Can memory-bound code benefit from the great power of GPUs?

——Acceleration of a 3D high-order Finite-Difference WENO Scheme for Large-Scale Cosmological Simulations on GPU

Chen Meng[1], Long Wang[1], Zongyan Cao[1],XianFeng Ye[1],Long-Long Feng[2]

1. Supercomputing Center, Computer Network Information Center, Chinese Academy of Sciences
2. Purple Mountain Observatory, Chinese Academy of Sciences

What is WENO?

Weighted essentially non-oscillatory (WENO) is a high order finite-difference method, based on structured grids, designed for shock capturing. It has been widely used in applications for high-resolution supersonic flow simulations, typically the cosmological hydrodynamic involving both shocks and complicated smooth solution structures. Consider the system of hyperbolic conservation laws, 3D Euler equation for the fluid without any viscous:

$$U_t = \frac{\partial f(U)}{\partial X} + \frac{\partial g(U)}{\partial Y} + \frac{\partial h(U)}{\partial Z} = F(t,U)$$

$$U = \begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ E\end{pmatrix} \quad f(U) = \begin{pmatrix}\rho u\\ \rho u^2+P\\ \rho uv\\ \rho uw\\ u(E+P)\end{pmatrix} \quad g(U) = \begin{pmatrix}\rho w\\ \rho uw\\ \rho vw\\ \rho u^2+P\\ w(E+P)\end{pmatrix} \quad h(U) = \begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+P\\ \rho vw\\ v(E+P)\end{pmatrix}$$

We used 5th order WENO scheme for the discretization of the fluxes in double precision, for example:

$$\frac{\partial f(u)}{\partial x}\Big|_{x=x_j} \approx \frac{1}{\Delta x}(\hat{f}_{j+1/2} - \hat{f}_{j-1/2}) \qquad \hat{f}_{j+1/2} = w_1\hat{f}^{(1)}_{j+1/2} + w_2\hat{f}^{(2)}_{j+1/2} + w_3\hat{f}^{(3)}_{j+1/2}$$

The success of WENO scheme relies on the design of the nonlinear weights w_i , which is the key to achieve automatically high-order accuracy and non-oscillatory property near discontinuities. The calculations of w_i takes most of the resources on hardware and time. And WENO computation becomes memory-bound derived from it.

Problems

The high-order and high-resolution of WENO scheme call for more computations. The advances of GPUs open new horizons to further accelerating WENO in large-scale cosmological simulations. But there are several challenges:

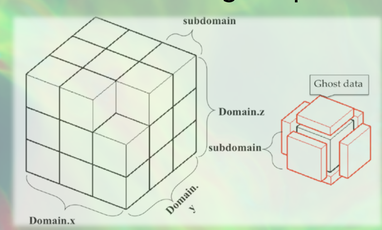
- Big data for 3D problem
- Double precision for high accuracy
- More ghost data for high order
- More memory access for calculations of weights.

High requirements for memory space and bandwidth seem not to be welcomed by GPUs. Can WENO computations benefit from GPUs? The answer is "YES!". Now our implementation has been applied for "Wigeon", which is a hybrid cosmological simulation software based on WENO scheme.

Implementation on CPU/GPU

- Decomposition based on MPI

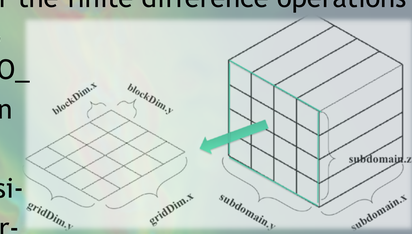
We used a hybrid parallel mode that involves CPUs and GPUs co-processing. The large computational workload should be suitably distributed among the processors first. 5th order WENO scheme requires a five-point stencil at least in the each direction, which is called the ghost data.



We subdivided the domain along each of three axial directions to achieve the smallest amounts of ghost data on the same scale and the best scalability. Then GPU processes the main WENO computations in the subdomain on one process.

