Paper reviews

Sarah

December 13, 2022

1 URANOS: a GPU accelerated Navier-Stokes solver for compressible wall-bounded flows

1.1 Context

- Solver for high-fidelity modeling of comopressible wall flows
- Massively parallel GPU-accelerated
- Based on modern highfidelity and high-resolution discretization strategies for time-accurate compressible flow predictions
- Provides
  - 6 different convective scheme implementations
  - a cutting-edge method for viscous terms treatment
  - 3 different frameworks for turbulence modeling (DNS, LES, WMLES)
  - A high-order FD approach (from 2nd -¿ 6^{th} order spatial accuracy)
- Combines multiple 3D MPI parallelization strategies with the open standard, OpenACC, f or machine wide, on node, and on GPU parallelism
- GPU version is ocmpared to the CPU only
- Validation through several benchmarks
- Solver handling compressible NS system of equaitons in a 3D Cartesian framework from low to high Mach and Reynyolds numbers conditions.
- Solver handling a wide range of complex wall-flows problems using DNS, wall-resolved, and wall-modeled LES approaches.
- DNS requires high computational costs for high-Reynolds flows.
- LES only solves the largest turbulent scales. But requires direct solutions for the near-wall boundary layer with resolutions comparable to DNS.
- Number of grid points estimated to resolve the near-wall eddies for DNS and WRLES is about $N_{DNS} \sim Re^{37/14}$ and $N_{WLRES} \sim Re^{13/7}$
- Approach to reduce the computational cost associated with LES is WMLES: solves directly the isotropic/turbulence-homogeneous flow away from the wall as in a classical LES framework, while modeling the close-to-wall regions with a greatly simplified method to reduce computational costs compared to a WRLES. N_{DNS} is a linear funciton of Re.
- GPGPUs are orers of magnitude more efficient than standard CPU-only architectures (both at the computational and energy-consumption levels).
- CFD community struggles building a general-purpose solver suitable for all-flows applications in a wide range of Reynolds and Mach numbers.

- URANOS developed for simulations of wall-bounded flow configurations and exploits the OpenACC paradigm (GPU acceleration configured as a standard GPU-enabling framework fully independent fo the computing architecture).
- URANOS combines DNS/LES/WMLES algorithms with advanced numerical methods for convective and viscous terms discretization.

1.2 Numerical methods

High-Order FD approach matching for both uniform and non-uniform Cartesian structures

1.2.1 Convective fluxes

URANOS implements 6 different convective schemes.

- A central, zero-dissipative, 6^{th} order fully-split convective Energy-Preserving (EP) method to deal primarily with shock-free or smooth flows
- 3 increasingly high-order WENO mehtods
- 2 low-dissipative Targeted Essentially Non-Oscillatory (TENO) approaches.

1.2.2 Shock detection

Shock-capturing reconstructions restricted just around the shocks/shocklets locations, and therefore letting the EP method to deal with smooth flow regions. Three dfferent implementations

- Density-gradient-based detector
- Density-jump formulation
- Ducros sensor

1.2.3 Viscous fluxes discretization

- Incompressible terms with a semi-consevative approximation using a HOFD approach.
- Incompressible terms can be trated with a fully conservative approach
- ???

1.2.4 Numerical treatment of temporal components

- 3^{rd} order TVD low-storage Runge-Kutta method
- Time step computing according the CFL and Fourier cirteria.

1.3 Acceleration

- GPU porting
- Different approaches vary in terms of their degrees of portability, adaptability and computational performance
- Things to consider: the initial cost fo code development & the long term maintenance costs
- Better to have a single code base which targets different architectures
- Rather than programming with vendor-specific languages, the programmer can focus on accelerate in a vendor-neutral manner (which allows OpenACC paradigm).
- The compiler transforms directives into device-specific application code.

1.3.1 OpenACC & MPI

- Profiling the code: identifying the hotspots with the condition that porting this region should amortize the expense of the data transfer to the GPU.
- After initialization, solver data migrates to the device.
- H2D/D2H data movements reduced as much as possible (`data directives`)
- Having a **single** data region outside the main time loop that manages all the H2D and D2H operations.
- `parallel` construct is preferrred rather than `kernels` as it allows the user more control, in particular when it comes to loop granularity.
- It enables controlling the loop granularity through the clauses *(gang, worker, vector)*
  - Coarse-grained parallelism (*gang*)
  - Fine-grained parallelism (*worker*)
  - Single Instruction Multiple Data level (*vector*)
- `collapse` clause allow unifying all the iteration of **nested** loops in a single one

- MPI: the standard for inter-node data transfers
- Transition to a multi-GPU logic is not straightforward and hardware-unrelated.

1.3.2 Acceleration performance analysis

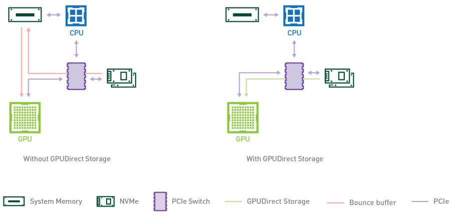
1.4 Validation and Results

1.5 Conclusion

2 Glossary

Time-accurate: That can provide solutions to the full unsteady equations. The time step is used through the grid.

GPU-Direct/non-Direct: NVIDIA GPUDirect is a set of technologies. It improves data movement and access for NVIDIA data center GPUs. We count several technologies as GPUDirect Storage, GPUDirect Remote Direct Memory Access (RDMA), GPUDirect Peer to Peer (P2P) and GPUDirect Video.


CPU
GPU
Without GPUDirect Storage
CPU
GPU
With GPUDirect Storage
System Memory
NVMe
PCIe Switch
GPUDirect Storage
Bounce buffer
PCIe