European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS 2004
P. Neittaanmäki, T. Rossi, K. Majava, and O. Pironneau (eds.)
K. P. Estola and G. Di Bona (assoc. eds.)
Jyväskylä, 24–28 July 2004

HOW TO MAINTAIN EFFICIENCY ON VECTOR AND SMP PLATFORMS FOR LARGE AERODYNAMIC CALCULATIONS

J. F. Boussuge⋆, S. Champagneux⋆, G. Chevalier⋆, L. Giraud⋆, F. Loercher⋆, M. Montagnac⋆

⋆CERFACS, 42 Avenue G. Coriolis, 31057 Toulouse Cedex, France
e-mails:boussuge@cerfacs.fr, champagneux@cerfacs.fr, chevalier@cerfacs.fr, giraud@cerfacs.fr, loercher@cerfacs.fr, montagnac@cerfacs.fr

Key words: Parallel aerodynamic simulation, large calculation, vector processor, cache based processor, OpenMP

Abstract. *Large industrial aerodynamic calculations are nowadays performed indifferently on parallel vector computers or on clusters of SMPs. From a software design point of view it is crucial to ensure the best possible sustained megaflops rate on both platforms while for sake of minimal labour effort it is important to maintain only one source code. In this paper we describe software techniques that have been implemented to comply with these constraints. We first show how we succeeded to obtain a sustained megaflops rate that is independent of the mesh size. We then investigate the possibilities of using OpenMP in some of the main time consuming routines to replace the vectorization by fine grain parallelism to better exploit the shared memory available within each node of the SMP computers.*

Numerical experiments are reported on HP-Compaq, IBM SP based on Power 3 and Power 4 where tremendous savings can be obtained at a cost of a very little code change.

1 INTRODUCTION

Nowadays the numerical simulation is fully integrated in the design process of the aeronautics industry. During the last decades, robust models and numerical schemes have emerged and have been implemented in the front-end CFD codes. These codes have been run for many years on high performance computers that were vector processors based platforms. With the advent of modern microprocessors the gap in performance between the vector and "scalar" processors has shrunk. Clusters of SMPs have spread in industry and the large calculations are currently performed indifferently on parallel vector computers or on clusters of SMPs. From a software design point of view this means that the code should be efficient on both types of platforms while from a maintenance viewpoint only one source code must be maintained with as little as possible difference between the final codes that will have to be compiled on the target machine.

In this work we present some of the solutions that have been investigated in the framework of the elsA (ensemble logiciel de simulation en Aérodynamique [2, 3]) project. elsA is an object oriented software of ONERA (`elsa.onera.fr`) jointly developed by CERFACS that solves the compressible three dimensional Reynolds Averaged Navier-Stokes equations in a cell-centered finite-volume formulation for multi-block structured grids. elsA is currently deployed within various Airbus business units for some of their computational fluid dynamics needs.

The elsA software is exploited on different computing platforms ranging from high-end parallel vector computers to clusters of SMP. It was preliminary designed for vector machines, so our first efforts were mainly focused on simple techniques to preserve the efficiency of the code on platforms based on SMP clusters. On these latter computers reusing the data loaded in the high level of the memory hierarchy (level 1, level 2 or level 3 cache or physical memory) is the main concern when trying to ensure high sustained megaflops rate of the computation. We show how based on the knowledge of the main features of the memory hierarchy (number of level of cache, size of the cache, cache management policy, TLB...) it is possible to get rid of some very poor performances observed for some size of the blocks.

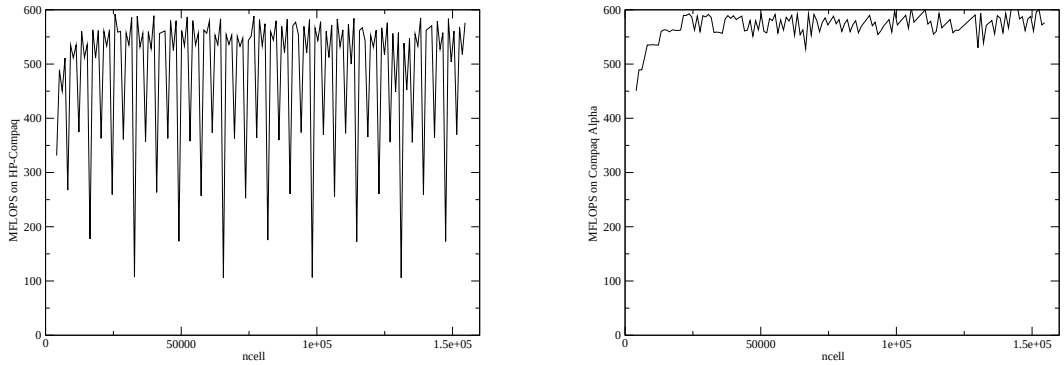
At a coarser level we investigated the possibility to use OpenMP [6] directives to parallelize the vectorial loops.

This paper is organized as follows. In Section 2 we describe how we address the management of the memory in the elsA software. In the next section we illustrate the potential benefit of using OpenMP to parallelize the main time consuming routines of the code. We finish with some concluding remarks.

