AstroBEAR2.0 and its Performance on BlueHive and BlueStreak

Baowei Liu, Jonathan J. Carroll-Nellenback
Center for Integrated Research Computing

Abstract

AstroBEAR is an Adaptive Mesh Refinement(AMR), multi-physics parallel code for astrophysics. AMR remains at the cutting edge of computational astrophysics. AMR simulations adaptively change resolution within a computational domain to ensure that the most important features of the dynamics are simulated with highest accuracy. By allowing quiescent regions to evolve with low resolution, AMR simulations achieve order of magnitude increases in computational speed. Current AMR simulations require algorithms that are highly parallelized and manage memory efficiently. Here we present both the AMR and parallelization algorithm used in the AstroBEAR 2.0 code. We also present the scaling test and optimization results of AstroBEAR on our local machines Blue Gene/Q and the newly-updated BlueHive Cluster at CIRC.

AMR and Hierarchical AMR Tree

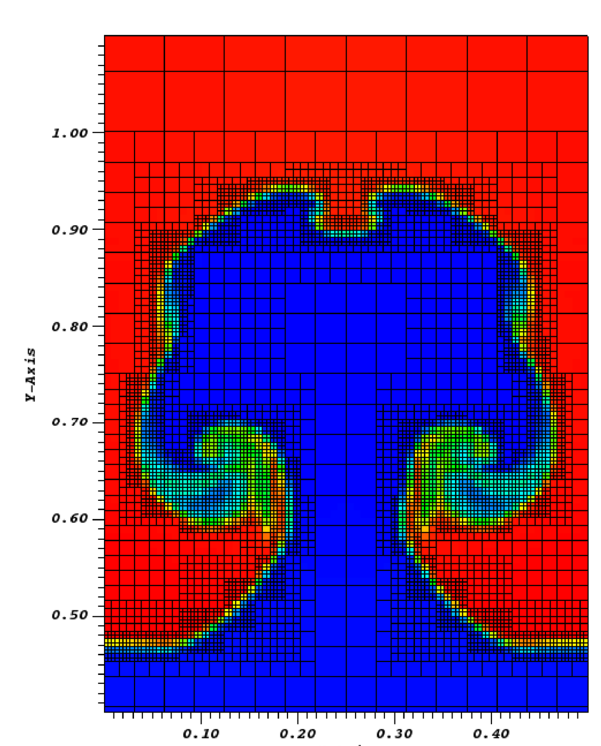

Figure 1: Grids showing 2-level AMR. Generated with 2D RTInstability module of AstroBEAR2.0

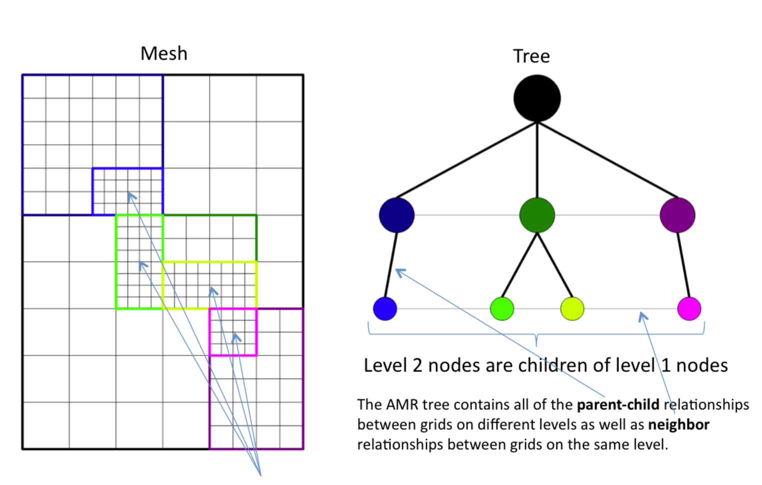

Figure 2: Hierarchical AMR tree in AstroBEAR2.0. The tree changes each time new grids are created by their parent grids. The nodes corresponding to these new grids have to be added to the tree and connected to the overlaps, neighbors and parents.

