GPU-ACCELERATED ADAPTIVE MESH REFINEMENT

Allocation: Director Discretionary/500 Knh
PI: Hsi-Yu Schive[1]
Collaborators: Matthew Turk[2], John ZuHone[3], Nathan Goldbaum[1]

[1]National Center for Supercomputing Applications
[2]University of Illinois at Urbana-Champaign
[3]Harvard-Smithsonian Center for Astrophysics

EXECUTIVE SUMMARY

GAMER is a highly scalable and multi-GPU-accelerated adaptive mesh refinement (AMR) code for astrophysics. It adopts a hybrid OpenMP/MPI (message passing interface)/GPU (graphics processing unit) parallelism model to utilize both CPU (central processing unit) and GPU computing power and to minimize MPI communication. Further, it overlaps CPU computation, GPU computation, and CPU–GPU communication to maximize computational efficiency. In this project, we conduct a series of performance benchmarks on Blue Waters and demonstrate high parallel efficiency for both weak and strong scaling using up to 4,096 XK nodes. The code thus provides a unique numerical tool to study various astrophysical phenomena requiring resolutions that are not realistically attainable by other CPU-based AMR codes. For example, based on the benchmark simulations of merging galaxy clusters, the performance of GAMER using 256 XK nodes is found to be 42 times faster than FLASH, a widely adopted CPU-based AMR code, using 256 XE nodes.

RESEARCH CHALLENGE

The AMR technique has played an indispensable role in computational astrophysics due to the large dynamical range demanded. However, compared to the uniform-resolution approaches, it remains extremely challenging for AMR codes to fully exploit the petascale computing power in heterogeneous CPU/GPU supercomputers like Blue Waters. This is mainly due to the complicated AMR data structure, load imbalance, expensive MPI communication, and the great amount of work required to convert existing time-consuming physical modules to run with high efficiency on GPUs.

Most previous GPU–AMR codes are based on considerably simplified test problems or run on a much smaller number of nodes. In comparison, here we measure performance directly from simulations of binary cluster mergers with large dynamic range capable of capturing the large-scale effects of the cluster merger as well as resolving the properties of turbulence down to the kiloparsec scale, almost an order of magnitude finer than previous work [1]. Moreover, we compare the overall performance directly with FLASH [2], a widely adopted CPU-based AMR code, and further demonstrate the parallel scalability of GAMER on hundreds to thousands of XK nodes, for which achieving good load balance becomes highly nontrivial.

METHODS & CODES

The GAMER code [3] has the following important features.

- **Hybrid OpenMP/MPI/GPUs.** GAMER uses GPUs as PDE solvers and CPUs to manipulate the AMR structure. It uses OpenMP for intra-node parallelization in CPUs, MPI for inter-node communication, and CUDA as the GPU programming interface.
- **Overlapping computation.** CPU computation, GPU computation, and CPU–GPU communication are allowed to overlap, greatly improving the overall throughput when CPUs and GPUs take a similar time to complete their own tasks.
- **Hilbert space-filling curve** for load balance.
- **Efficient usage of memory.** GAMER stores all data in the CPU memory and only temporarily transfers data to GPU, which allows exploitation of the larger CPU memory. Communication overhead between CPU and GPU is usually negligible since it can be overlapped by both CPU and GPU computations efficiently using CUDA streams.
- **Bitwise reproducibility.** The order of all floating-point operations in GAMER is carefully designed to be deterministic. It thus supports bitwise reproducibility when (1) running simulations with different numbers of MPI ranks and OpenMP threads, and (2) restarting simulations from checkpoint files. This feature is essential for scientific reproducibility.
- **Inline analysis with *yt*.** GAMER supports the ability to pass in-memory data structures to *yt* [4], a powerful Python-based package for analyzing and visualizing volumetric data. Moreover, using *yt* allows one to share the data analysis scripts to the community straightforwardly, which greatly improves the scientific reproducibility.

RESULTS & IMPACT

We have compared both the performance and accuracy between GAMER and FLASH based on the merging galaxy clusters simulation setup of [1]. These simulations include hydrodynamics, self-gravity, dark matter particles, and AMR, and thus provide a comprehensive test for the simulation codes. Fig. 1 shows the slices of gas temperature evolution visualized with *yt*, demonstrating that the results of the two codes are remarkably consistent.

Fig. 2 shows the strong scaling. Blue and green lines indicate the comparison with a relatively lower resolution of 7 kiloparsecs. GAMER (on XK nodes) is found to be ~27 to 42 times faster than FLASH (on XE nodes) when using the same number of nodes. Most importantly, this speedup remains reasonably flat when increasing the number of nodes. The red line shows the performance of GAMER with a resolution eight times higher (~0.9 kiloparsecs), which achieves much higher parallel efficiencies of ~50% on 512 nodes and ~30% on 2,048 nodes.