2 CACHE AND TLB MANAGEMENT

The root of this study is the observation that the performance of the code was very much dependent on the size of the mesh involved in the simulation. This is illustrated in Figure 1 where we report on the megaflops rate to perform the computation on a cube which size varies. It can be observed that this rate can drastically drop for some grid cell numbers up to a factor of 5. It can also be seen in Figure 1 that the location of the peaks is periodic. They correspond to the situation where the limited associativity of either the cache or the Translation Lookaside

Buffer (TLB) becomes the main bottleneck. The first step of the optimization that attempts to



Initial code Code with updated leading dimension

Figure 1: Megafbps rate of elsA on HP-Compaq varying the size of the cube

remove these peaks consists in identifying what level of the memory hierarchy is responsible for that. The methodology consists in using a probing code that mimics the most often occurring memory access pattern. Most of the computation is performed in low level routines written in Fortran. These routines mainly access 2-dimensional arrays, which leading dimensions are the number of cells in the mesh. The number of columns is the number of variables associated with each cell and depends on the type of simulation. This storage and memory access is kwown to be the most efficient one for vector computing.

Using this probing technique we were able to identify that the bottleneck was the L1 cache of the Alpha processor on the HP-Compaq and was the limited number of TLB on the IBM machines. After this step, we have changed the size of the dynamically allocated arrays so as to avoid the memory contention. The change in the size of the leading dimension is computed by a simple function that depends on the number of sets and the size of these sets of the cache or TLB. This idea is very similar to the one used on vector machines to avoid memory bank conflicts and has the main advantage to only require a very minor change in the code. Figure 1 clearly shows the improvement obtained : the megaflops rate is now nearly independent of the number of cells in the mesh.

3 OpenMP: AN ALTERNATIVE TO THE VECTORIZATION ON SMP

Some of the numerical schemes implemented in elsA were selected for their ability to fully take advantage of the vectorization. Many vectorization directives are inserted to help the compilers and we wondered if we could simply replace them by some OpenMP commands. We first focused our attention on one of the main time consuming routine that implements a relaxation scheme [7] either to solve the Jacobian or to be used as the smoother in a nonlinear multi-grid method [4]. The simple replacement of vectorization directives by OpenMP ones gave very

disappointing performance. In particular rather poor speed-ups were observed on large meshes and speed-downs were observed on small meshes. This is far to be optimal in the framework of multi-grid because going down in the grid hierarchy quickly leads to small grids. To get good speed-ups even on small grids we have implemented a 1D block partition of the grid. The relaxation scheme implements a pipelined technique between blocks, each block being handled by a thread. Unfortunately, the resulting code differs quite deeply from the initial code for vector processors. The speed-up performances of the multi-threaded code are reported in Figure 2. It

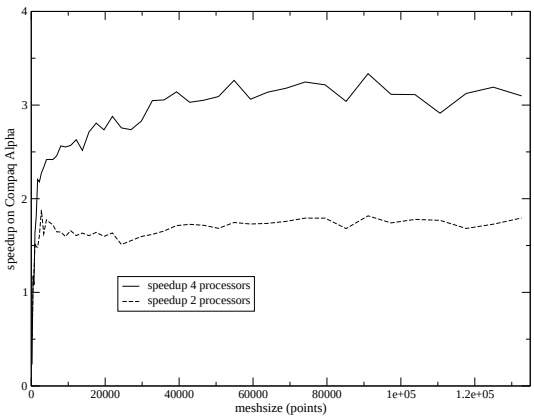


Figure 2: Speed-ups on a 4-processor HP-Compaq SMP node

can be seen that an asymptotic speed-up close to 1.7 is quickly observed when two threads are used. For 4 processors the asymptotic behavior is observed for slightly larger mesh sizes that are still satisfactory for a usage in a multi-grid framework.

4 CONCLUDING REMARKS

In this work we have presented the preliminary results of a feasibility study that aims at developing a single code as efficient on vector processors as on SMP. Part of the techniques we have described have been integrated in the elsA software [5] and are daily used for aerodynamic calculations in Airbus business units. More details on this study will be soon available in [1].

REFERENCES

[1] J. F. Boussuge, S. Champagneux, G. Chevalier, L. Giraud, F. Loercher, and M. Montagnac. Optimization of the elsA software for SMP platforms. Technical report, CERFACS, 2004. in preparation.

[2] L. Cambier and M. Gazaix. elsA: an efficient object-oriented solution to CFD complexity. In 40th AIAA Aerospace Science Meeting & Exhibit, Reno, number AIAA 2002-0108, pages 14–17, january 2002.

[3] M. Gazaix, A. Jolles, and M. Lazareff. The elsA object-oriented computational tool for industrials applications. In *23rd Congress of ICAS*, Toronto, Canada, september 8-13 2002.

[4] A. Jameson. Solution of the Euler equations for two dimensional transonic flow by a multigrid method. *Appl. Math. Comput.*, 13:327, 1983.

[5] F. Loercher. Optimisation d'un code numérique pour des machines scalaires et parallélisation avec OpenMP et MPI. Technical report TR/CFD/03/104, CERFACS, October 2003.

[6] OpenMP Architecture Review Board. OpenMP Fortran Application Program Interface. Technical Report Version 2.0, 2000.

[7] S. Yoon and A. Jameson. A LU-SSOR scheme for the Euler and Navier-Stokes equations. In *AIAA 25th Aerospace Sciences Meeting*, number AIAA-87-0600, Reno, Nevada, USA, january 12-15 1987.