

Code Performance of AstroBEAR2.0 on Stampede

Adam Frank, Jonathan J. Carroll-Nellenback, Baowei Liu, Martín Huarte-Espinosa,
Shule Li, Erica L. Kaminski, Edward Hansen

April 10, 2014

— Will update by Baowei

Here we present the strong (1) and weak scaling (4) results for AstroBEAR with 3D Colliding Flows on Stampede. The strong scaling tests are done with the exact same resolution as our proposed production runs ($40^3 + 5$ levels AMR but for much shorter final time (0.5% of one frame while the production runs need 200 frames) and without IOs.

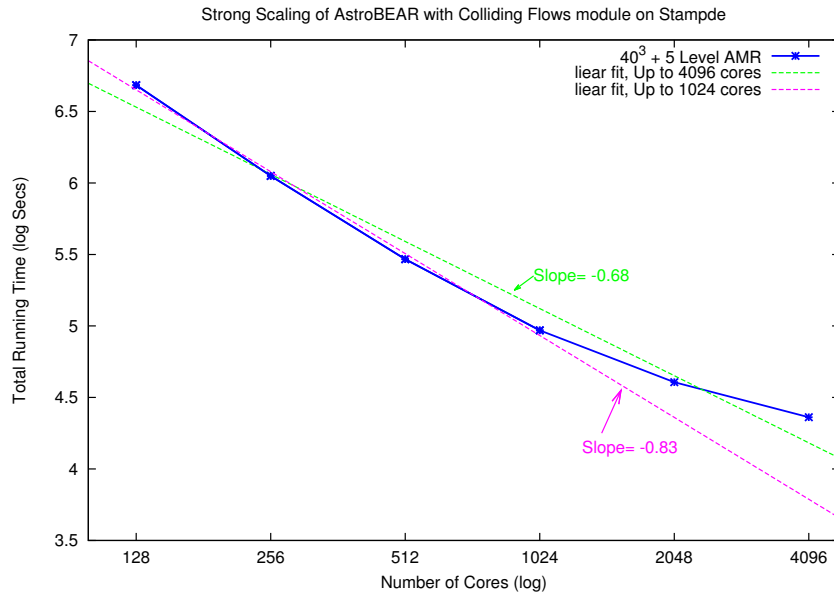


Figure 1: Current strong scaling behavior for AstroBEAR with 3D Colliding Flows module on Stampede. For these tests we run with the exact same resolution as our proposed production runs ($40^3 + 5$ levels AMR for Colliding Flows and $21 \times 105 \times 21 + 5$ levels AMR for Jet), but for much shorter final time (1% of one frame while the production runs need 1000 frames) and without IOs. The current code with the Colliding Flows modules shows excellent scaling on Stampede (with slope= -0.83 while the perfect scaling has a slope -1) up to 1024 processors. The behavior drops with this particular module and production set up when running on more processors (with slope= -0.68) although the code with other modules behaves very well on other systems, i.e., Kraken. Our typical production runs are on 1024 processors and the SUs we request are based on the estimate time on 1024 processors.

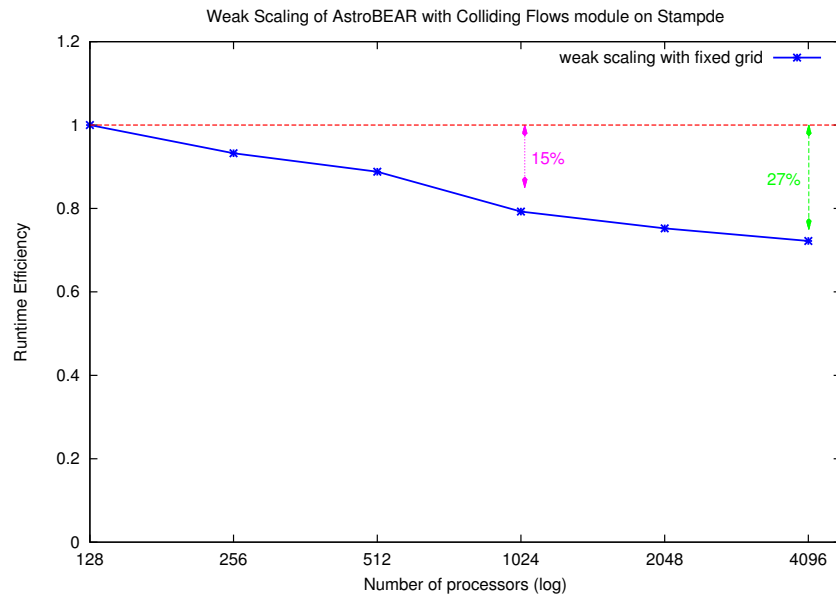


Figure 2: Current weak scaling behavior for AstroBEAR with 3D Colliding Flows module. The test is done with fixed grid and same work load on each processor (each processor has 64^3 cells). Weak scaling is good to test the data communication part of the code since the work load on each processor are about same. The runtime efficiency of AstroBEAR drops only 15% on up to 1096 processors and 27% up to 4096 processors which shows a very good weak scaling. Our typical production runs will be on 1024 cores.

