The Dust Between Us and the Big Bang: Scaling Study

PI: David C. Collins (Florida State University)

We will perform two suites of simulations; one will be a pair of high resolution 2048^3 simulations of plasma turbulence, and the other will be three high resolution AMR simulations of isolated galaxies. There will be many things to measure from these simulations, our first focus is to study the observed polarization produced by plasma turbulence in the galaxy. The study of this dusty, magnetized turbulence will allow us to remove the polarization signal due to our own galaxy from the polarization of the cosmic microwave background (CMB). This, in turn, will allow us to see gravitational waves from the big bang.

The code we will use is Enzo (Bryan et al. 2014; Collins et al. 2010). This is an open source adaptive mesh refinement (AMR) code that has been used for many astrophysical applications on the world's largest computers. Enzo uses higher order Godunov methods to solve the MHD equations, fourier transform based methods for gravity, and the algorithm of Berger & Colella (1989) for the AMR. The code also includes chemistry and star particles, but their cost is negligible compared to the MHD, Gravity, and AMR. Here we will profile the performance of the three primary systems in the manner they will be used in the proposed simulations.

The cost of each of the three systems is independent of the state of the plasma, so we simulate boxes with uniform density and temperature for simplicity. We perform three suites of study. For the MHD and Gravity study, single resolution boxes are used. For the AMR study, we use the same constant work-per-level that will be employed in our galaxy simulations; the central 1/8th of the box is refined by a factor of 2. While the proposed simulations will use 8 levels of refinement, one level is enough to engage the AMR machinery that keeps the information consistent between levels. We have additionally tested the full 8 level configuration at scale, and it performs exactly as expected from the scaling.

We estimate the cost for our simulations as

$$SU = t_{wall} N_N \tag{1}$$

$$t_{wall} = \frac{\sum_\ell N_Z N_U}{\zeta} \frac{1}{N_C}, \tag{2}$$

where N_N is the number of nodes, N_Z is the number of zones per level, N_U is the number of updates per level, N_C is the number of cores, and most importantly ζ is the cost in (zone-updates)/(core-second.) Given perfect scaling, ζ is independent of the number of cores. We model the actual performance by measuring ζ as we increase N_Z. By targeting the same configuration we will use for the production simulations, this is an accurate estimate of the ultimate cost.

The result of the scaling study can be found in Figure 1, which shows the performance, ζ, vs. number of cores, N_C. For each simulation, we take one root grid timestep, which includes the AMR steps in the AMR suite, then divide by the time. Here we perform simulations with fixed work of 64^3 zones per core, in three suites of simulations. The AMR suite has two grids of 64^3 per core. We use $256, 512, 1024, 1536,$ and 2048 zones per side for all three suites, with the AMR suite including a second level of the same number of zones. We use 56 cores per node throughout. As the number of cores is not an exact multiple of 56, one node has a few idle cores, but this is not enough to impact the scaling.

The blue curve in Figure 1 shows the scaling of the MHD solver. This component scales with the number of zones simply as N_Z, so as we increase both in concert we see very little drop in performance. ζ=4.4$\times 10^5$ for 10 nodes, and $\zeta = 4.1 \times 10^5$ for 586 nodes, less than 10% decrease in performance for a factor of 64 in core count. It is this outstanding scaling that enables us to perform such large simulations.

The orange curve in Figure 1 employs the MHD solver and the gravity solver. The cost of the FFT based gravity sovler increases as $N_Z \ln N_Z$, so the performance, ζ, suffers from the increased cost. Additionally, the FFT is done on pencils spaning the domain, but the data is stored in cubes. This rearangement of data also costs in the scaling.

