DEVELOPMENT AND APPLICATIONS OF A MODULAR PARALLEL PROCESS FOR LARGE SCALE FLUID/STRUCTURES PROBLEMS

Guru P. Guruswamy
NASA Advanced Superomputing (NAS) Division,
Ames Research Center

SUMMARY

A modular process that can efficiently solve large scale multidisciplinary problems using massively parallel supercomputers is presented. The process integrates disciplines with diverse physical characteristics by retaining the efficiency of individual disciplines. Computational domain independence of individual disciplines is maintained using a meta programming approach. The process integrates disciplines without affecting the combined performance. Results are demonstrated for large scale aerospace problems on several supercomputers. The super scalability and portability of the approach is demonstrated on several parallel computers.

INTRODUCTION

During the last decade significant progress has been made in the area of supercomputing using parallel computers and it has started making impact on major engineering fields such as aerospace design. The aerospace community that was one of the main driving forces behind the supercomputing technology using serial computers is again playing a major role in adapting parallel computers for its ever increasing computational needs. Because of the large effort required to restructure softwares, particularly in the area of multidisciplinary applications using high-fidelity equations, there is a latency in using parallel computers in day-to-day use for analysis and design of aerospace vehicles. This paper presents a technology that leads the parallel computers based supercomputing to the real world aerospace applications.

Large scale multidisciplinary problems are common in engineering design.
They involve coupling of many high-fidelity single disciplines. For example, aeroelasticity of large aerospace vehicles that involve strong coupling of fluids, structures and controls is an important element in the design process[1]. Fig. 1 illustrates a mission critical instability that can occur for a typical space vehicle. The instability can occur soon after the launch vehicle gets separated from the aircraft. The phenomenon was dominated by complex flows coupled with structural motions. From the results presented in Ref. 1 it can be concluded that low-fidelity software was not adequate to completely understand the instability phenomenon which involved non-linear flows coupled with structural motions.

Methods to couple fluids and structures by using low-fidelity methods such as the linear aerodynamic flow equations coupled with the modal structural equations are well advanced. Although low-fidelity approaches are computationally less intensive and used for preliminary design, they are not adequate for the analysis of a system that can

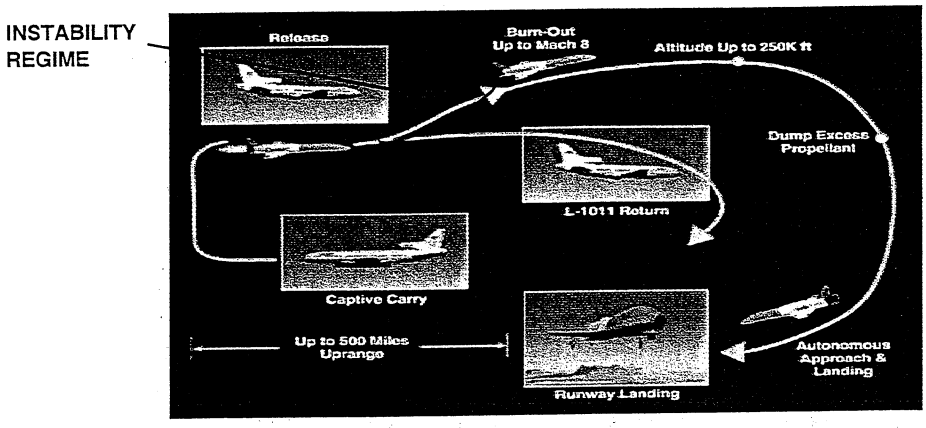

Fig. 1. Illustration of a mission critical aeroelastic instability.

experience complex flow/structure interactions. High-fidelity equations such as the Euler/Navier-Stokes (ENS) for fluids directly coupled with finite elements (FE) for structures are needed for accurate aeroelastic computations for which complex fluid/structure interactions exist. Using these coupled methods, design quantities such as structural stresses can be directly computed. Using high-fidelity equations involves additional complexities from numerics such as higher-order terms. Therefore, the coupling process is more elaborate when using high-fidelity methods than it is for calculations using linear methods. High-fidelity methods are computationally intensive and need efficient algorithms that run on parallel computers. Fig. 2 illustrates the increase in complexity when using high-fidelity approaches.

