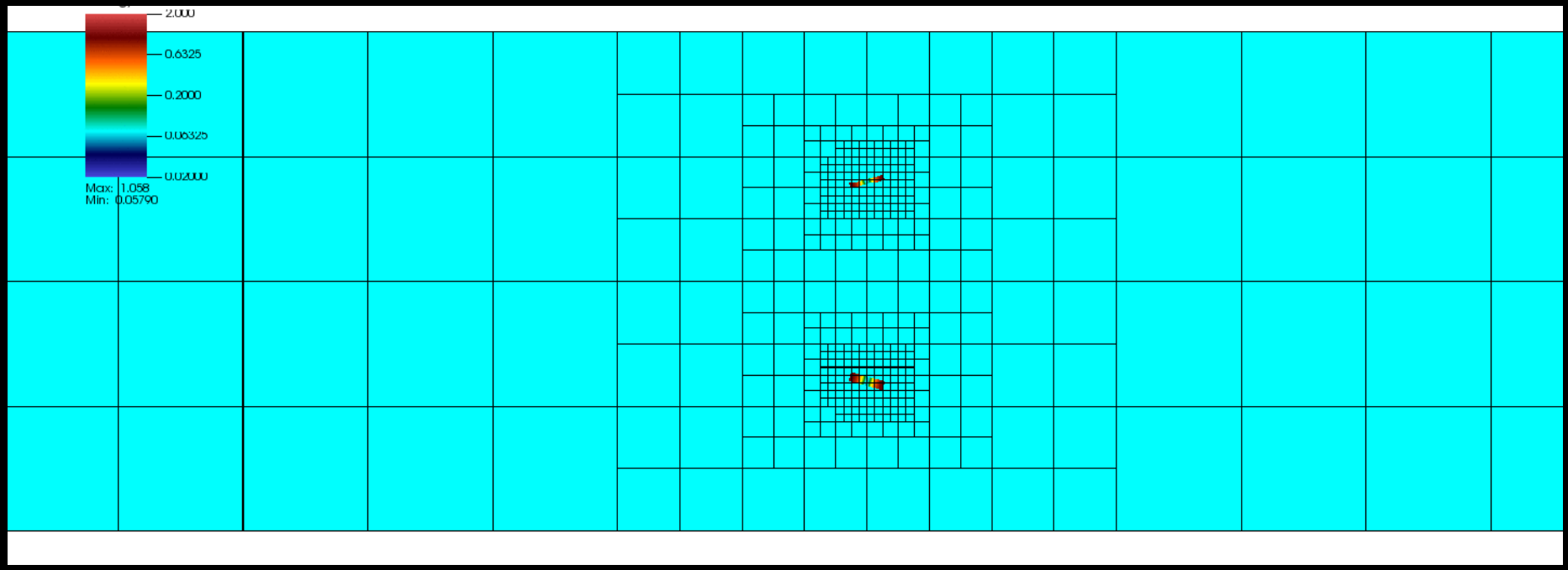

GAMER
GPU-accelerated Adaptive-MEsh-Refinement
Justin Schive (薛熙于)
Edward Seidel, Gabrielle Allen,
John ZuHone, Matthew Turk,
Nathan Goldbaum

Key Features

- Framework of GPUs + Adaptive-Mesh-Refinement
- Hybrid MPI/OpenMP/GPU parallelization
  - Overlapping between CPU-GPU data transfer, CPU computations, and GPU computations
- Weak and strong scaling using up to 4,096 XK nodes on Blue Waters
  - 5 ~ 40 times faster than CPU-based codes (using the same number of XE/XK nodes)
- YT for data analysis and visualization

Adaptive Mesh Refinement (AMR)

- Simulation resolution adapts automatically



CPU-GPU Collaboration

- **Two main tasks in AMR:**

1. **Patch construction:** octree construction, decision making, data interpolation, data copy…