The extremely high resolution and performance demonstrated in this project will allow astrophysicists, for the first time, to resolve the turbulence cascade in the intracluster medium down to scales lower than the particle mean free path, reducing the numerical viscosity down to a scale below where the effects of a physical viscosity are expected to become relevant. It will greatly help calibrate the mass estimation of galaxy clusters, refine the active galactic nucleus feedback model, and improve our understanding of the source of cosmic rays.

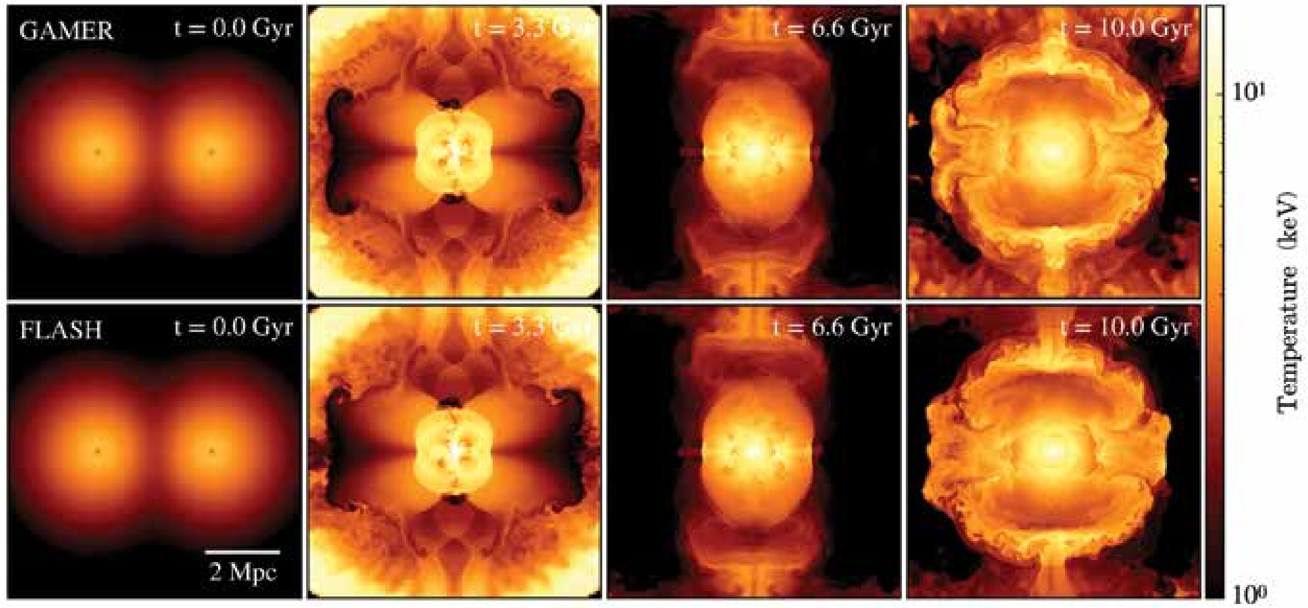

Figure 1: Comparison of the gas temperature evolution in the binary cluster merger simulations between GAMER (upper panels) and FLASH (lower panels). The results are remarkably consistent.

WHY BLUE WATERS

The benchmark conducted in this project would be impossible without Blue Waters. These simulations require a great amount of CPU and GPU memory, and also require the balance among CPU performance, GPU performance, and MPI communication to achieve an optimum throughput. Most importantly, running thousands of GPUs in parallel is extremely demanding on system stability. Blue Waters is the supercomputer that best fits these requirements.

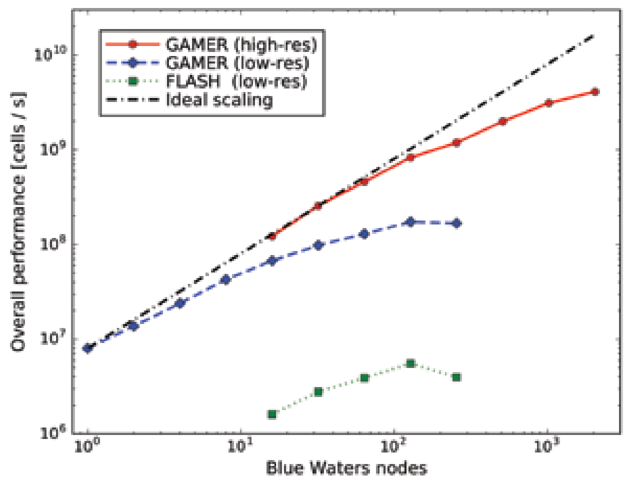

Figure 2: Strong scaling of GAMER in the binary cluster merger simulations.