In recent years, significant advances have been made for single disciplines in both computational fluid dynamics (CFD) using finite-difference approaches [2] and computational structural dynamics (CSD) using finite-element methods (see chapter I of ref. 3). These single discipline methods are efficiently implemented on parallel computers. For aerospace vehicles, structures are dominated by internal discontinuous members such as spars, ribs, panels, and bulkheads. The finite-element (FE) method, which is fundamentally based on discretization along physical boundaries of different structural components, has proven to be computationally efficient for solving aerospace structures problems. The external aerodynamics of aerospace vehicles is dominated by field discontinuities such as shock waves and flow separations. Finite-difference (FD) computational methods have proven to be efficient for solving such flow problems. Parallel methods that can solve multidiscipline problems are still under development. Currently there are several multidiscpline parallel codes that solve a monolithic system of equations using unstructured grids[4] mostly modeling Euler flow equations. This single computational domain approach has been in use for several years for solving fluid-structural interaction problems[5]. There were several attempts to solve fluid-structural

interaction problems using a single FE computational domain (see Chapter 20 of Ref. 5). While using the single domain approach, the main bottleneck arose from ill-conditioned

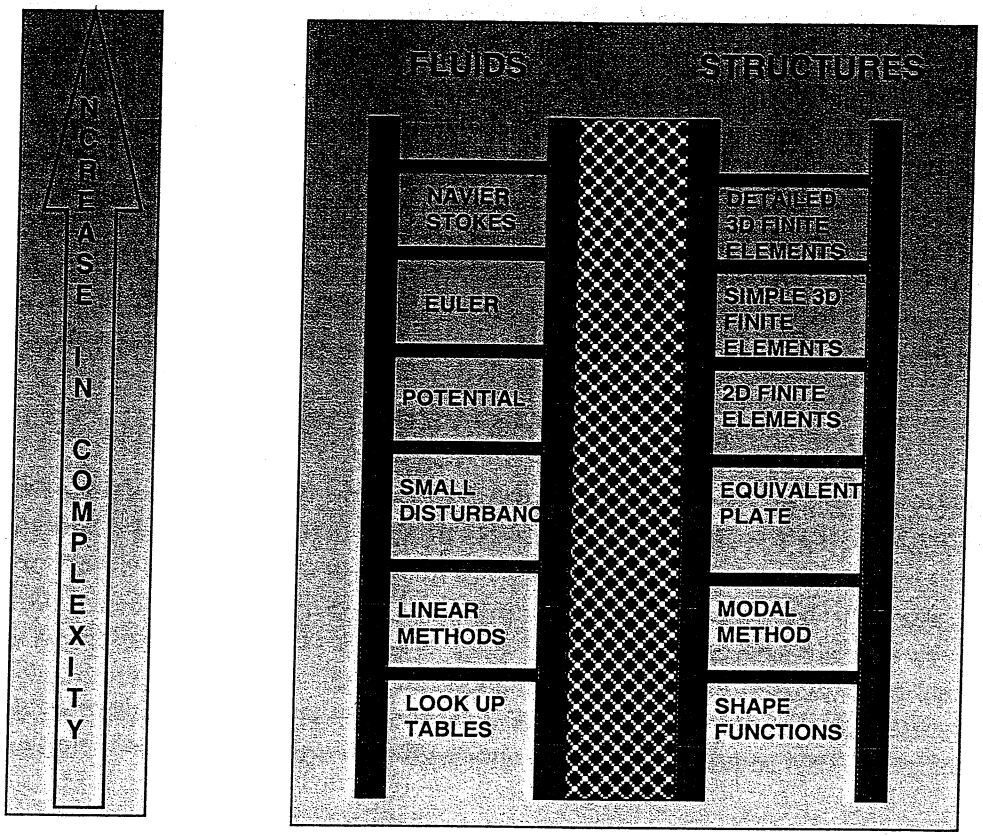

Fig. 2. Increase in simulation complexities in physics/geometry of aerospace vehicles.

matrices associated with two physical domains with large variations in stiffness properties. The drop in the convergence rate from the rigid case to the flexible case in Ref. 6 indicates the weakness of the single domain approach. As a result, a sub-domain approach is needed where fluids and structures are solved in separate domains and solutions are combined through boundary conditions.