   ~ complicated, but consume less time

   ➡ CPUs

2. **3-D PDE solvers (e.g., hydrodynamics, Poisson):**

   ~ straightforward, but time-consuming

   ➡ GPUs

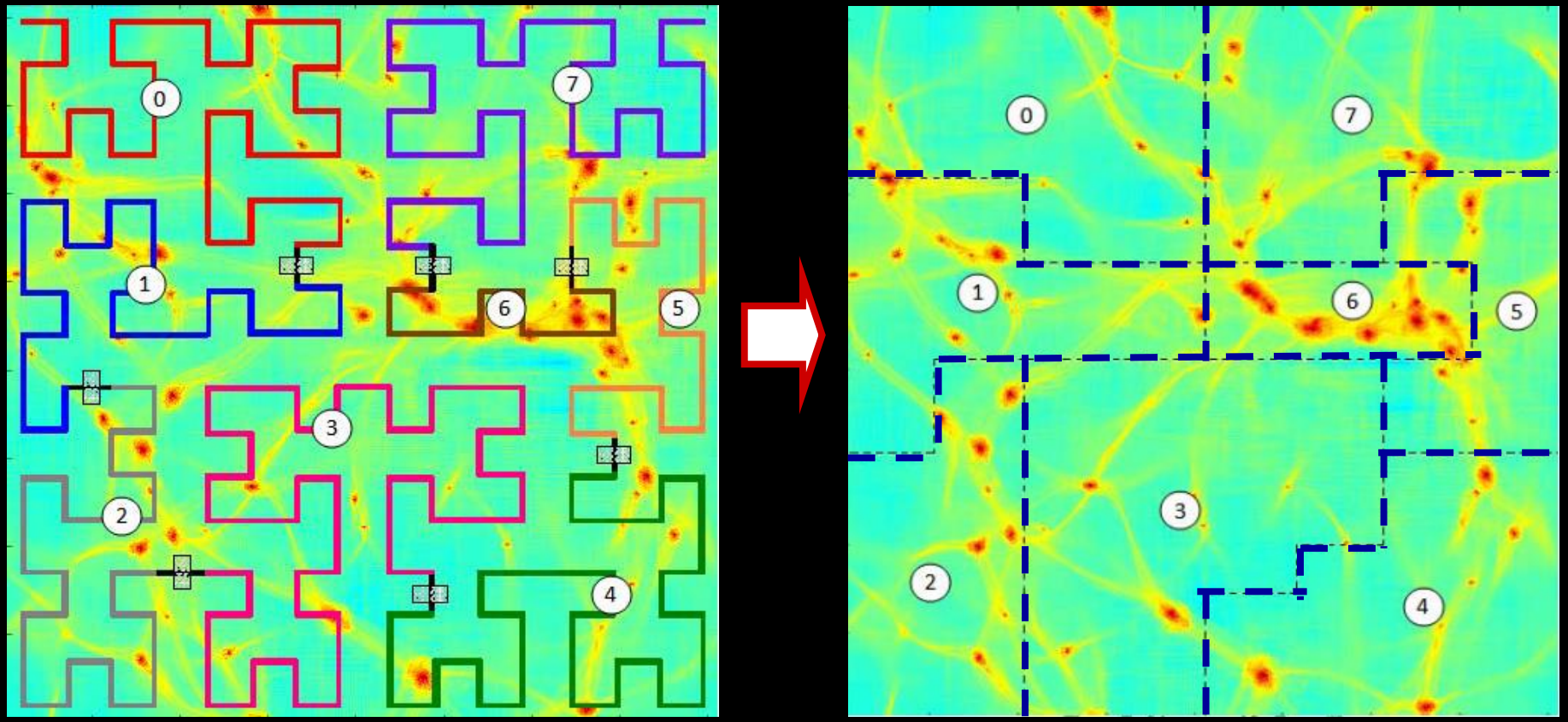
   ➡ feed with hundreds of patches simultaneously

