Code Performance of AstroBEAR2.0

Adam Frank, Jonathan J. Carroll-Nellenback, Baowei Liu,Luke Chamandy

October 15, 2016

1 Abstract

Strong and weak scaling behavior of the current AstroBEAR2.0 code were tested on Stampede of TACC and Comet of SDSC (strong scaling only). To have an accurate estimate the resource and optimal core numbers for our production runs, we use the same multiphysics features (with self-gravity, radiation transfer and HYPRE library) as our production runs when running the scaling tests. For the strong scaling test with the Model 7 of the proposed production runs,we got good scaling result both on Comet and Stampede (with slope -0.78 on 1728 cores of Comet, -0.86 and -0.73 on 1024 and 4096 cores of Stampede respectively while the perfect scaling slope is -1). For the weak scaling with fixed grid, we found the running efficiency drops about 17% for up to 1024 cores on Stampede and 27% on 4096 cores due to the communication increase between the processors. The size of our typical run jobs will be 1024 cores on Stampede or 1200 cores on Comet.

2 Scaling Test Results

To understand how the code behaves on the machines we apply for and especially to estimate on how much resources we will need for our project, we choose one of our production runs (Model 5) and use the same resolution but for much shorter final time. We also restart from a point where the disk forms around the secondary (Figure 1) to get an even better estimate.

Figure (2) shows the strong scaling test result of AstroBEAR on Comet and Stampede. The strong scaling tests are done with the same resolution our proposed production runs ($80 \times 80 \times 48$ + 7 levels AMR) but for much shorter final time (1/10 of one frame while each of the production runs needs 500 frames).

The current code with the Model 5 of proposed production runs shows very good scaling on Comet (with slope=-0.78 while the perfect scaling has a slope -1) up to 1728 cores or 72 nodes. On Stampede the behavior is even better up to 1024 cores (with slope=-0.86) and the behavior drops only a little when running on 4096 cores (with slope=-0.73). We plan to run our simulations on 1200 cores of Comet and 1024 cores of Stampede but can also run on more cores

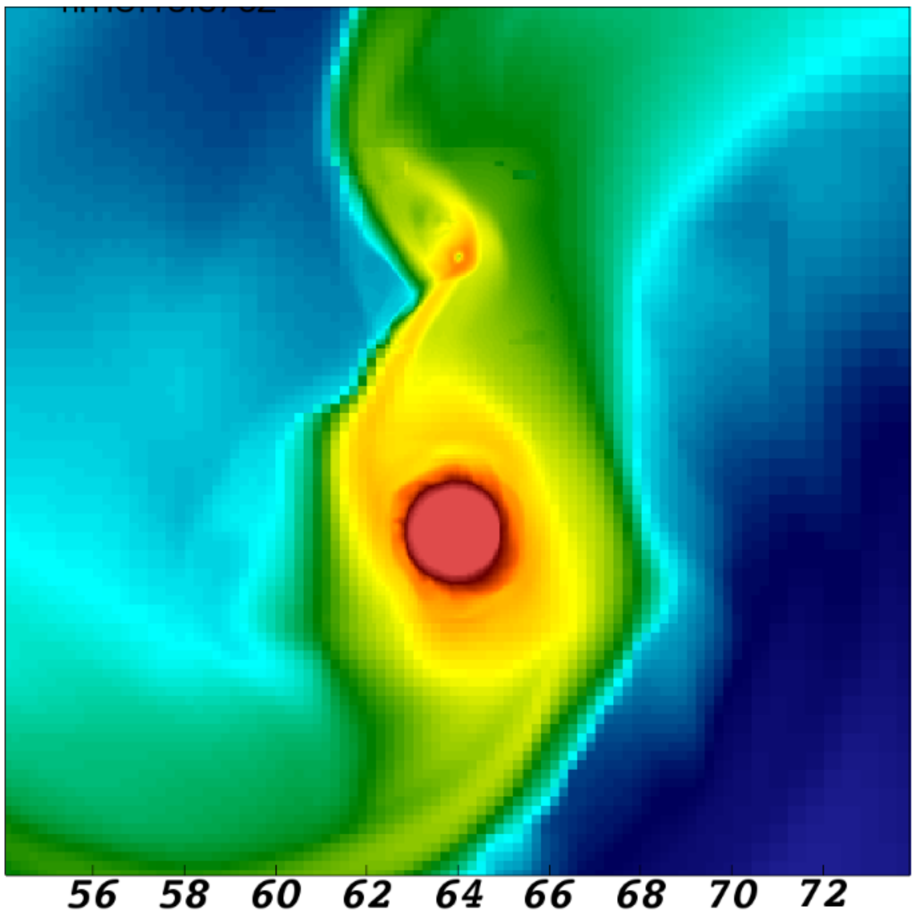

Figure 1: The start point of strong scaling test for AstroBEAR. It shows a disk forming around the secondary. This start point has the same base dimension as our proposed runs and 5 levels of AMR. In our scaling tests, we restart from this frame and run with 7 levels of AMR for a short time (1/10 of one frame). We also propose to run at 7 levels of AMR for our production runs.