This paper presents an efficient alternative to the monolithic approach. The approach in this work is based on a domain independent approach that is suitable for massively parallel systems. Fluids and structures disciplines are interfaced through discipline-independent wrappers.

DOMAIN DECOMPOSITION APPROACH

A method highly suited for state-of-the-art parallel supercomputers is presented in this paper. When simulating aeroelasticity with coupled procedures, it is common to deal with fluid equations in an Eulerian reference system and structural equations in a Lagrangian system. The structural system is physically much stiffer than the fluid system, and the numerical matrices associated with structures are orders of magnitude stiffer than those

associated with fluids. Therefore, it is numerically inefficient or even impossible to solve both systems using a single numerical scheme (see section on Sub-Structures in ref. 5).

Guruswamy and Yang [7] presented a numerical approach to solve this problem for two-dimensional airfoils by independently modeling fluids using the FD-based transonic small- perturbation (TSP) equations and structures using FE equations. The solutions were coupled only at the boundary interfaces between fluids and structures. The coupling of solutions at boundaries can be done either explicitly or implicitly. This domain-decomposition approach allows one to take full advantage of state-of-the-art numerical procedures for individual disciplines. This coupling procedure has been extended to three-dimensional problems and incorporated in several advanced serial aeroelastic codes such as ENSAERO [8,9] that uses the Euler/Navier-Stokes equations for fluids and modal equations for structures. The main emphasis in this paper is to further develop theses methods for parallel computers using a highly portable and modular approach.

PARALLELIZATION EFFORT

Though significant progress has taken place in high-fidelity single discipline codes such as NASTRAN [10] for structures and OVERFLOW[11]for fluids, the effort to combine these single discipline codes into a multidiscipline code or process is still in progress. Several attempts have been made to expand single discipline codes to multidiscipline codes such as ENSAERO [9], ENS3DE [12], STARS [13] etc.. These codes are tightly dependent on pre-selected individual disciplines. Due to rapid progress that may take place in individual disciplines, freedom is needed to replace individual modules with improved ones. This requires a different approach than traditional code development.

One of the major drawbacks of using codes with high-fidelity methods is the need for large requirements of computer resources, both in memory and speed. The start of the parallel computer technology initiated new ways of solving individual disciplines with scalable performance on multiple processors. The use of the computer industry standard Message Passing Interface (MPI) [14] utility led to successful parallel solution procedure.

In order to couple different discipline domains, communication between domains is accomplished through an interface at the end of each time step. This is achieved by creating inter-disciplinary communicator using an MPI application programming interface (API) called mpi_intercomm_create[15]. For aeroelastic computations that involves fluids and structural domains, the aerodynamic loads are converted into the structural loads through the fluid-structural interface. Furthermore, the structural deformation is passed to the fluid domain through the interface. Then, the surface grid is deformed according to the structural deformation. In addition, control surface deflection computed in a controls domain is superimposed on the deformed surface grid.

The overall communication design is shown in Fig. 3. In using the MPI library, a communicator is used to identify a group of processors where a processor can communicate with others within the same group. Each group is represented by a box defined by dashed lines as shown in Fig. 3. In this case, however, only one processor is assigned to each group for a single coupled analysis. All the allocated processors have a

common communicator called mpi_comm_world as shown in Fig. 3. The MPIAPI, mpi_comm_create, creates a distinct communicator, denoted as mpirun_app_com for each group of computational processors when it loads the executable program onto the processors. Using the mpirun_app_com communicator, any processor can communicate with others within a group. Communications are also defined using the MPIAPI mpi_intercomm_create to communicate between different discipline modules or different groups. They are denoted by solid and dashed lines with arrows, respectively.

