CFD Primer: Introduction to Computational Fluid Dynamics

Author: Kevin Tubbs

In a series of blogs, I will layout the foundation of what Computational Fluid Dynamics (CFD) is and how it applies to High Performance Computing (HPC) applications. I will cover how these foundations are used to develop CFD solutions using HPC.

What is CFD?

CFD uses numerical methods and algorithms to solve and analyze the equations that govern fluid dynamics. Fluid dynamics has applications in a wide range of science and engineering fields. Generally, CFD is associated with computers with large amounts of memory and high processing speeds. These computer systems also require storage and interconnect systems capable of storing, transferring and examining the large amounts of data associated with the computed results. CFD codes are composed of three main stages: (1) preprocessing, (2) simulation or flow solver, and (3) post-processing.

HPC can be viewed in two categories, capability and capacity. HPC capability computing uses the maximum compute power to solve a single large problem in the shortest amount of time. HPC capacity uses the most efficient configuration to solve multiple complex problems simultaneously. This can be either a small number of large problems or a large number of small problems. HPC CFD users generally aim for maximum capability, capacity or some combination of both. HPC computing becomes necessary when the time to final solution needs to be reduced or the problem size is too large to fit in the memory of one machine. Other cases for HPC in CFD include design or engineering problems that require multiple realizations of involving parameter sweeps, providing simulation resources for small to large user groups, and real-time computing. Time to final solution with CFD solvers greatly influences the rate at which analysis and design decisions can be made. Because there are a wide variety of methods and uses, HPC components play an important role in the delivery of efficient HPC tools for scientist and engineers.

Three stages of CFD

The preprocessing stage consists of the defining of the geometry of the region of interest, selecting the physics being modeled, selecting a numerical method for solving the physics and the mapping of governing equations to a computational domain. Numerical methods in CFD can be separated into traditional methods such as Finite Difference Method (FDM), Finite Element Method (FEM) and Finite Volume Method (FVM) and Discrete Simulation or Particle methods which include lattice gas automata (LGA), the lattice Boltzmann equation (LBE), discrete velocity methods (DVM), dissipative particle dynamics (DPD), smoothed-particle hydrodynamics (SPH), direct simulation Monte Carlo (DSMC), stochastic rotation dynamics (SRD), molecular dynamics (MD), and hybrid methods. The choices selected results in a set of algebraic equations that replace the differential equations governing the fluid flow. The algebraic equations need to be solved using different algorithms or solvers.

The simulation stage is the computational intensive part. For the majority of traditional CFD methods, the algebraic equations created are solved using matrix equations in the form Ax=b. The resulting matrices are usually sparse matrices. Best practices in CFD are to use iterative solvers. Solvers are chosen based on convergence rate and parallel efficiency. Simulation parameters, choice of solvers, and numerical schemes all affect the time to solution.

The post-processing stage consists of visualizing and analyzing data stored during the simulation in order to understand the physics and/or make engineering decisions. The post-processing stage is used to massage the data and show the results in graphical and easy to read format. For large CFD datasets, remote and parallel data processing may be required for visualization since the data would be too large to send across the network in an efficient manner. In some cases, multiple realizations of simulations may need to be processed and interpreted.

Performance

The performance of a CFD application is generally measured by the time it takes to complete the simulation. This is commonly converted to a more general measurement, rating, which is the number of simulations that can be run in a day. The performance of a CFD application on a HPC system is measured by its parallel performance or speedup. Speedup can be defined in terms of execution time or rating. It is defined by the formula $S_p = T_1/T_p$, where p is the number of processors, T_1 is the execution time of the sequential application, and T_p is the execution time of the parallel application with p processors. Linear speedup is considered ideal and is obtained when parallel speedup is equal to the number of processors. For example, doubling the number of processors for an application with linear speedup will double the speed or decrease the execution time by one-half. Another useful parallel performance metric is the ratio of the speedup to number of processors or efficiency defined as $E_p = S_p/p = T_1/(pT_p)$.A common task in measuring the parallel performance of an application on multi-node clusters is measuring the scalability. This measurement indicates how efficient an application is when using increasing numbers of parallel processing elements (CPUs, cores, GPUs, etc.). There are two basic ways to measure the parallel performance of an application, depending on whether the application is CPU-bound or memory-bound. For CPU-bound applications, strong scaling is used to evaluate how the solution time varies with the number of processors for a fixed total problem size. For memory-bound applications, weak scaling is used to evaluate how the solution time varies with the number of processors for a fixed problem size per processor. Strong and weak scaling are application and study specific. The performance can also be compared to power consumption or price to determine best configurations for HPC clusters for CFD. Stay tuned for my follow-up blogs, where I will cover what we are doing with CFD solutions in High Performance Computing (HPC) applications and talk about Dell CFD solutions and technologies being developed today.