GAMER
(GPU-accelerated Adaptive-MEsh-Refinement) & Out-of-core Computation

H. Y. Schive (薛熙于)
Graduate Institute of Physics, National Taiwan University
Leung Center for Cosmology and Particle Astrophysics (LeCosPA)

T. Chiueh (闕志鴻), Y. C. Tsai (蔡御之)
Graduate Institute of Physics, National Taiwan University
Leung Center for Cosmology and Particle Astrophysics (LeCosPA)

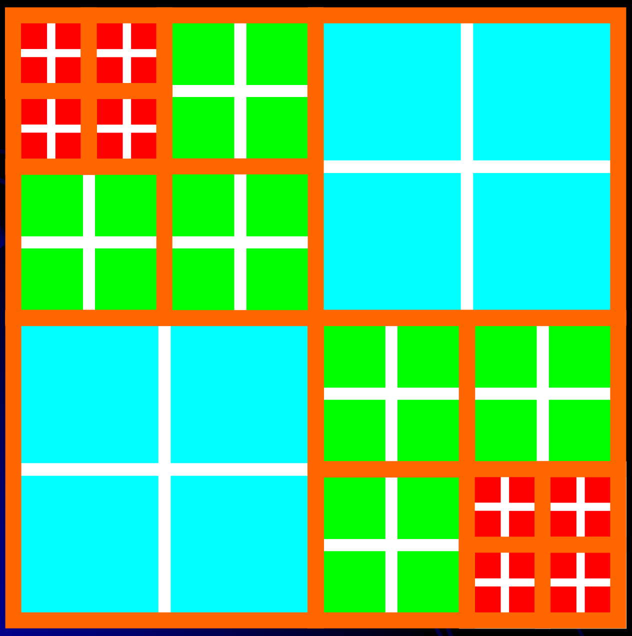
IAUS 270 (06/03/2010)

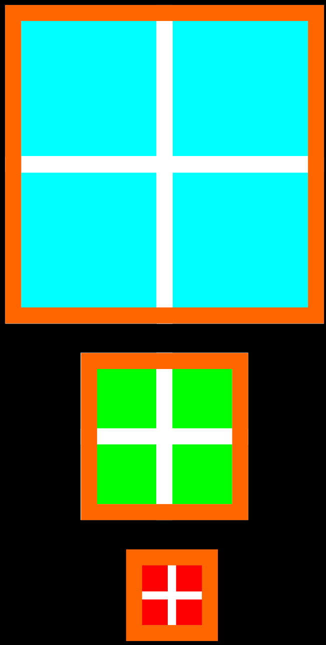
Outline

- AMR
- AMR + GPUs
  - Performance (Hydrodynamics / Poisson / Overall)
  - Optimization
- AMR + GPUs + OOC (out-of-core)
- Conclusion and Future Work

AMR Scheme in GAMER

- Refinement unit : patch (containing a fixed number of cells, e.g., 8^3), similar to FLASH
- Hierarchical oct-tree data-structure
- Individual time-step





CPU-GPU Collaboration

- Two main tasks in AMR:

1. Patch construction : decision making, interpolation, complex data-structure, data assignment …

   ~ complicated, but consume less time

   ⇨ CPUs

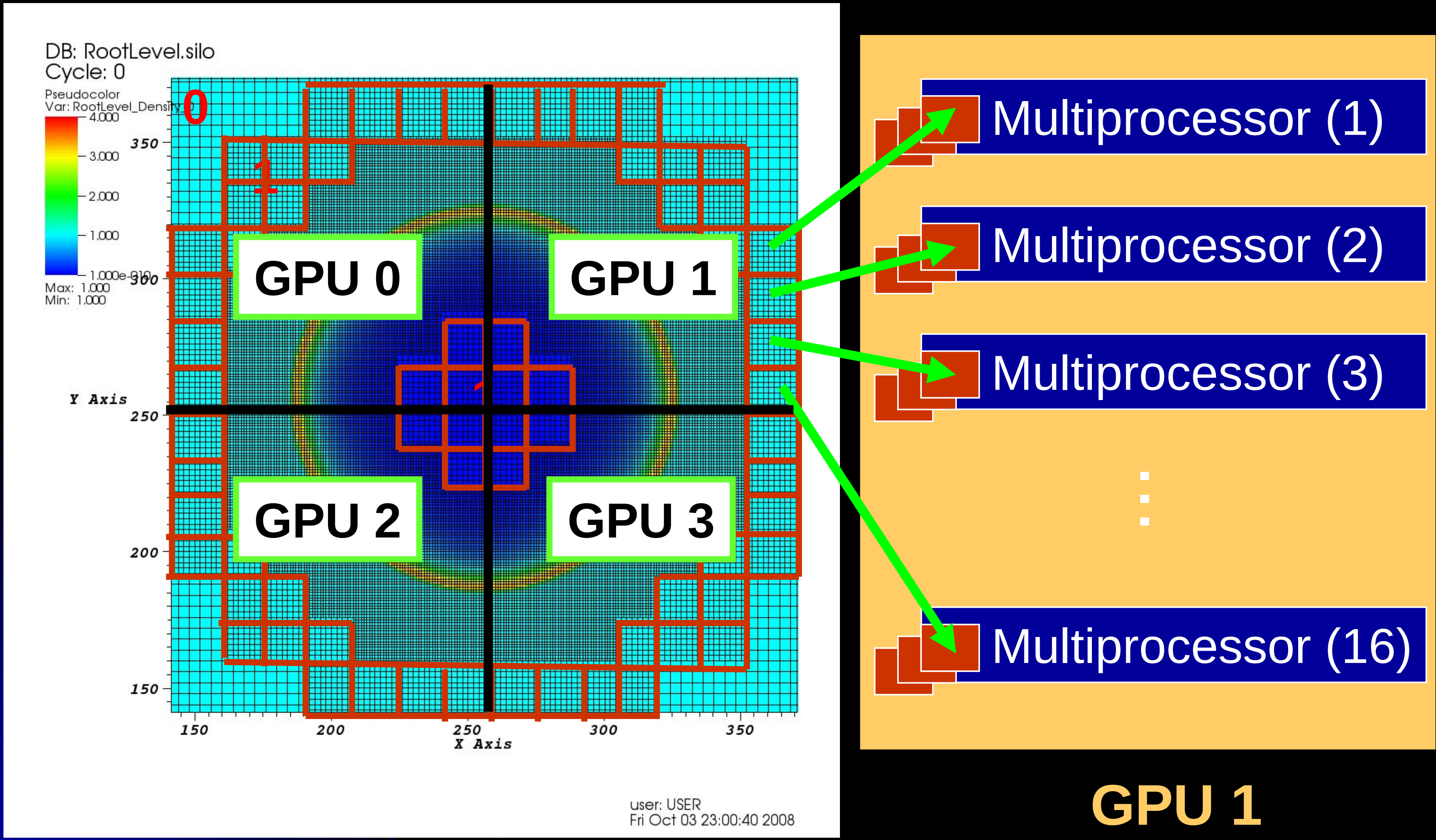
2. 3-D hydrodynamic + Poisson solvers :