Furthermore, the MPI library has the functionality to create a new communicator for a subset of the allocated processors. Communicators for each discipline are defined so that collective operations can be accomplished within a discipline module. Once a communicator for each discipline is defined, it is quite convenient to do a collective operation within a discipline, such as computing lift and drag coefficients. The

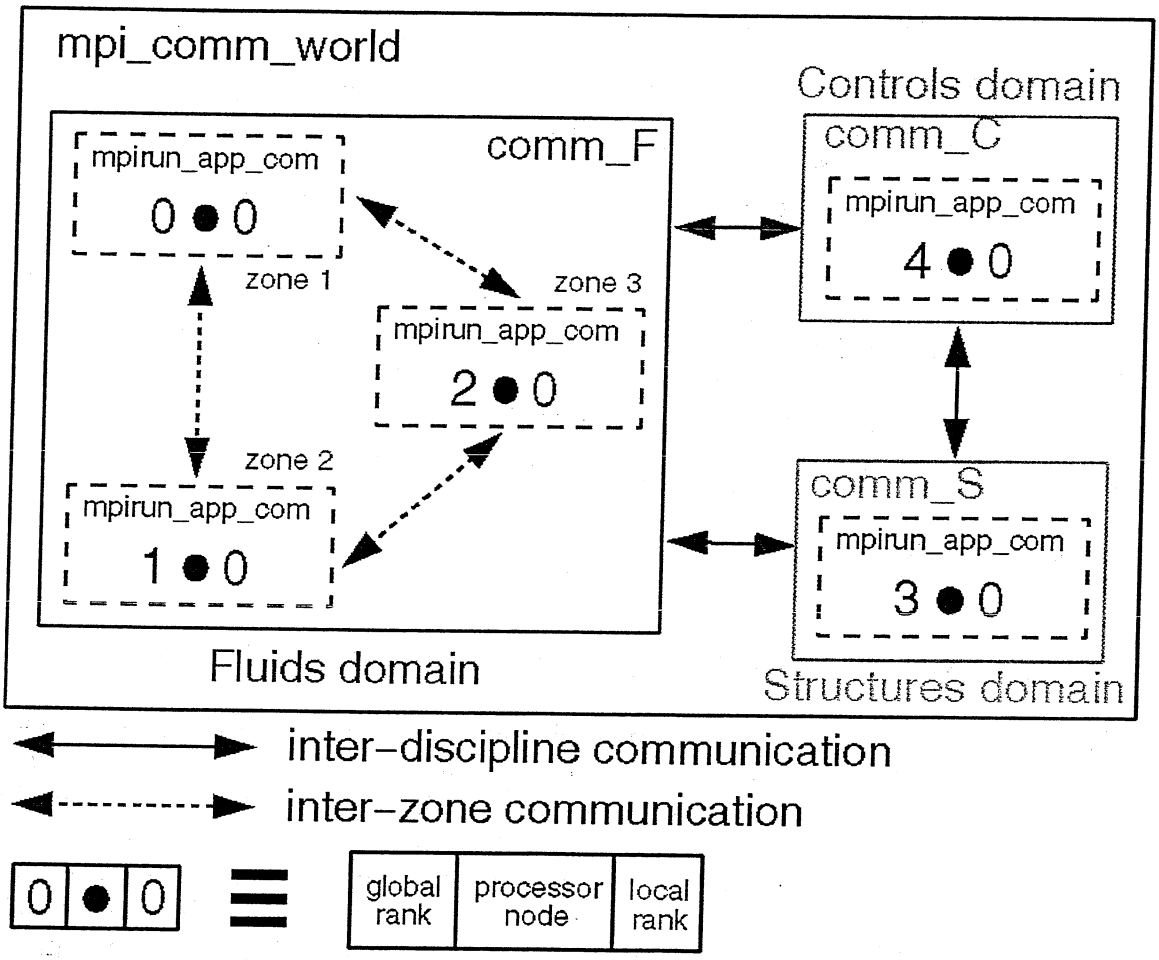

Fig. 3. Data communication design for multizonal applications on parallel computers.

communication design shown in Fig. 3 only explains the coupling of three different computational modules, e.g. fluids, structures, and controls. However, if needed, additional modules can be easily added to the process.