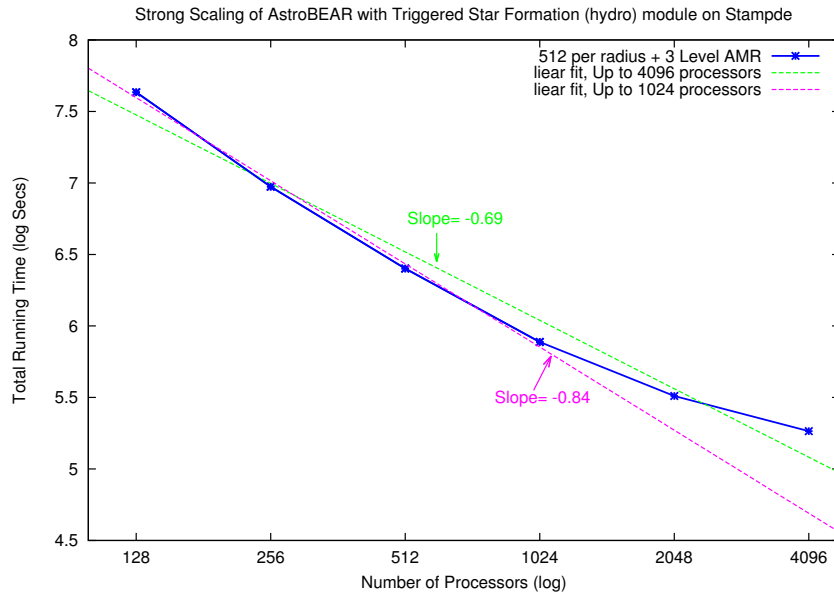


Figure 3: Current strong scaling behavior for AstroBEAR with 3D Triggered Star Formation module on Stampede. The base dimensions for these tests are same as our production runs but with less levels of AMRs (3 levels AMR while the planned production runs has 4 levels). We restart from a point where the star formations starts to get a better estimate for much shorter final time (1% of one frame while the production runs need 100 frames) and without IOs. The current code with the Triggered Star Formation module shows excellent scaling on Stampede (with slope= -0.84 while the perfect scaling has a slope -1) up to 1024 processors. The behavior drops when running on more processors (with slope= -0.69 on up to 4096 processors). Our typical production runs are on 1024 processors and the SUs we request are based on the estimate time on 1024 processors.

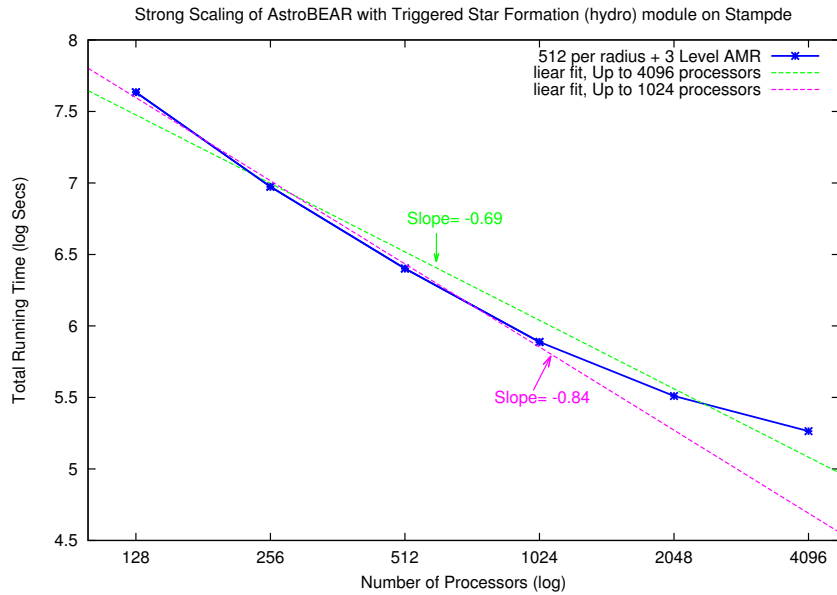


Figure 4: Current strong scaling behavior for AstroBEAR with 3D Triggered Star Formation module on Stampede. The base dimensions for these tests are same as our production runs but with less levels of AMRs (2 levels AMR while the planned production runs has 4 levels). We restart from a point where the star formations starts to get a better estimate for much shorter final time (1% of one frame while the production runs need 1000 frames) and without IOs. The current code with the Triggered Star Formation module shows excellent scaling on Stampede (with slope=-0.84 while the perfect scaling has a slope -1) up to 1024 processors. The behavior drops when running on more processors (with slope=-0.69 on up to 4096 processors). Our typical production runs are on 1024 processors and the SUs we request are based on the estimate time on 1024 processors.