Performance Optimization

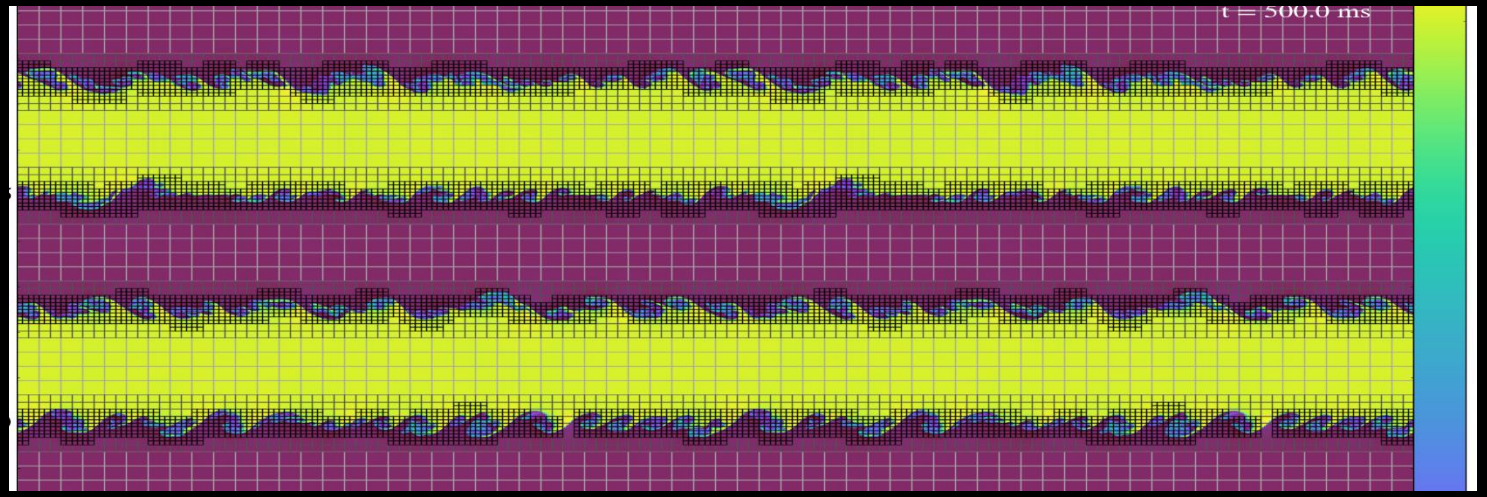
- Overlap CPU↔GPU communication with GPU computation
- Overlap CPU and GPU computations
- OpenMP → Fully exploit the computational horsepower in heterogeneous systems like Blue Waters
  - E.g., 4,096 nodes, 4,096 GPUs, 65,536 CPU cores
- MPI → Multi-GPU and multi-CPU acceleration
  - Hybrid MPI / OpenMP / GPU parallelization
  - Hilbert space-filling curve → load balance