   ~ straightforward, but time-consuming

   ⇨ GPUs

   ⇨ feed with hundreds of patches simultaneously

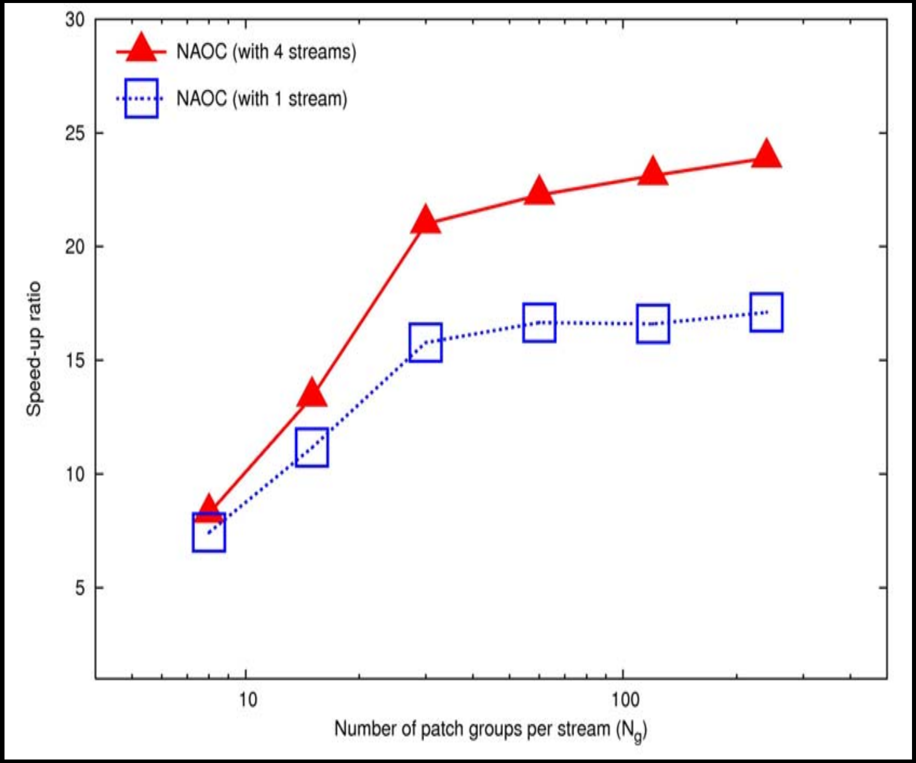
Multi-GPU Example



Performance : Hydrodynamic Solver

- **Second-order relaxing TVD scheme**
- **Data transfer between CPU and GPU is overlapped by GPU computation**
- **Currently the ghost-zone interpolation is performed by CPU**
- **One T10 GPU vs. one Xeon E5520 CPU core**

  → Speed-up ratio : 23.9x



: Asynchronous memory copy

: Synchronous memory copy

Performance : Poisson Solver

- Root level : fast Fourier transform (FFT)

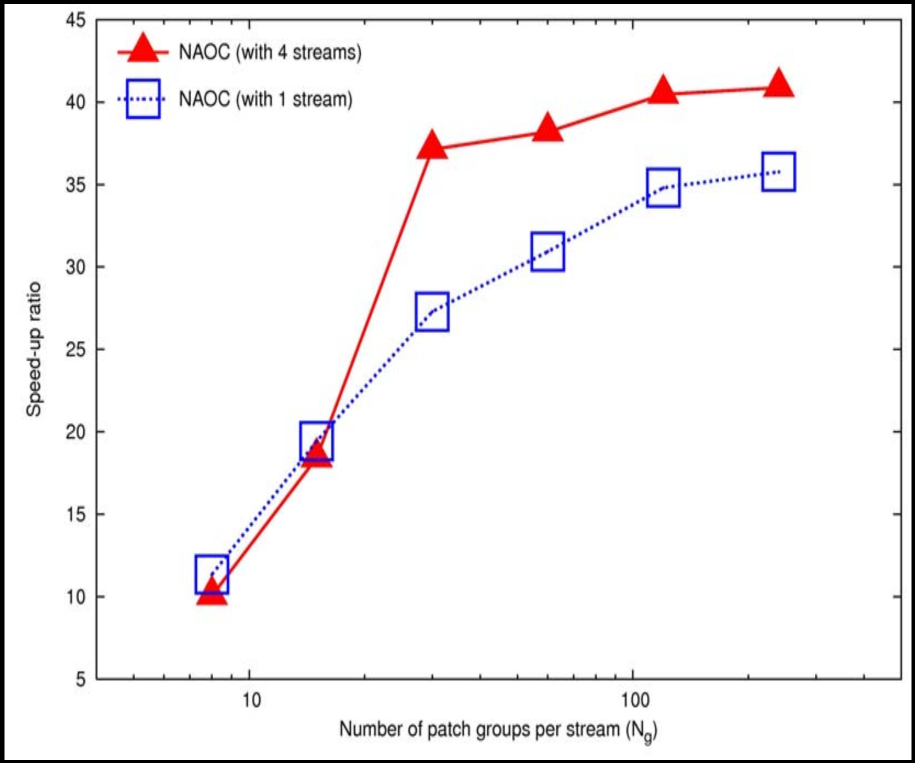
  → use CPUs only

- Refinement levels : successive overrelaxation method (SOR)

