573380

The following paper will be presented at:

Parallel CFD '96, Capri, Italy, May 20-23, 1996

Sponsors: AEREA (Association for European Research Establishment in Aeronautics), CONVEX, ECCOMAS, ENEA, IBM, Silicon Graphics

Organized by C. I. R. A.

Title:

Real-Time Visualization of an HPF-based CFD Simulation

Mark Kremenetsky
Silicon Graphics, Inc.
Kremenet@sgi.com

Arsi Vaziri
Numerical Aerodynamic Systems Division
NASA Ames Research Center
vaziri @nas.nasa.gov
&
Robert Haimes
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
haimes@orville.mit.edu

Note from Arsi Vaziri, the NASA author, with respect to this 427 request: The work presented here is an integration of two existing computer programs: a CFD solver written in High Performance Fortran (HPF) by Mark Kerementsky of SGI, Inc. and a parallel visualization package, pV3, written by Robert Haimes of M. I. T. Both software systems have been previously published independently in various forms. The current work extend these codes by providing additional subroutines to transfer the solution data to the visualization system.

Abstract

Current time-dependent CFD simulations produce very large multi-dimensional data sets at each time step. The visual analysis of computational results are traditionally performed by post processing the static data on graphics workstations. We present results from an alternate approach in which we analyze the simulation data in situ on each processing node at the time of simulation. The locally analyzed results, usually more economical and in a reduced form, are then combined and sent back

HPF uses special distribution statements to create distributed arrays. The elements of distributed arrays are scattered over the nodes of the virtual computer. The visualization and
computing tasks need to exchange data. Ideally, HPF should have a standard mechanism so that routines not written in HPF can query the data distribution. If this were the case, the
pV3 API could be written directly to handle the data in-place.

As a first step, we have examined the "quasi-serial" connection between visualization and solver parts. Here, we copy a distributed CFD solution after each time step to the scalar array on one of the processors and then through the pV3_Update function running only on this thread we send the data to the pV3 kernel. Although this approach is simple, it does not achieve a reasonable performance because of serialization of communications between the two parts of the application. In this case the visualization is performed serially on the same designated processor.

In our second implementation we have used an intrinsic procedure approach to provide an effective parallel communication between graphics and computing tasks and also to utilize the ability of pV3 to do multi-threaded co-processing. In this approach, pointers to the parts of the memory containing the solution data are passed to the visualization program. The global grid address of the local data on each processor is calculated and send to the visualization program for a seamless reconstruction of the solution data.

A data-parallel Navier-Stokes solver was implemented for testing the pV3 package and HPF compiler (from the Portland Group, Inc.) on the Silicon Graphics Inc. Power Challenge cluster array at NASA Ames Research Center. This code computes three dimensional transonic transient laminar viscous flow encountered in a blast wave. The initial configuration consists of a high pressure and density region at the center of a cubic cell with the low pressure gas at rest. Periodic boundary conditions are applied to the array of cubic cells which form an infinite network.