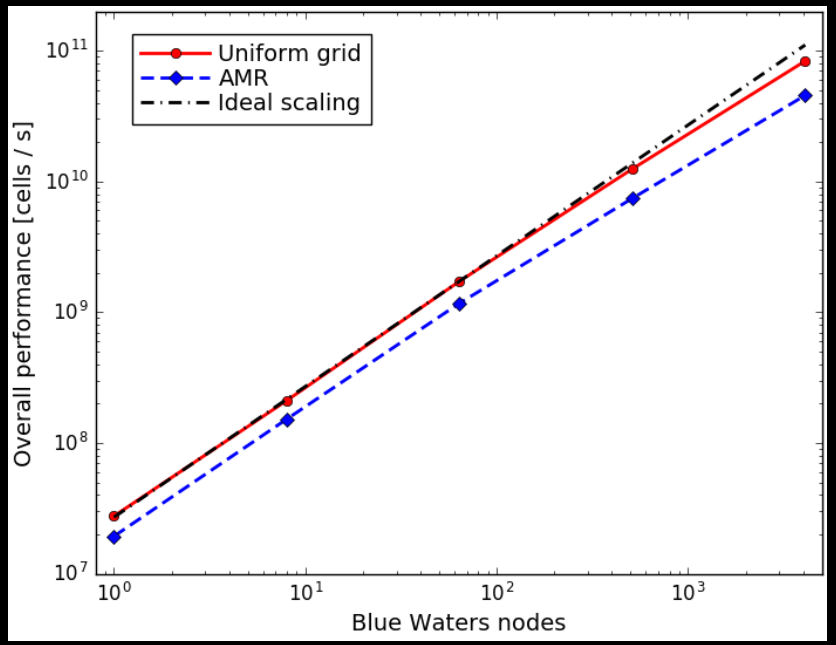
Workload Balance

- **Hilbert space-filling curve** domain decomposition
  → load balance, data locality