The communication design for a single coupled analysis can be further extended to perform multiple analyses concurrently. Figure 4 shows the extension of the

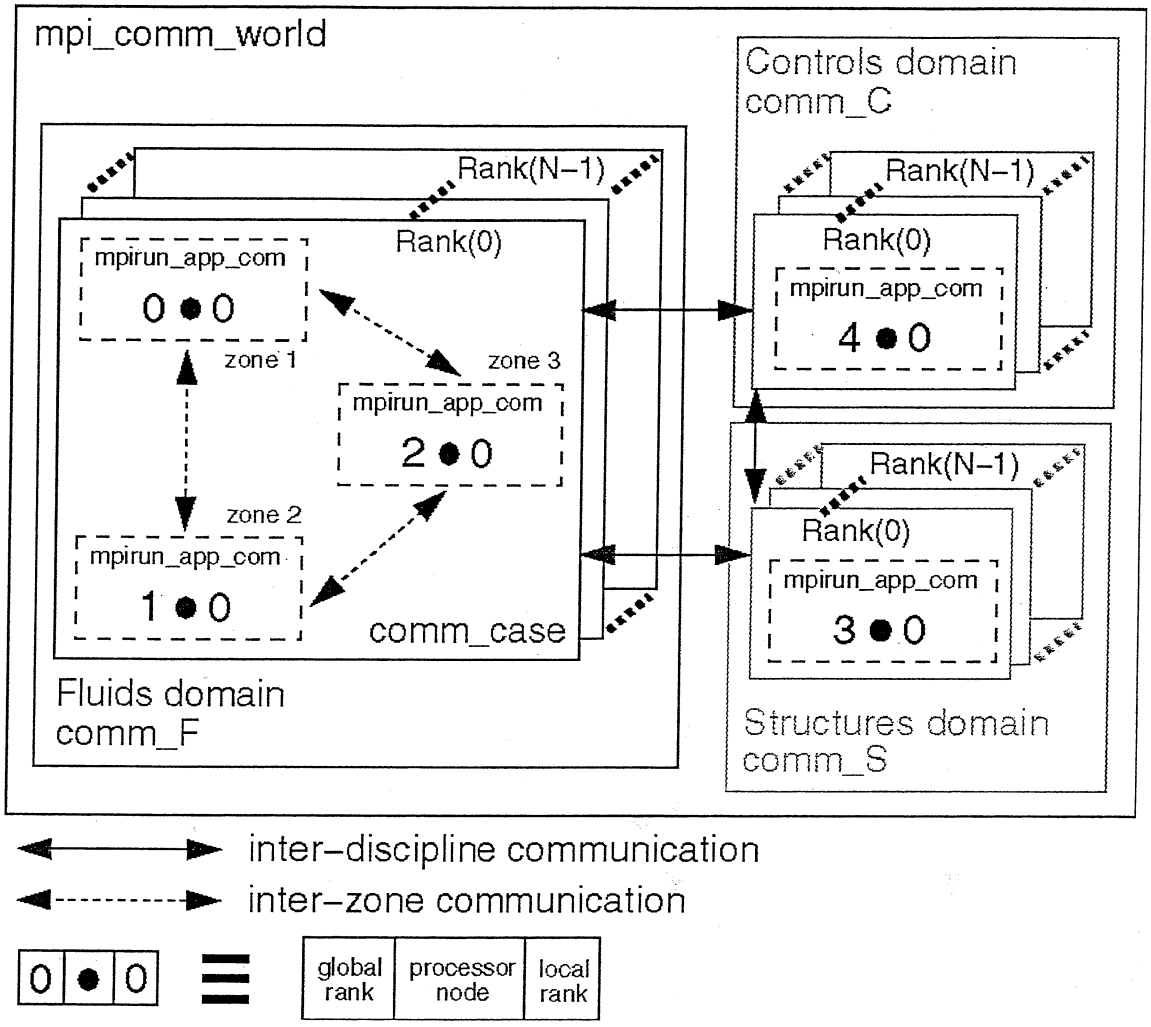

Fig. 4. Multilevel communication among fluids, structural and controls domains.