AstroBEAR
An Adaptive Mesh Refinement Code for Computational Astrophysics

http://astrobear.pas.rochester.edu

Distributed Tree and Level Threading

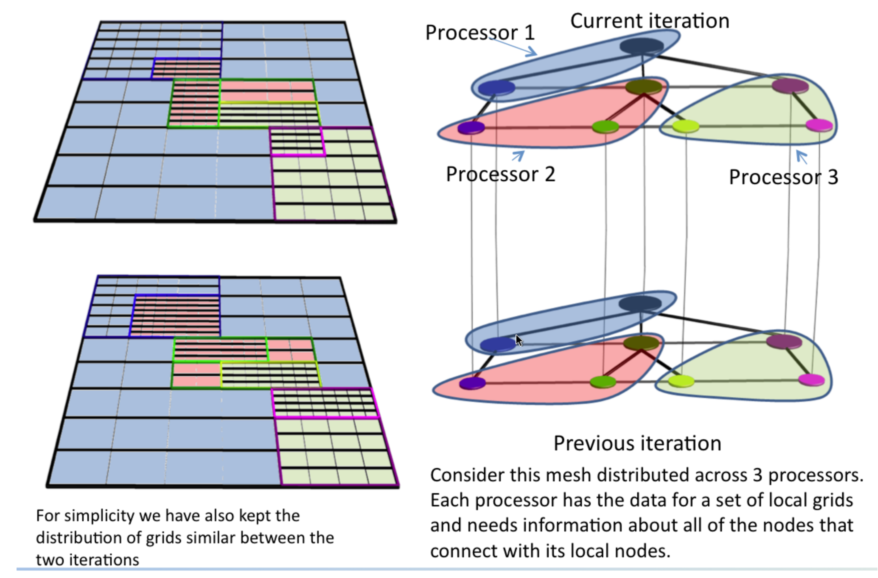

Figure 3: Distributed tree in AstroBEAR2.0. Nodes contain the meta-data that includes the physical location of the grid and the processor containing the grids data. So large-scale patches of data are distributed to computing nodes using MPI. This technique makes AstroBEAR code highly parallelized and able to manage memory efficiently.

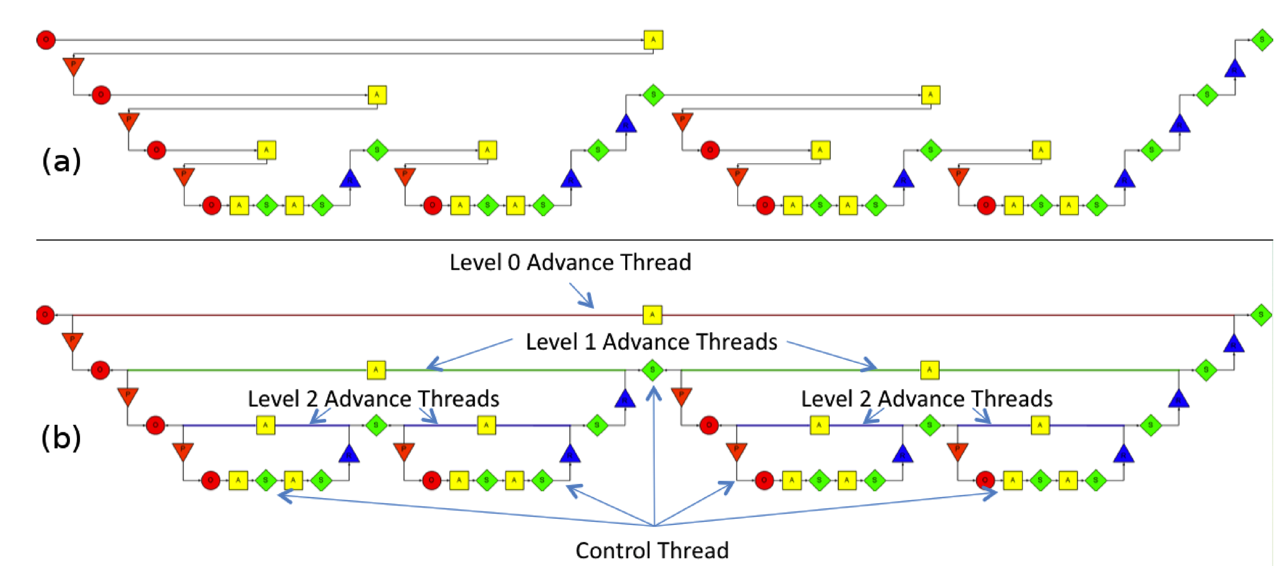

Figure 4: Level threading in AstroBEAR2.0 (a) serial execution; (b) threaded execution. Level threading uses multi-threads to do Advance, Prolongation, Synchronization and Restriction for each level of AMR so it brings extra parallelization to the code.

CIRC Supercomputing Resources

(a) (b)

Figure 5: The Blue Gene/Q system and BlueHive Cluster at the University of Rochester: (a) BlueStreak(# 276 on the November 2014 Top500 list). It consists of 1,024 nodes, 16,384 CPU cores, 16 TB of RAM,and 400 TB of storage. (b) BlueHive. It consists of 284 nodes, 5,656 CPU cores, 21 TB of RAM. It includes the nodes with the Intel Ivy Bridge processors, NVIDIA's K20X GPUs and the Intel Phi 5110P accelerators.