  → use GPUs

- Coarse-grid interpolation is performed by GPU
- One T10 GPU vs. one Xeon E5520 CPU core

  → Speed-up ratio : 40.9x

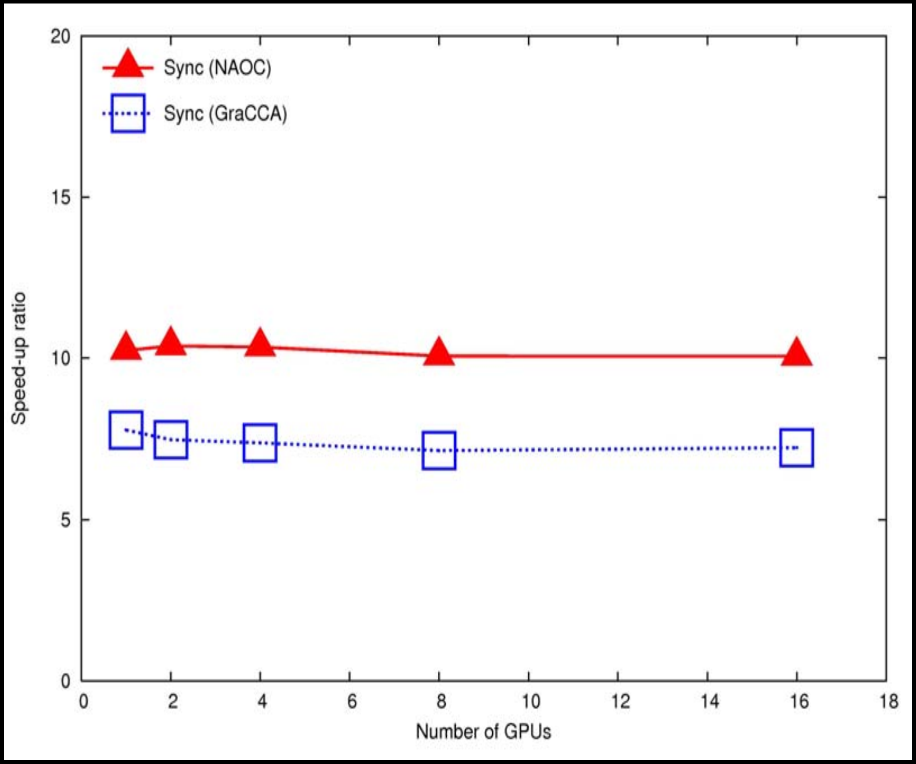


: Asynchronous memory copy

: Synchronous memory copy

Performance : Overall

- GPU vs. CPU
  - # of GPUs : 1 ~ 16
  - One GPU in each computing node
- Purely baryonic cosmological simulation
  - Root level: 256^3
  - 5 refinement levels
    → Effective resolution: 8192^3
- Speed-up ratio
  - 10.23x (1 GPU vs. 1 CPU core)
    ↓
    10.05x (16 GPUs vs. 16 cores)
- z=100 to z=0, 16 GPUs
  → 8 hours (725 root-level steps)

