MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1534 April, 1995

Submitted to ACM/IEEE Supercomputing 95

Aeroacoustics on Non-Dedicated Workstations

Panayotis A. Skordos
pas@ai.mit.edu

This publication can be retrieved by anonymous ftp to publications.ai.mit.edu.

ABSTRACT

The simulation of subsonic aeroacoustic problems such as the flow-generated sound of wind instruments is well suited for parallel computing on a cluster of non-dedicated workstations. Simulations are demonstrated which employ 20 non-dedicated Hewlett-Packard workstations (HP9000/715), and achieve comparable performance on this problem as a 64-node CM-5 dedicated supercomputer with vector units. The success of the present approach depends on the low communication requirements of the problem (small ratio of communication to computation) which arise from the coarse-grain decomposition of the problem and the use of local-interaction methods. Many important problems may be suitable for this type of parallel computing including computer vision, circuit simulation, and other subsonic flow problems.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-92-J-4097 and by the National Science Foundation under grant number 9001651-MIP.

1. Introduction

The use of workstations for parallel computing is a viable and powerful approach. Workstations are widely available and relatively inexpensive because the technology is driven by a strong market. A number of supercomputers have been built using workstation-type technology combined with a suitable communication network. At the same time, the idea of exploiting clusters of workstations for parallel computing has been attracting more and more attention and is growing in popularity.

One of the challenges of exploiting a cluster of workstations for parallel computing is to design the computation appropriately to match the communication capacity of the cluster, which is usually limited as in the case of a shared-bus Ethernet network. Another challenge is to exploit clusters of workstations which are *not* dedicated; namely, it would be nice if the workstations can be used concurrently by a parallel application and by the regular “owners” of the workstations for text-editing and other small tasks. In this paper, both of the above issues are discussed.

First, an important class of problems is identified which is highly suitable for parallel computing on a cluster of workstations. This is the area of subsonic computational fluid dynamics (CFD) or simply subsonic aeroacoustics. Then, a prototype distributed system is described which includes automatic process migration, and successfully exploits a cluster of 25 non-dedicated Hewlett-Packard workstations (HP9000/715). In terms of performance, 20 non-dedicated workstations achieve comparable performance on simulations of aeroacoustics as a 64-node CM-5 dedicated supercomputer with vector units. To demonstrate practical use, the present distributed system is also applied to solve a real problem, the simulation of wind instruments. Performance measurements of the distributed system as well as representative simulations of wind instruments are presented.

2. A suitable class of problems

A workstation cluster can be viewed as a distributed-memory multiprocessor with small communication bandwidth and high latency. Accordingly, numerous small messages between the processors must be avoided, and few large messages are preferred. Further, a computation which can be decomposed at a coarse-grain level to reduce the communication requirements is a better match for a workstation cluster than a fine-grain parallel computation. Suitable problems which possess the above characteristics include problems with local interactions and spatial organization. When such problems are decomposed into subregions, the communication-to-computation ratio is proportional to the surface-to-volume ratio of the subregions. Because of locality, only the boundaries of the subregions need to be communicated between processors. Thus, one can increase the size of the subproblems to reduce the communication-to-computation ratio, and to match the communication capabilities of the cluster.

Problems with local interactions and spatial organization can be found in computer vision, in circuit simulation using waveform relaxation methods (reference [1]), in simulations of subsonic aeroacoustics, and possibly other areas. Aeroacoustics simulations are the focus of this paper. As we will see below, very good results can be achieved on a cluster of about 20 workstations linked together by a shared-bus Ethernet network. Further, the present computations have practical value as they solve real problems in subsonic CFD; they are not abstract computations which simply demonstrate good performance.

Aeroacoustic simulations involve the numerical solution of a set of partial differential equations (PDE). All PDE-based problems employ a numerical grid (spatial organization) to discretize the equations, and a numerical method to calculate the future values of variables defined on the grid. There are basically two classes of numerical methods for solving PDEs: explicit methods which employ local interactions only, and implicit methods which lead to matrix equations and non-local interactions. Although explicit methods are local and very simple to program, they are usually avoided because they have the disadvantage of requiring small time steps for numerical stability.