Scaling Test Results on BlueStreak and BlueHive

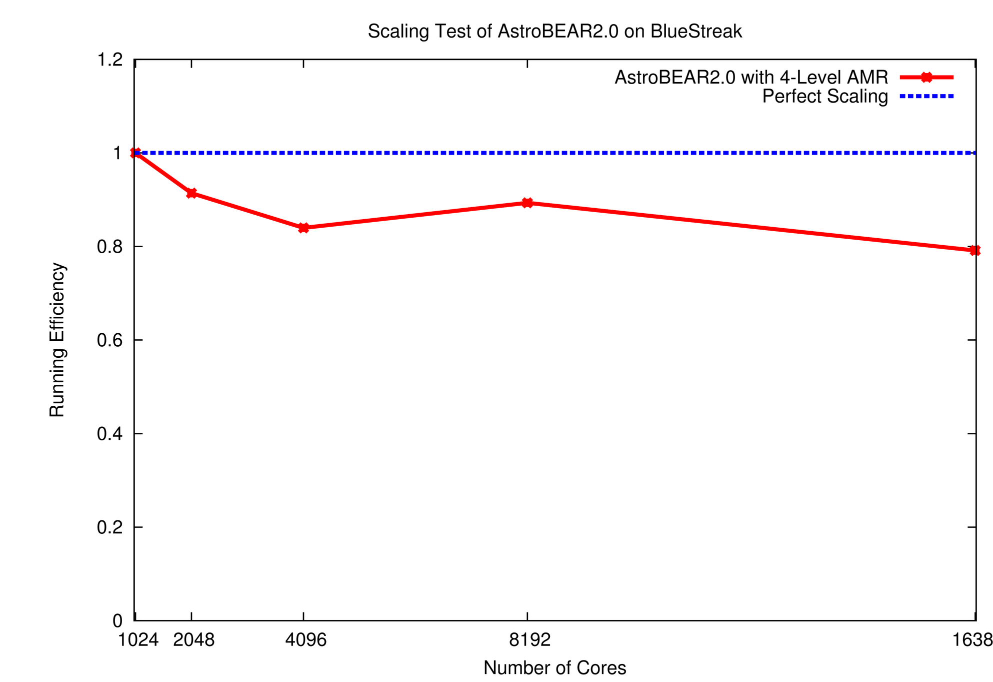

Figure 6: Scaling Test of AstroBEAR2.0 on BlueStreak. The running efficiency of AstroBEAR only goes down about 20% when running on the whole BlueStreak machine (16384 processors).

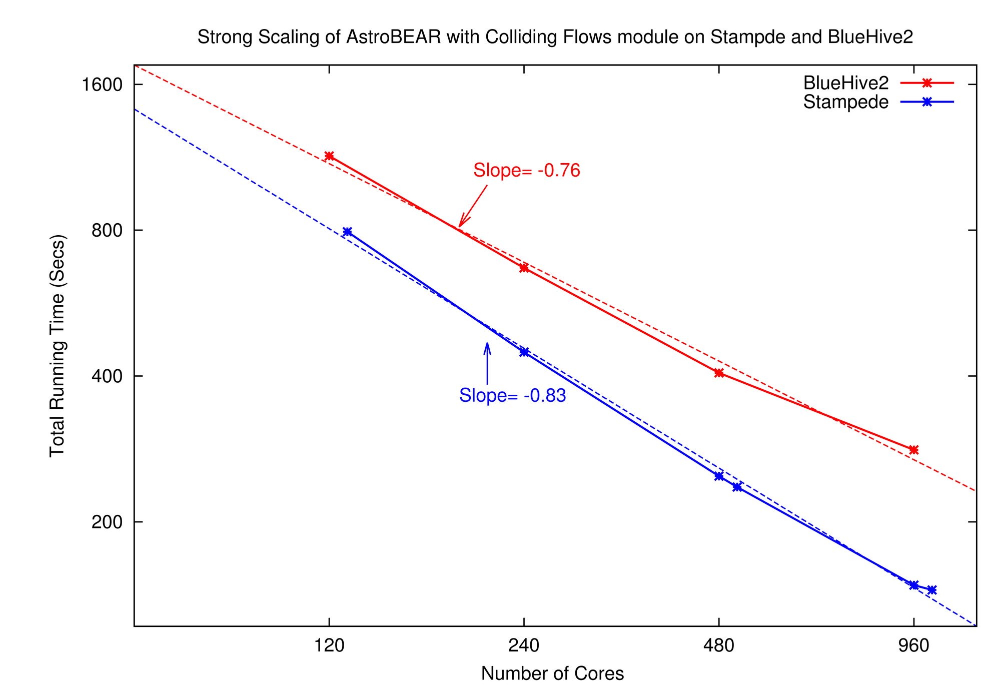

Figure 7: Strong Scaling Test of AstroBEAR2.0 on BlueHive comparing with the result of the same setup on up-to 1000 processors of Stampede (TACC, #7 on the November 2014 Top500 list). It shows good scaling results on BlueHive (with slope$=-0.76$ comparing slope$=-0.83$ on Stampede while the perfect scaling has a slope -1).

Conclusion

AstroBEAR is a multi-dimensional AMR MHD code developed by the UR Astrophysics group. Here we introduce the two key techniques – the adaptive mesh refinement(AMR) and parallelization algorithm with distributed tree and level threading in AstroBEAR2.0. We also show the excellent scaling test results of AstroBEAR2.0 on our Blue Gene/Q system BlueStreak and BlueHive cluster at the University of Rochester. These results tell us AstroBEAR2.0 are highly parallelized and manage memory efficiently.