicable to the *foregrounds* and *turbulence* suites, contains only the MHD solver and random forcing. This is extremely parallelizable, as the work is entirely local. The orange curve, applicable to the *cores* and *galaxies* projects, uses the MHD solver, gravity solver, and AMR. The performance of this combination sharply declines at 512 threads. This is due to the gravity solver and AMR overhead.

We use a value of $\zeta = 10^5$ for the *foregrounds* and *turbulence* simulations, and 7.4×10^4 for the *cores* and *galaxies* simulations to estimate the total cost for each suite of simulations.

References

Abel, T., Bryan, G. L., & Norman, M. L. 2002, Science, 295, 93

Balsara, D. S. 2001, J. Comput. Phys, 174, 614

Bryan, G. L., Norman, M. L., O'Shea, B. W., Abel, T., Wise, J. H., Turk, M. J., Reynolds, D. R., Collins, D. C., Wang, P., Skillman, S. W., Smith, B., Harkness, R. P., Bordner, J., Kim, J.-h., Kuhlen, M., Xu, H., Goldbaum, N., Hummels, C., Kritsuk, A. G., Tasker, E., Skory, S., Simpson, C. M., Hahn, O., Oishi, J. S., So, G. C., Zhao, F., Cen, R., Li, Y., & Enzo Collaboration. 2014, ApJS, 211, 19

Colella, P. & Woodward, P. R. 1984, J. Comput. Phys, 54, 174

Collins, D. C., Xu, H., Norman, M. L., Li, H., & Li, S. 2010, ApJS, 186, 308

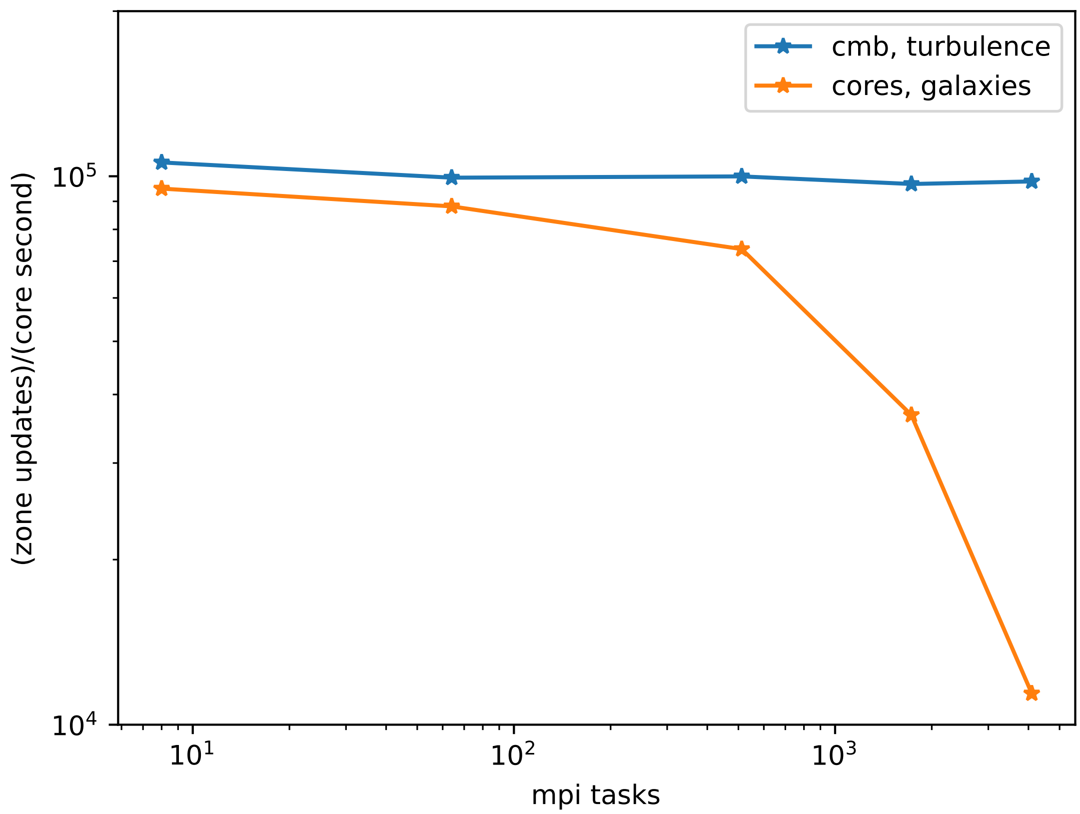

Figure 1: Updates per core-second for weak scaling on Stampede 2. The blue curve only employs the MHD solver, in the configuration used for the *foregrounds* and *turbulence* simulations. The orange curve employs the MHD solver as well as the gravity solver and AMR, and will be used for the *cores* and *galaxies* simulations. The gravity and AMR degrade the performance above 512 threads. This study used 64 cores per node when possible.

Gardiner, T. A. & Stone, J. M. 2005, J. Comput. Phys, 205, 509

Li, S., Li, H., & Cen, R. 2008, ApJS, 174, 1

Smith, B. D., Bryan, G. L., Glover, S. C. O., Goldbaum, N. J., Turk, M. J., Regan, J., Wise, J. H., Schive, H.-Y., Abel, T., Emerick, A., O'Shea, B. W., Anninos, P., Hummels, C. B., & Khochfar, S. 2017, MNRAS, 466, 2217

Xu, H., Li, H., Collins, D. C., Li, S., & Norman, M. L. 2011, ApJ, 739, 77