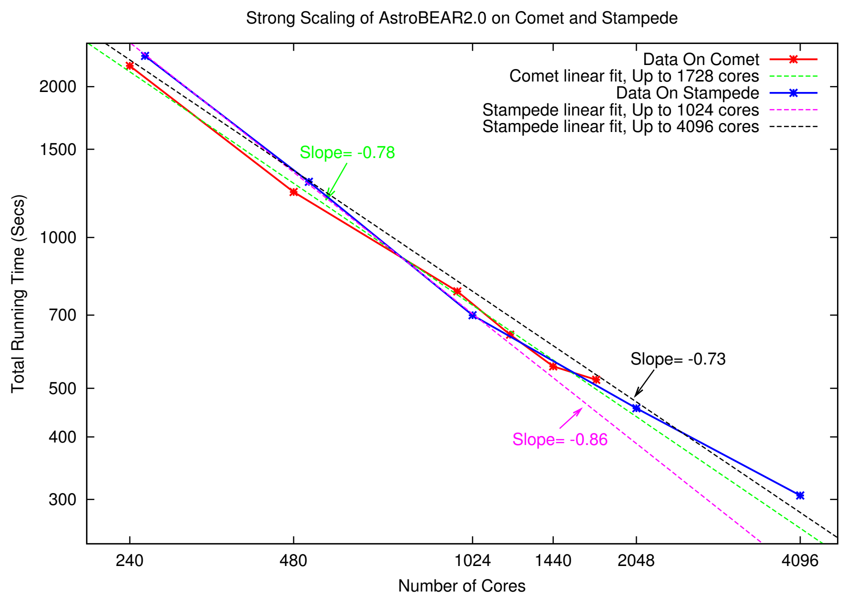

Figure 2: Current strong scaling behavior for AstroBEAR on Comet and Stampede. For these tests we run with with the same resolution as Model 5 in our proposed production runs ($80 \times 80 \times 40$ + 7 levels AMR), but for shorter final time (1/10 of one frame while the production runs need 500 frames). The current code with the Binary modules shows very good scaling on Comet (with slope equals to -0.78 up to 1728 cores) and Stampede (with slope equals to -0.86 up to 1024 cores and -0.73 up to 4096 cores). Perfect scaling has a slope -1. We are planning to run our productions runs on 1024 cores of Stampede and 1200 cores of Comet to gain the best SU-efficiency. The SUs we request are based on the estimate time on 1024 cores of Stampede.

to save the walltime. The SUs we request are based on the estimates on 1024 cores of Stampede (See Table 1 and Table 2 in the main document).

In Figure (3) we show the weak scaling test result of AstroBEAR on Stampede. In these tests we use fixed grid to make the number of cells on each core to be constant (64^3 per core). Weak scaling is good to test data communication part of the code since the work load on each processor are about same. We run these tests with self-gravity,radiation transfer and HYPRE library. The runtime efficiency of AstroBEAR drops only 17% on up to 1096 processors and 27% up to 4096 processors which shows a very good weak scaling considering the multiphysics features in these runs.

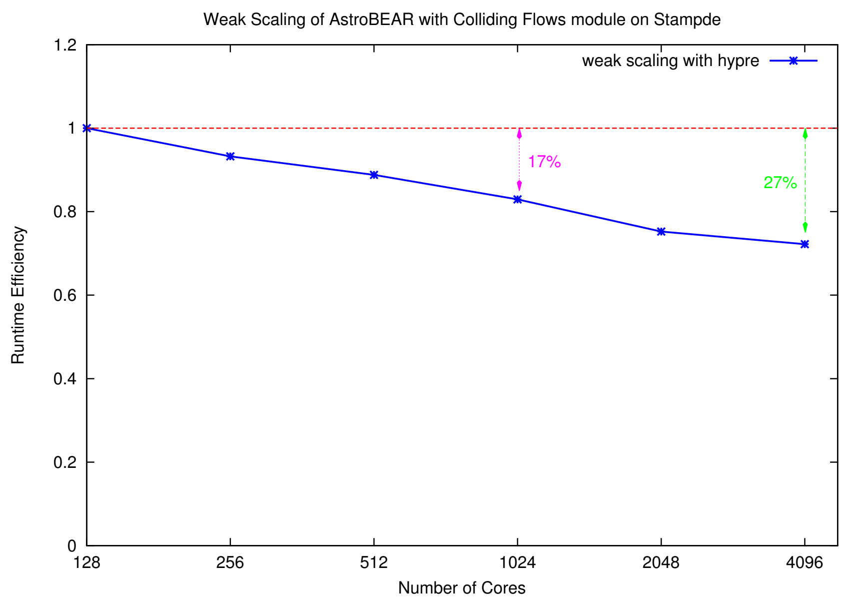

Figure 3: Current weak scaling behavior for AstroBEAR on Stampede. The test is done with fixed grid and same work load on each processor(each processor has 64^3 cells). We run these tests with self-gravity, radiation transfer and using the HYPRE library. The run-time efficiency of AstroBEAR drops only 17% on up to 1096 processors and 27% up to 4096 processors which shows a very good weak scaling considering the multiphysics features involved in these tests. Our typical production runs will be on 1024 cores of Stampede.

3 Summary

Strong scaling behavior of the current AstroBEAR2.0 code were tested on Comet of SDSC and Stampede of TACC. We got very good scaling result on both machines. We also did weak scaling test on Stampede. The result proved the data communication part in the AstroBear code scales very well.