Aeroacoustic simulations are special among other PDE-based problems in that they are well suited for explicit numerical methods. In particular, the simulation of subsonic flow and acoustic waves requires small time steps to follow accurately the acoustic waves step by step. Namely, the time step must be comparable to the grid spacing divided by the speed of sound, which produces a very small time step in the case of subsonic flow. Thus, there is a match between the requirements of the problem and the requirements of explicit numerical methods for small time steps. This match encourages the use of explicit methods and makes aeroacoustic simulations very suitable for parallel computing on a cluster of workstations.

3. The distributed system

It is straightforward to develop a distributed system for solving spatially-organized local-interaction problems on a cluster of workstations. The present distributed system has been developed directly on top of UNIX and TCP/IP utilizing also the facilities of a clustered HP-UX environment such as file-locking semaphores and a common-file-system. General programming environments such as PVM (reference [2]) have not been used in this work because the goal is to experiment with new ideas and a prototype system. The present distributed system consists of four modules:

- initialization of the global problem,
- decomposition into subregions,
- job submission to free workstations, and
- job monitoring including the automatic process migration from busy hosts to free hosts.

The job submission and job monitoring are performed by one workstation which can be thought of as the "master", while the other workstations are the "slaves". The slaves calculate their assigned subproblems independently at every integration step, and then communicate the boundaries of their subproblems with their neighbors, and then the cycle repeats. The communication step enforces a partial synchronization between neighbors. More details on the behavior of the system and the implementation can be found in reference [3]; here, only the main design ideas are outlined.

The basic ideas which are responsible for the success of the present distributed system are as follows. First, the small ratio of communication-to-computation has already been mentioned earlier, and will be discussed again in the next section. Second, the system utilizes a fixed number of workstations (fixed static decomposition of the problem). Third, the system utilizes typically about 4/5 of the total number of non-dedicated workstations which are available in the cluster; namely, 20 out of 25. This strategy simplifies things, and enables the migration of a parallel subprocess from a workstation which becomes busy to a free workstation when necessary. Other approaches which vary dynamically the load per workstation and the number of workstations (for example, the idea of "work stealing" in Blumofe&Park, reference [4]) are worth exploring, but they have a disadvantage for the particular problem at hand. Namely, such approaches would require a finer decomposition of the problem into many small tasks to be allocated dynamically, and this would increase the communication overhead. By contrast, the present approach of using large coarse-grain subproblems and allocating one subproblem per workstation is very simple, has small overhead, and has produced very good results in practice.

Regarding migration, the automatic process migration in the present system is successful and straightforward because the parallel subprocesses know how to handle migration requests. In particular, there is a global synchronization signal which is used before a migration to instruct all the processes to continue running until the start of some integration step. When this step is reached, the processes that need to migrate save their state on disk and exit, while the remaining processes pause and wait for a signal to continue the computation. A monitoring program finds new workstations to submit the migrating jobs, and then instructs all the processes to continue. Each process migration is not particularly fast (it lasts about 20-30 seconds), but migrations do not happen too often. In the present system (20 out of 25 non-dedicated workstations), there is approximately 1 migration every 40 minutes on the average, and a typical simulation of aeroacoustics lasts about 48 hours. Thus, the simple approach works well in practice.

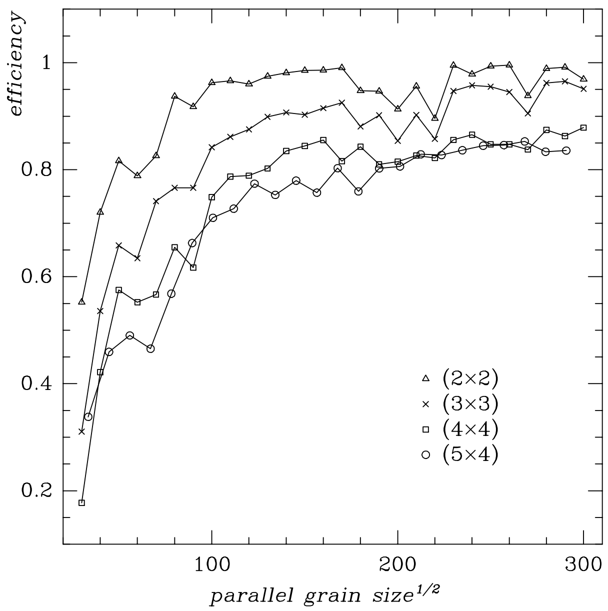

Figure 1: Parallel efficiency of 2D simulations.