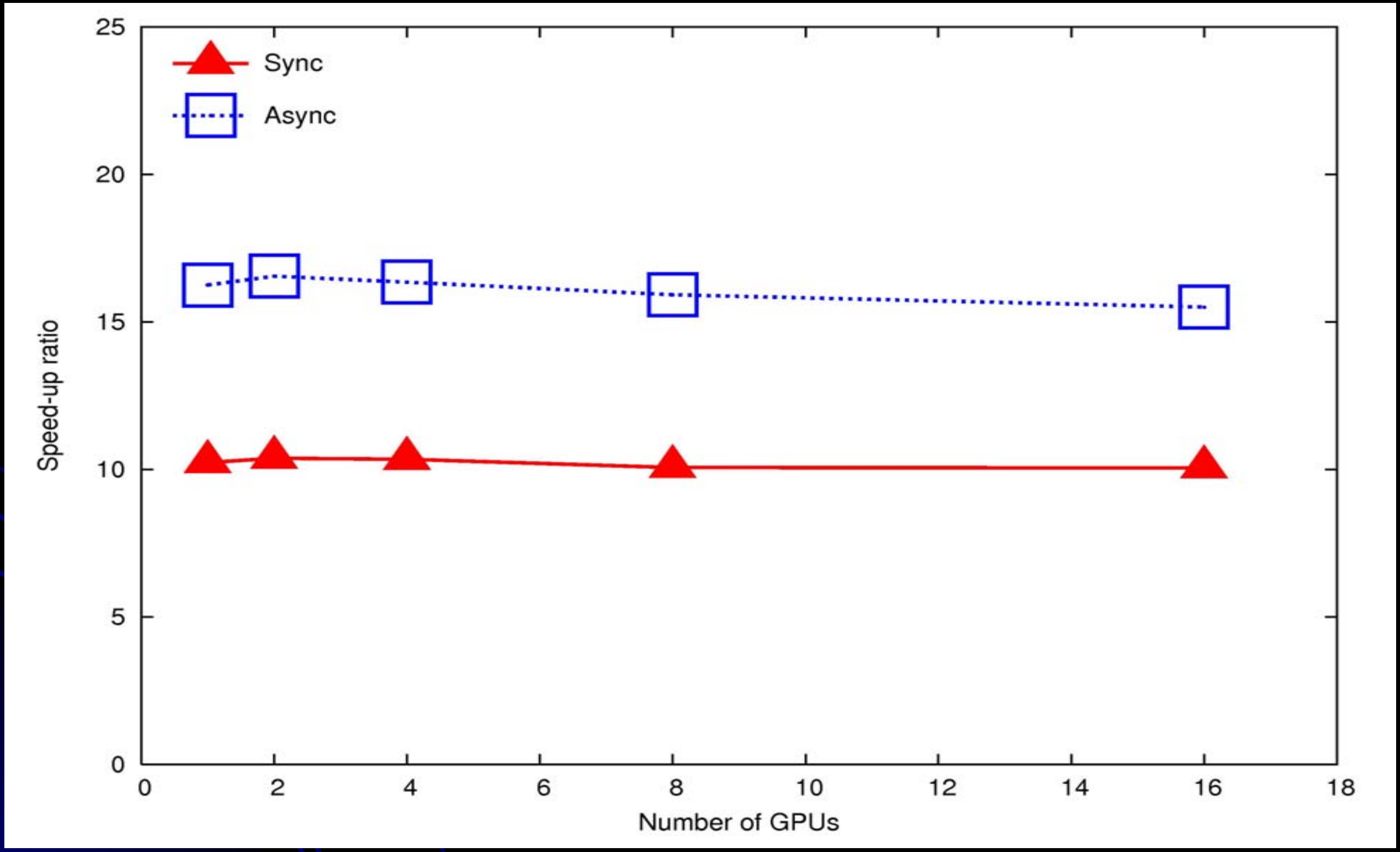


: T10 vs. Xeon E5520
: GeForce 8800 GTX vs. Athlon 3800

Optimization :

Concurrent Execution between CPU and GPU

- **Speed-up ratio : 10.23x → <u>16.25x</u>**



: Asynchronous

: Synchronous

Future Optimizations

- To be honest, # of CPU cores / GPUs per node is usually 2~4
- Issue : Fluid solver: CPU time >> GPU time

  1. Perform the ghost-zone interpolation in GPU

  2. Relaxing TVD scheme is not very computation-intensive

  → Adopt a more accurate scheme, e.g., PPM, approximate/exact Riemann solver …
- SOR method is too slow …
  - Multi-grid, FFT, super-stepping …
- Not load-balance → space-filling curve
- 128 GPUs benchmark tests are on the way !