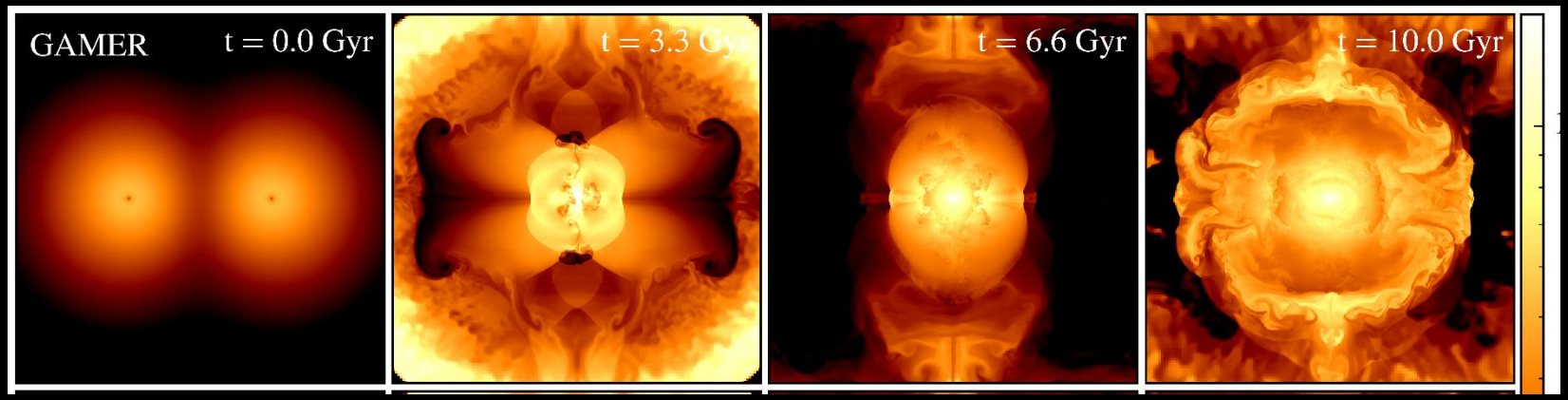
Weak scaling on Blue Waters

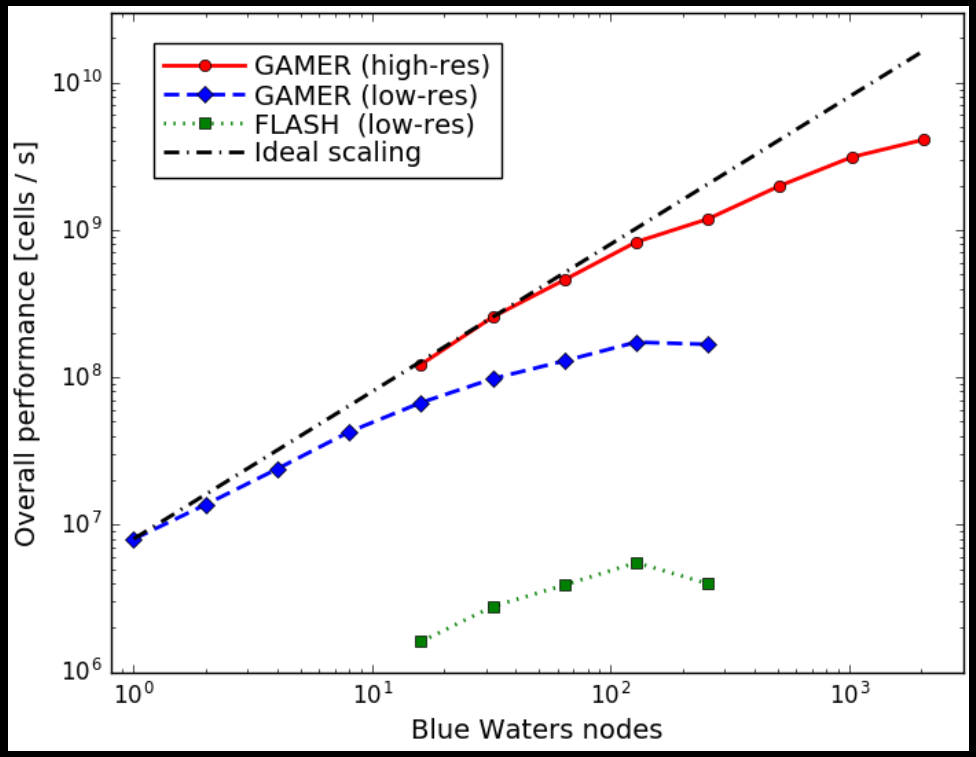




- Up to 4,096 GPUs
- Parallel efficiency ~ 70%
- $10{,}240^3$ resolution
- Sustained performance ~ 8.3×10^{10} cells/sec
- Pure hydro in this test. MHD performance is similar.