- Mapping strategy on GPU

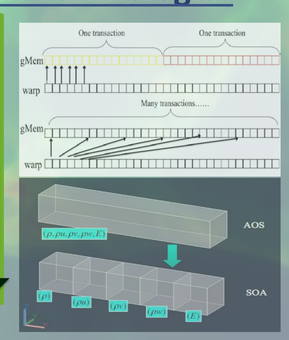
The subdomain distributed on each process would undergo a second decomposition on the local GPU. The whole WENO part is split into three kernels for the finite difference operations in X-axis, Y-axis, Z-axis. For WENO_fx, each point on the Y-Z plane of subdomain is assi--gned on a differ--ent thread in a different block and corresponds to the processing of a group of all points along X-axis. The parallel strategy for WENO_gy, WENO_hz is in the same but along the different axis directions.

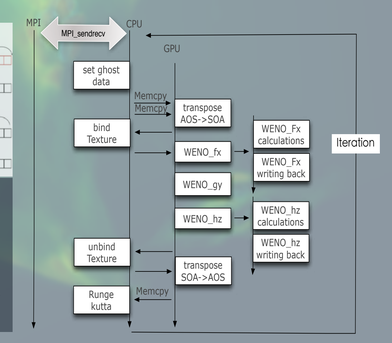


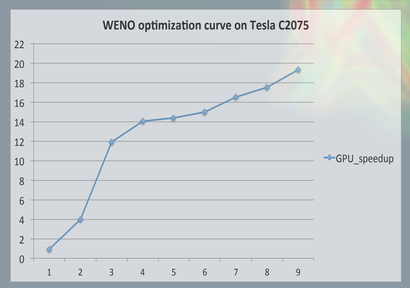
Optimization Technologies

Optimizations

- Memory throughout
  - Address pattern in global memory
  - Hierarchical memory
- Instruction throughout
- Latency





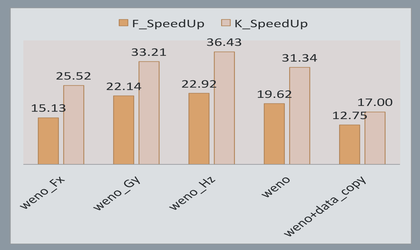


1、Initial Kernel
2 Consolidate 'if' statements and unroll small 'for' bodies
3、Global memory coalescing by transposing array of structure(AOS) to structure of array(SOA)
4、Using Texture memory for read-only data in the procedure of WENO computations
5、Selecting optimal block-size by multi - running
6、Selecting optimal compiler arguments, like "-Xptxas –dlcm =cg" to shut down L1 cache
7、Control use of registers to reduce "register spilling"
8、Split big Kernel to multi-kernels
9、Using Shared Memory for new kernels

Results

Speedup of 128^3 WENO scheme on single GPU

- C2075 with Fermi architecture
- K20m with Kepler architecture



Euler solver based on WENO scheme on multi-CPU/GPU

28 iterations

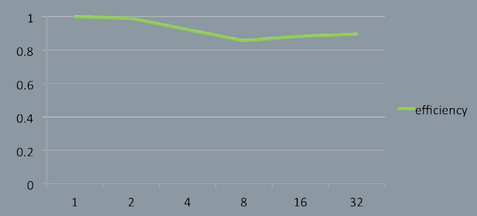
CPU:AMD Phenom(tm) 9850

GPU:C2075 with Fermi architecture

- The strong scaling of 256*256*128

NP	MPI(s)	MPI+ CUDA(s)	MPI efficiency	MPI+CUDA efficiency
1	671.14	70.56		
2	340.26	41.35	98.62%	85.32%
4	188.36	24.30	89.07%	72.60%
8	97.69	15.56	85.87%	56.68%

- The weak scaling of 128*128*128 per process



Future work

1. We can reduce the data-copy time by porting all the computations related to the data in WENO computations to GPU.
2. We can hide the MPI-communication time by overlapping it with the computing time. The ghost data can be updated first. Then, we proceed the communication of the ghost data and the left data updating simultaneously.
3. We can develop a hybrid algorithm that both CPU and GPU take part in computing simultaneously. In some supercomputers (Titan,Tianhe-1A), one node is equipped with a lot of CPU cores but only one GPU. It is important that how to distribute computational task between CPU and GPU for good load balancing.