AMR + GPUs + Out-of-core

Motivation

- **Performance :** GPU / CPU → 10x

1 small simulation ⇒ 10 small simulations

~~1 larger simulation ?~~

Limited memory

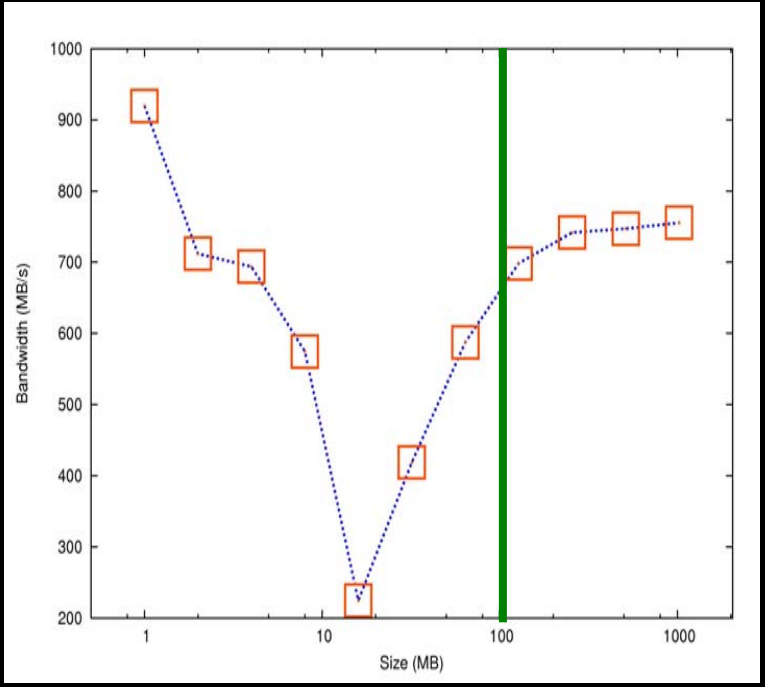
- **Memory :** Hard disk / Ram → 10x ~ 100x

⇒ 1 ~ 8 TB memory per node ?

Issue I : Hard Disk Bandwidth

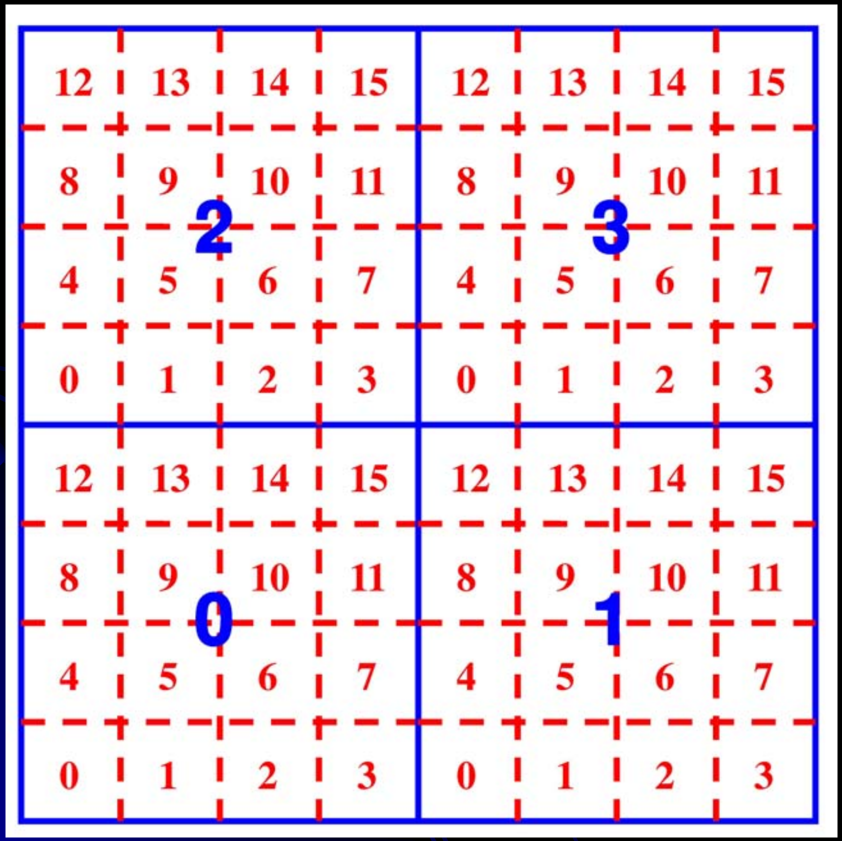
- Single HD : ~ 100 MB/s → Multiple HDs ??
- Prototype : 8 HDs → 750 MB/s
  - Distribute data by direct I/O, not RAID
    → More detailed control of data storage

