Calibrating Parallel Computers and Interconnects for CFD++ Applications

Sukumar Chakravarthy and Thomas Milford*
Metacomp Technologies, Inc.
28632 Roadside Drive Suite 255
Agoura Hills, California 91301, USA
Email: src@metacomptech.com

Key Words: Parallel Computing and Communications, Unstructured Grid CFD, Scalability, Performance Estimation

ABSTRACT

Computational Fluid Dynamics (CFD) simulations increasingly exploit parallel computing and high speed interconnects to perform very large scale simulations for analysis, design and troubleshooting. The computing hardware and the interconnect hardware are constantly evolving and new hardware configurations are being deployed with greater frequency. In such an environment, CFD users are finding it very difficult to determine how to match CFD software with parallel computing configurations. In this paper, we describe a simple but powerful and automated approach we have taken to help CFD++ (a specific CFD software suite) users determine a few parameters that help calibrate the use of CFD++ (in any chosen mode of operation such as compressible perfect gas, real gas, incompressible flow, etc.) on any given parallel computing platform.

The few parameters mentioned here help to determine the performance of CFD++ for any given mesh, given computing platform and given mode of operation, but without having to actually perform the simulation. Knowing these parameters provides a predictive capability to estimate the performance. The parameters themselves are determined by actually running an automated tool, devised for this purpose, once (or a few times). This tool automatically generates a trial mesh in a distributed manner for a certain number of processes (corresponding to processors, CPUs, etc.), runs CFD++ in the selected mode of operation and measures certain timings. It can automatically repeat this process for different grid sizes and different numbers of processes if a maximal set of values is specified. From the measured performance, this tool will automatically determine these parameters.

The choice of the performance parameters is based on an idea first developed by the first author more than a decade ago but tested then only for a single block structured grid. In that original work and in this paper, the approach is based on the simple idea that the time spent on work done by a single process of a cooperating set of parallel processes includes the following components:

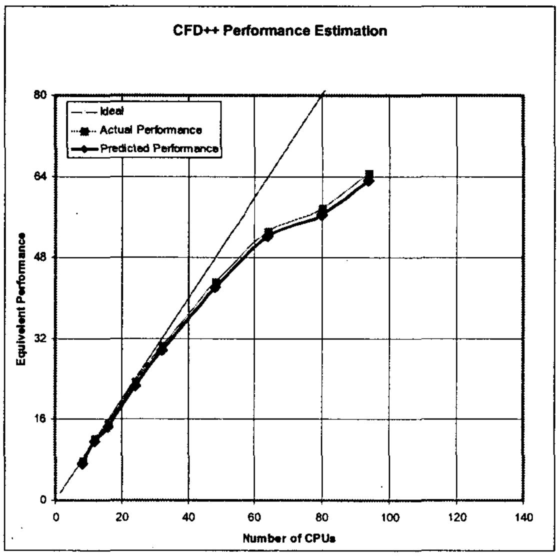
a) Time spent on work done at every cell
b) Time spent on work done at cells that are at a physical boundary
c) Time spent on work done at cells that are at a process boundary (CPU boundary); this is principally the time spent in sending and receiving messages from other processes.

Our simple parameterization then postulates that the consolidated CPU time, T, for an individual process can be expressed as

$$T = a + b*B + c*C$$

C represents the number of cells in each CPU and B represents the number boundary cells in the same CPU. The coefficients of performance are a, b, and c. The first one represents a latency effect that is not related to any number of cells. The second parameter represents a proportionality constant primarily related to the amount of inter-process communications. The third parameter represents a proportionality constant primarily related to computational work common to all cells.

Once these three parameters have been estimated by fitting three cases done with different B and C values (as a result of different domain decompositions for 3 different choices of the number of processes to apply), the same a, b and c can be used to estimate performance. An example is illustrated in the figure below. We will show many more examples in the full paper corresponding to different operational modes of CFD++ and how the prediction made by using this simple approach is valid for very different grids and computer hardware and interconnect types.



REFERENCES

S.R.Chakravarthy and S.Palaniswamy, "Some Aspects of Single-Zone Structured-Grid CFD For a Hypercube MIMD Computer," presented at the Conference on Unstructured Scientific Computation on Scalable Multiprocessors, held October 29-31, 1990, Nags Head, North Carolina; sponsored by ICASE.