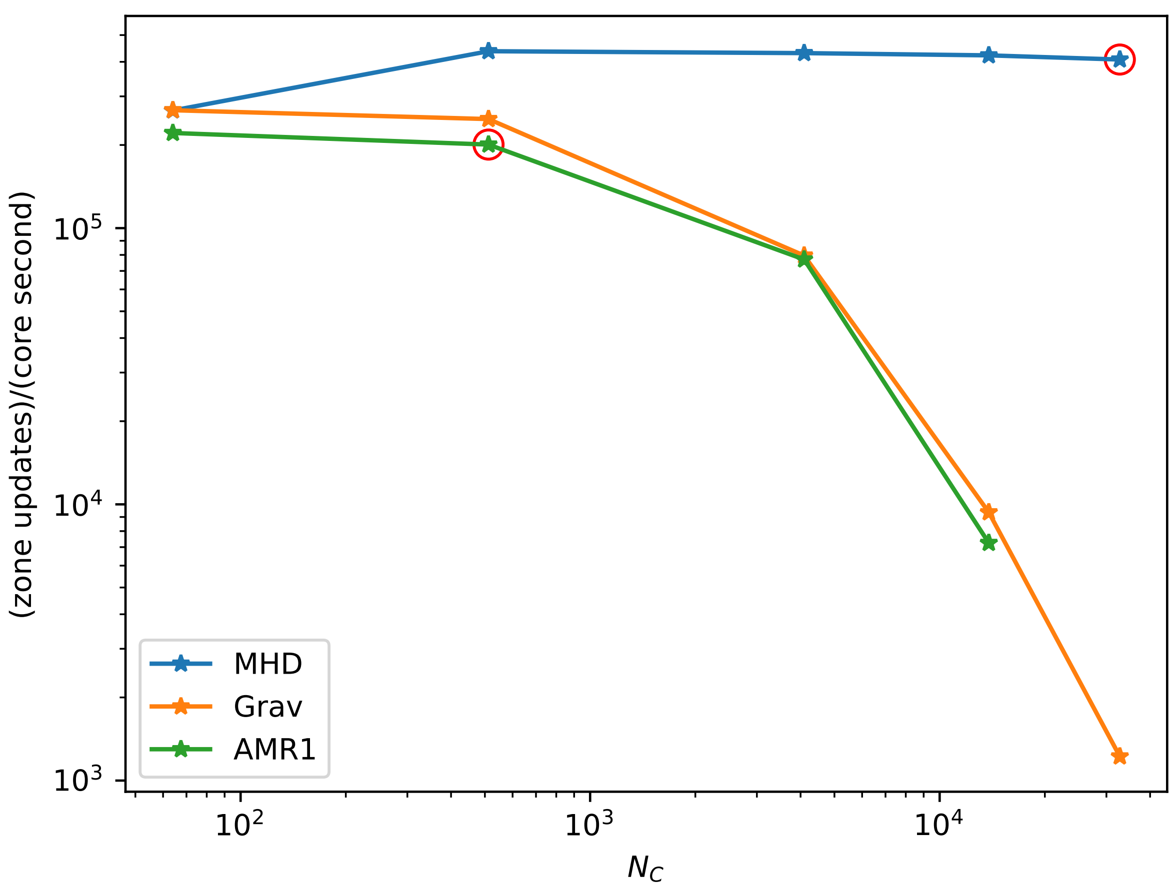

Figure 1: The performance, ζ vs. number of cores, N_C, for Enzo in three configurations. The MHD solver alone (blue curve) scales quite well. The orange curve shows MHD and the gravity solver, while the green curve shows the MHD solver, gravity solver, and AMR overhead. Red circles show the target simulation configuration.

Table 1: The number of nodes, N_N, zones, N_Z, and cores, N_C used for each node in our scaling study.

N_N	N_Z	N_C
2	256^3	64
10	512^3	512
74	1024^3	4096
274	1536^3	13,824
586	2048^3	32,768

The AMR suite can be seen in the green curve in Figure 1. This suite employs the MHD solver, gravity solver, and AMR overhead, which represents the performance of the galaxy solver. The additional cost above the gravity can be seen to be small on all runs. When tring to run 2048^3 with one level of additional 2048^3, the code threw a segmentation violaton. This was not tracked down for this study due to the fact that we are quite far from being ready to run such a simulation.

The red circles show ζ for the target configurations. The turbulence simulations will run 2048^3 zones on 586 nodes, with a performance of $\zeta = 4 \times 10^5$. The AMR simulations will be run at roughly 512^3 per level for 8 levels.

References

Berger, M. J. & Colella, P. 1989, J. Comput. Phys, 82, 64

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

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