Spartan



Issue II : Out-of-core + AMR

- Just apply the same domain decomposition as the case using MPI only



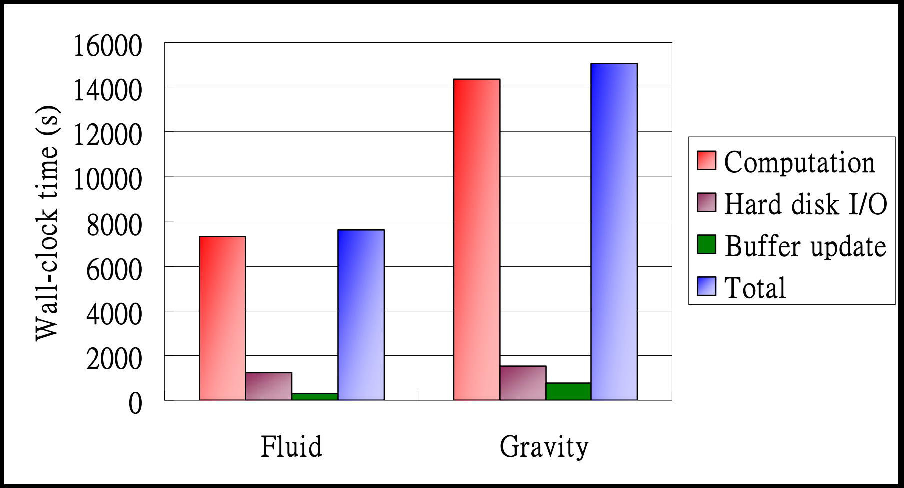
- BLUE number : MPI rank
  - In different nodes
  - Updated in parallel
  - Data transfer : network
  - MPI_Send, MPI_Recv

- RED number : OOC rank
  - In the same node
  - Updated sequentially
  - Data transfer : hard disk
  - OOC_Send, OOC_Recv

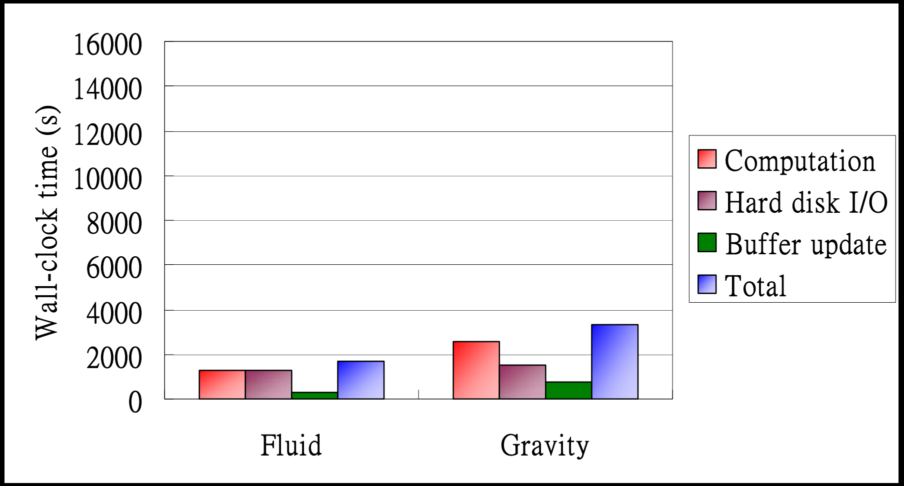
Performance I : Uniform Mesh

- Resolution: 2048^3 grids
- Total memory requirement: ~ 400 GB
  - 50x larger than the ram in our prototype system
- Decomposed into 8^3 OOC ranks in a single node
- Each OOC rank works on 256^3 grids