4. Computational performance

As stated earlier, a cluster of 20 non-dedicated Hewlett-Packard workstations (HP9000/715) and a shared-bus Ethernet network achieve comparable performance on simulations of aeroacoustics as a 64-node CM-5 dedicated supercomputer with vector units. A comparison was done by measuring how many hours it takes to solve the same problem by the 20 workstations and by the 64-nodes CM-5. The CM-5 was programmed using the C* programming language, and the size and the geometry of the problem (grid of 800x600 fluid nodes) was fixed and known at compile time; otherwise, the performance of the CM-5 degrades. The 64-node CM-5 and the 20 workstations achieved roughly the same performance.

The above result is not surprising because each HP9000/715 workstation is 3-4 times faster than the individual processors of the CM-5. Therefore, if the communication is not a bottleneck, the cluster of 20 workstations has comparable computational power as the 64-node CM-5. Indeed, as we will see below, the communication takes only 20/100 of the total running time of a cluster of 20 workstations, while 80/100 of the time is spent on computation. One last comment regarding the comparison between the cluster and the CM-5 is that the comparison should not be taken too far because other problems which have high communication requirements would not run efficiently on the cluster, but would run efficiently on a parallel computer such as the CM-5 which has a powerful communication network.

Figure 1 shows the efficiency (speedup/processors) of simulations of subsonic flow as a function of grain size for 2x2, 3x3, 4x4, and 5x4 decompositions (triangles, crosses, squares, circles). The horizontal axis plots the square root of the number of fluid nodes in each subregion which is assigned to each workstation. We see that good performance is achieved in two-dimensional simulations when the subregion per processor is larger than 100x100 fluid nodes. This is because the ratio of communication to computation (the relative overhead) decreases as the size of the subregions increases. The poor performance for very small subregions (abrupt drop in performance) is expected because the Ethernet network has high latency and a disproportionate cost for small messages. In these tests, the lattice Boltzmann numerical method (reference [5]) is used, which is a recently-developed explicit method for subsonic compressible flow. Similar results are obtained using traditional finite difference methods. A detailed description of the numerical methods and the measurements can be found in reference [3].

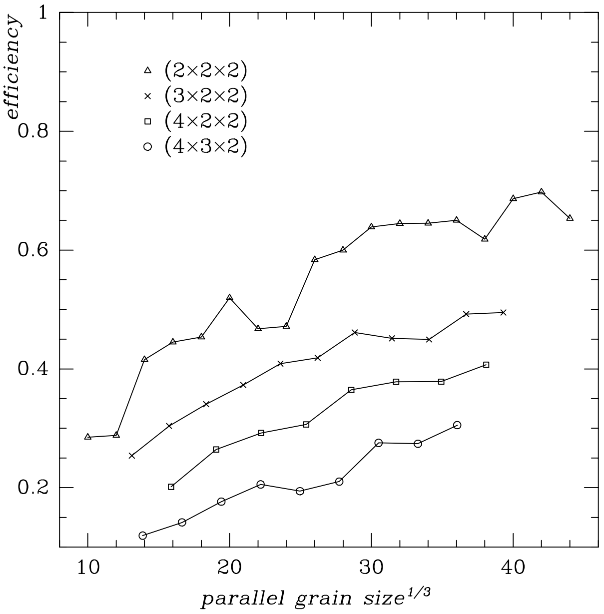

Figure 2: Parallel efficiency of 3D simulations.

A limitation of the present simulations of subsonic flow is that although two-dimensional simulations perform very well on the shared-bus Ethernet network, three-dimensional simulations perform rather poorly. This can be seen in figure 2 which plots similar data as figure 1 for 3D simulations and for 3D decompositions. The size of the subproblems per processor is comparable between 3D and 2D (the largest size 44x44x44 in 3D is very close to the largest size 300x300 in 2D). These sizes are dictated by practical considerations, the run time of the computation and the memory of the workstations. In principle, extremely large subregions per processor would achieve high efficiency in 3D, but they are not practical, and they are not considered here. Instead, our recommendation for 3D simulations is to improve the communication network using FDDI, ATM, or simply an Ethernet switch which provides virtual dedicated connections between pairs of workstations.