communication design for concurrent multiple analyses. In contrast to a single coupled analysis, several processors are assigned to each group. In this figure, each group has N processors, which is the number of different cases running concurrently. They are locally ranked from zero to N-1 within a group. In the first run, the initialization data within a group is distributed from the leading processor of each group through a broadcast call using mpirun_com communicator. This makes it easy to distribute initial input data within a group. Once the initial data distribution is completed, each processor of a group will participate in a different analysis. For example, if N cases with different initial angles of attack are concurrently executed, each processor within a group has the same grid data of a zone but computes solutions for the different flow conditions. Within the flow domain, after solving the flow equations at every time step, each zone needs to exchange zonal boundary data with adjacent zones to advance to the next step. For this purpose,

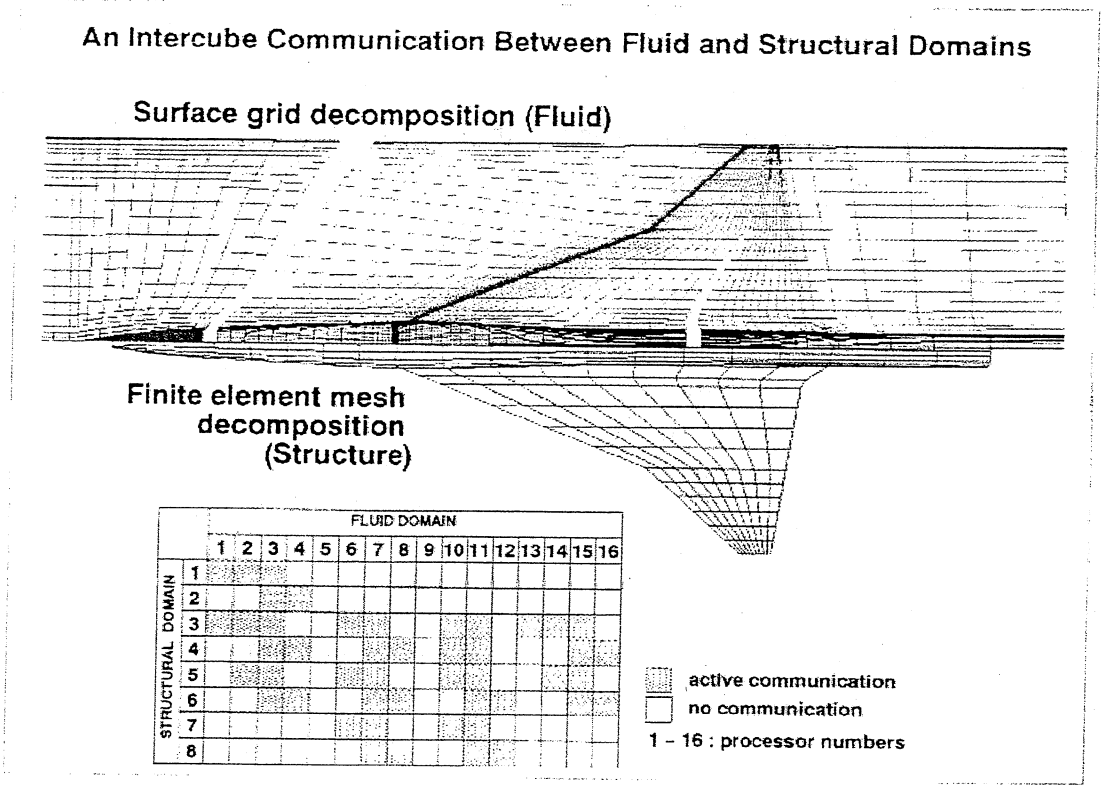

Fig. 5. Typical fluid structures communication on a parallel computer.

data communication is limited only among computational processors with the same local rank. In this communication strategy, each processor can distinguish itself from other processors assigned to different cases. Therefore, each processor having different local rank can participate in different simulations. For multiple multidisciplinary simulations, the same communication strategy is applied for data exchange among the discipline domains. Further details of this process are described in Ref. 16. This high-fidelity multidisciplinary analysis process along with software which includes solution modules and MPI/ MPIAPI library calls is referred to as HiMAP.