CPU



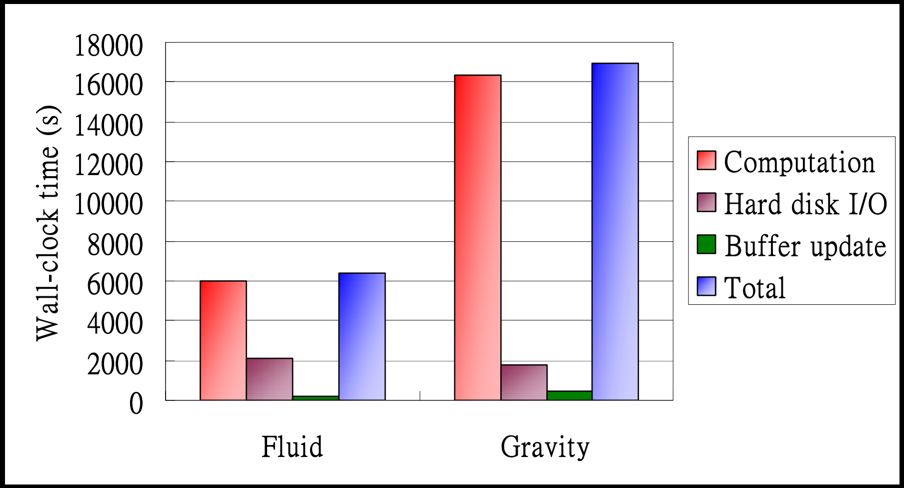
GPU



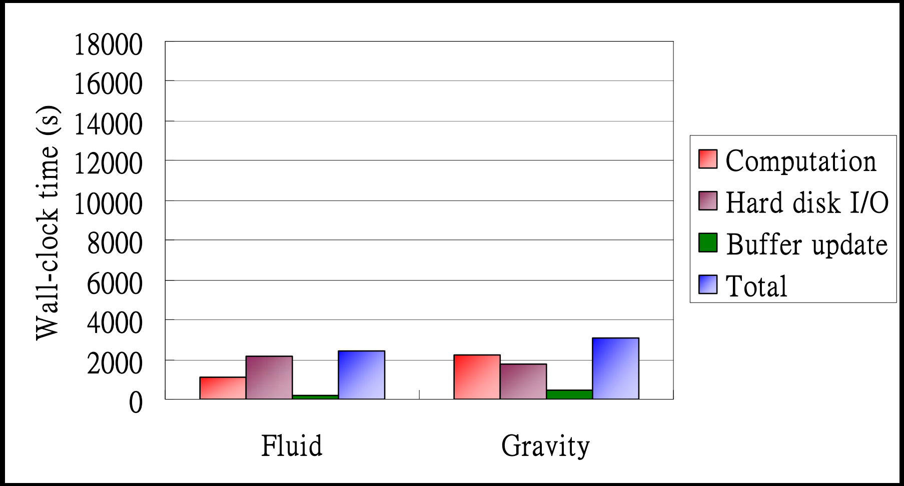
Performance II : AMR

- Root level: 512^3
- 5 refinement levels
- Effective resolution: **$16{,}384^3$**
- **Total memory requirement: ~ 100 GB**
  - 12.5x larger than the ram in our prototype system
- Decomposed into **4^3 OOC ranks** in a single node

CPU



GPU



Future Work

- More physics
  - I want to write my own MHD code
  - Dark matter particles
  - Cooling, feed-back, radiation transfer …
- Out-of-core computation
  - Optimization
  - Multi-node test
- OpenMP + MPI + GPU
  - Fully exploit the computing power of a single node
- OpenCL
- Open source

Conclusion

- GAMER : GPU-accelerated Adaptive-MEsh-Refinement Code
  - GPU hydrodynamic and Poisson solvers
  - Parallelized (multi CPUs + multi GPUs)
  - A framework of AMR + GPUs → general-purpose, flexible
  - 16x faster than CPUs (N GPUs vs. N CPU cores in NAOC)
  - Ref : Schive, H-Y., et al. 2010, ApJS, 186, 457
- Optimizations
  - Concurrency of memory copy and kernel execution
  - Concurrency of CPU work and GPU work
- Out-of-core
  - Increase the simulation size : 10x ~ 100x
  - Small-scale GPU cluster vs. large-scale CPU cluster