5. Simulations of wind instruments

The distributed system has been applied to simulate directly the flow of air and the generation of tones in wind instruments using the compressible subsonic flow equations. Although physical details are not given here (reference [6] and [7]), a few representative results of the simulation of a soprano recorder are shown in figures 3 and 4. In the first figure, we can see the two-dimensional geometry of the soprano recorder, and the decomposition into 22 workstations (dashed lines). This picture is a snapshot of the simulation about 30 ms after the initial blowing of air into the recorder. The flow of air and the generated vortices are plotted using iso-vorticity lines. About 0.8 million fluid nodes are used in this simulation. Figure 3 shows a magnified view of the jet-labium region, and we can see the jet of air oscillating at a frequency of about 1100 Hz and generating a musical tone. It is worth noting that physical measurements of the acoustic signal generated by the recorder are in satisfactory agreement with the predictions of the simulations (reference [7]).

6. Conclusion

Problems with local interactions and spatial organization are well suited for parallel computing on a cluster of workstations. The simulation of subsonic CFD (aeroacoustics) has been identified as a particularly good example because the nature of the problem matches the computing requirements of the cluster of workstations. Further, a simple approach of automatic process migration has been described which allows the exploitation of a cluster of 25 non-dedicated workstations. The distributed system has been applied successfully to perform direct simulations of the aeroacoustics of wind musical instruments. Apart from aeroacoustic problems, there are probably many other PDE-based problems which are suitable for parallel computing on a cluster of workstations. By combining computer science with other disciplines, the computer technology can be better matched with the physical applications.

References

1. A. Lumsdaine, J. M. Squyres, and M. W. Reichelt, “Waveform iterative methods for parallel solution of initial value problems,” in *Proceedings of the Second Scalable Parallel Libraries Conference*, (Mississippi State, MS), pp. 88–97, IEEE Computer Society Press, 1994.

2. V. S. Sunderam, “A framework for parallel distributed computing,” *Concurrency: Practice and Experience*, vol. 2, no. 4, pp. 315–339, December 1990.

3. P. Skordos, “Parallel simulation of subsonic fluid dynamics on a cluster of workstations,” in *Proceedings of High Performance Distributed Computing 95, 4th IEEE Int'l Symposium, Pentagon City, Virginia*, August, 1995.

4. R. Blumofe and D. Park, “Scheduling large-scale parallel computations on networks of workstations,” in *Proceedings of High Performance Distributed Computing 94, San Franscisco, California*, pp. 96–105, 1994.

5. P. Skordos, “Initial and boundary conditions for the lattice Boltzmann method,” *Physical Review E*, vol. 48, no. 6, pp. 4823–4842, December 1993.

6. P. Skordos, *Modeling flue pipes: subsonic flow, lattice Boltzmann, and parallel distributed computers*. Department of Electrical Engineering and Computer Science, MIT, Ph.D. Dissertation, January 1995.

7. P. Skordos and G. Sussman, “Comparison between subsonic flow simulation and physical measurements of flue pipes,” in *Proceedings of ISMA 95, International Symposium on Musical Acoustics, Le Normont, France*, July, 1995.

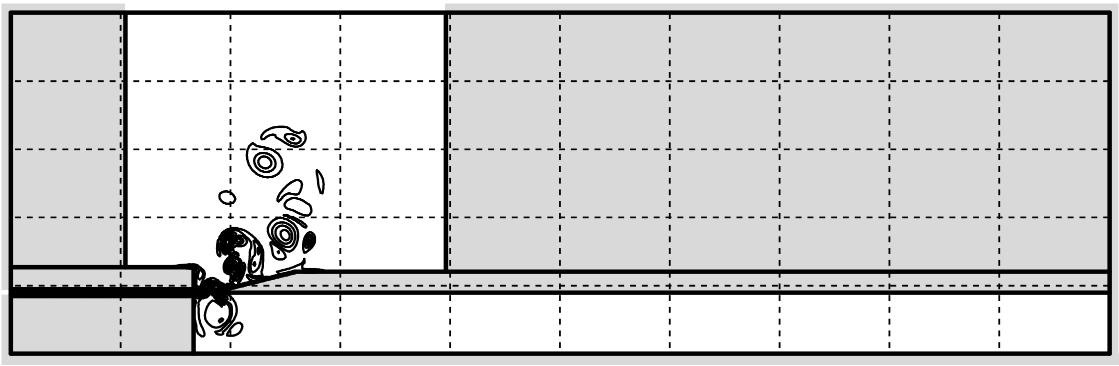

Figure 3: Simulation of a 20 cm closed-end soprano recorder. Iso-vorticity contours are plotted. The decomposition is shown as dashed lines, and 22 workstations are used. The gray-shaded areas are not simulated.