Strong Scaling on Blue Waters

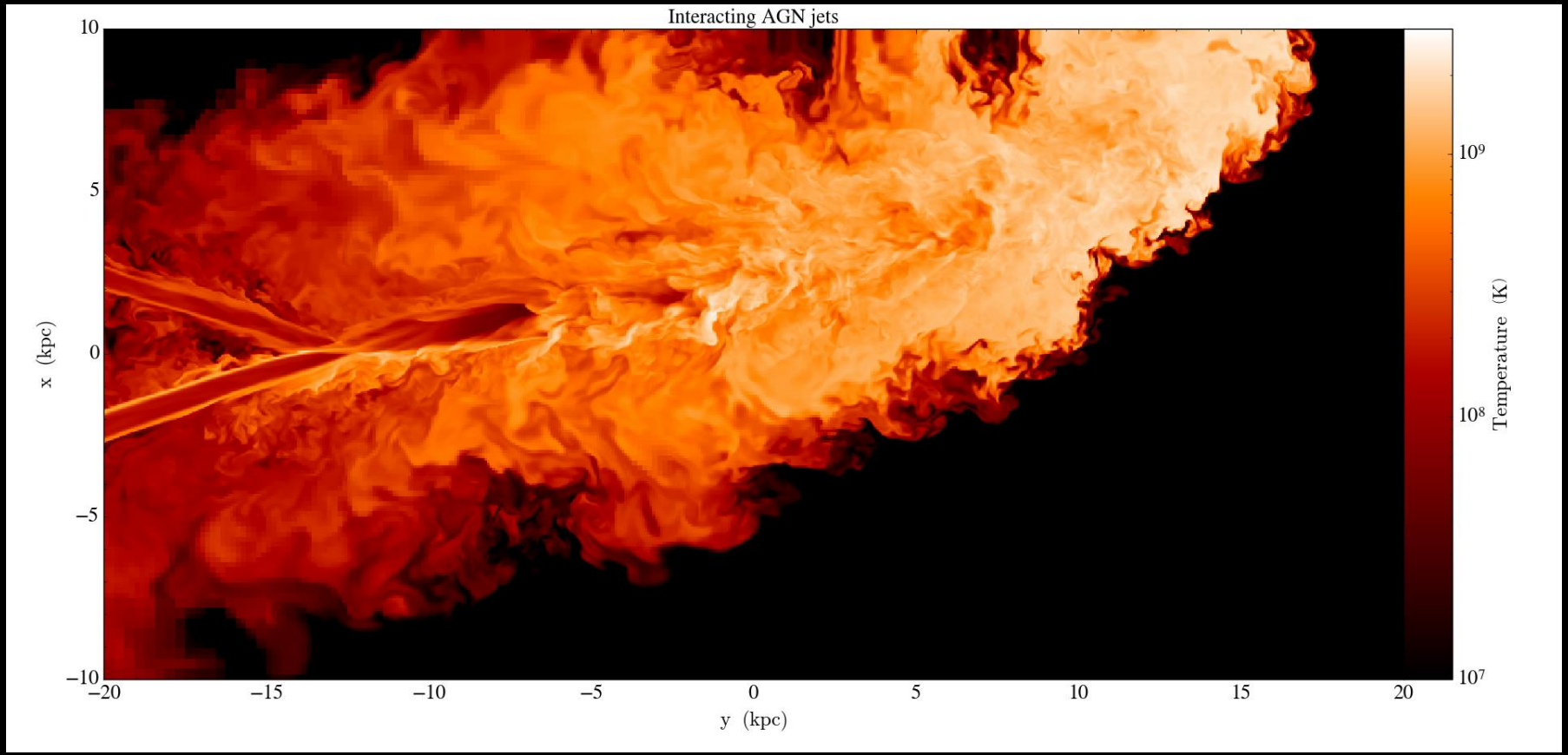




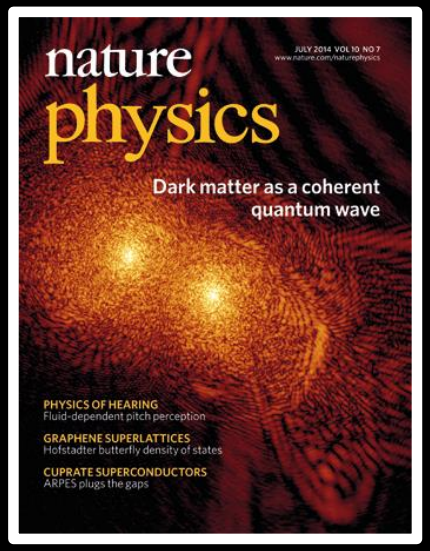
- Up to 2,048 GPUs
- Parallel efficiency: 30% ~ 50%
- GAMER (XK nodes) vs. FLASH (XE nodes): 27 ~ 42 speedup
- Merging cluster simulations with both hydro, self-gravity, particles, and AMR

YT Support

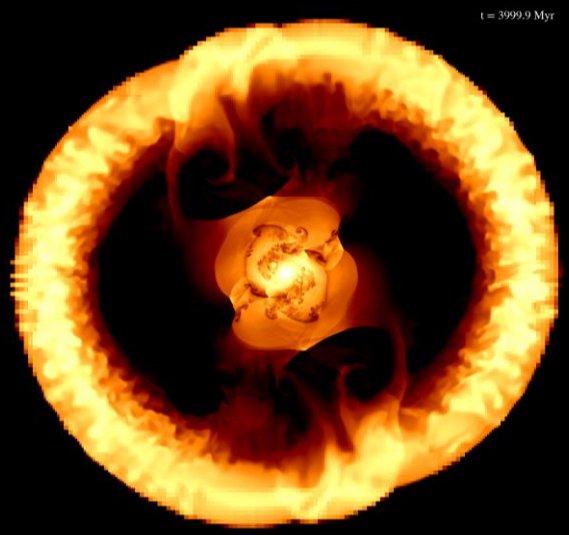
- YT: extremely powerful package for data analysis and visualization
  - Python-based and support various frontends → extremely important for scientific reproducibility
- libYT: use YT for inline analysis and visualization



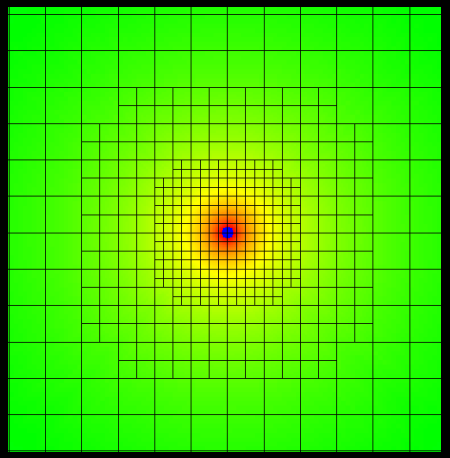
Astrophysical Applications



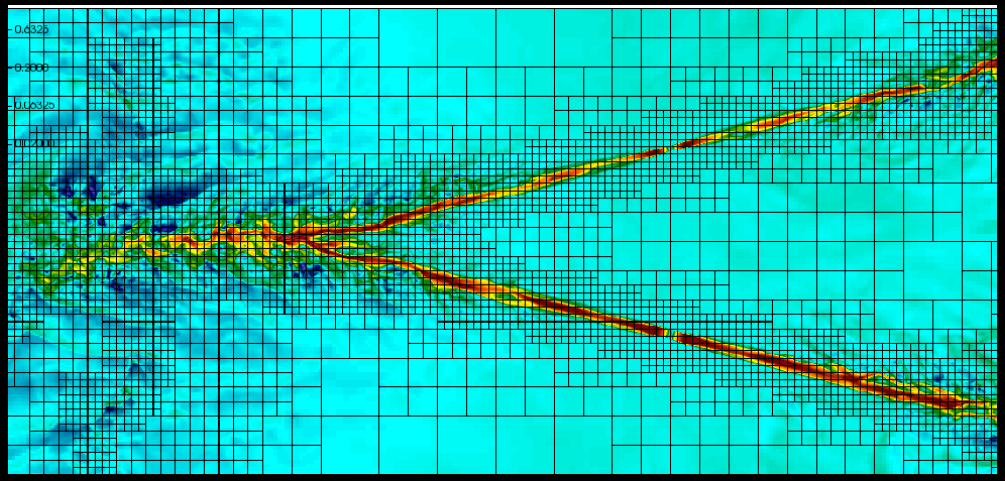
Wave dark matter



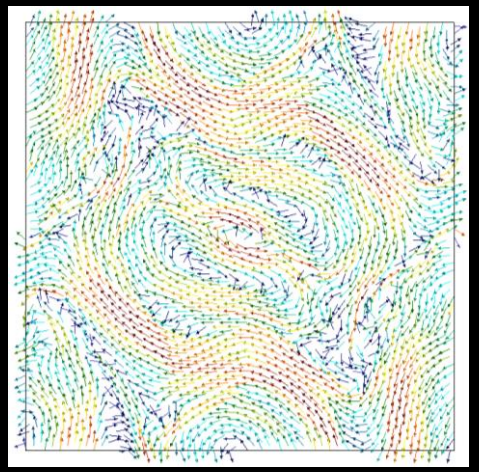
Merging clusters of galaxies



Supper-massive black halo accretion



Supersonic jets in active galactic nucleus



MHD turbulence

Future work

- Publicly available
- Improve load balancing
- More physical modules
  - Radiation cooling (with Grackle), star formation, baryonic feedback, ...
- Improve the inline analysis with libYT
- Science
  - GAMER can reach extremely high resolution that has been impossible until now