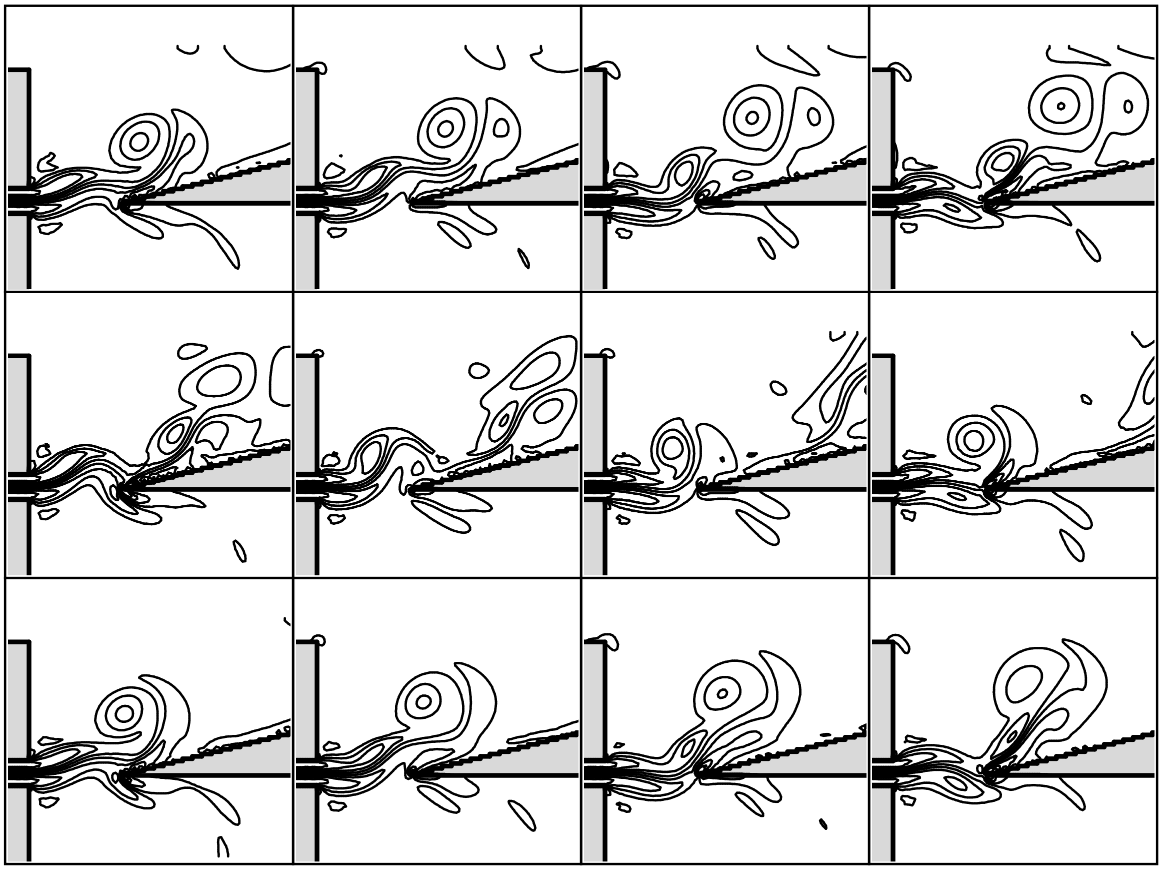

Figure 4: Jet oscillations of the 20 cm closed-end recorder at blowing speed 1104 cm/s. Frames are 0.22 ms apart, from left to right. Iso-vorticity contours are plotted. 35.6 ms after startup.