A typical fluid structure communication is illustrated in Fig. 5 for an aerospace vehicle . In this case, 16 and 8 processors are assigned to fluids and structures, respectively. The shaded areas show active communication and blank areas show no communication. Active communication takes place where fluid zones are in contact with structural zones.

LOAD BALANCING

Efficient methods to solve fluids and structures commonly use a domain decomposition approach based on zonal or block grids[2]. Each zone may contain a CFD or CSM (Computational Structural Dynamics) grid specific for a component of the full configuration. To efficiently solve complex configurations with large number of varying size grid blocks, a robust load balancing approach is needed. Load balancing can be achieved as follows.

In this work load balancing is achieved by a zone-coalescing and partitioning approach. This parallelization approach achieves the goal of load-balanced execution provided that

there are enough processors available to handle the total number of zones. One-to-one assignment of zones to processors does not guarantee an efficient use of the parallel system. The processors might be working with less than the optimal computational load and performing a lot of expensive inter-processor communications, and hence be data-starved. Both problems are alleviated by introducing a zone-coalescing and splitting capability to the parallelization scheme. In zone coalescing, a number of zones are assigned to a single processors resulting in economy in number of the computational resources and also a more favorable communications-to-computations ratio during the execution. This method was first tried for simple configurations[17] and its general capability is shown in Fig. 6. Figure illustrates that a single zone can be split into several sub-zones or several sub-zones can be merged into a single super-zone depending on the memory available per processor.

In order to obtain maximum performance the above load balancing scheme is further developed. In this scheme which is developed for complex configurations that involve grids with large band width, a further extension of the zone coalescing-splitting approach is implemented.

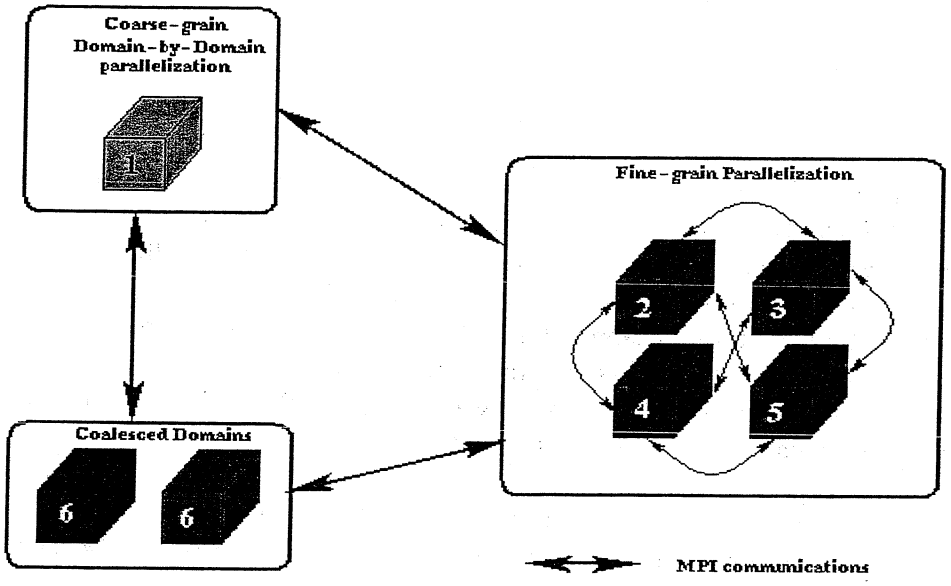

Fig. 6. Zone(Domain) coalescing-partitioning approach.

A number of zones will be assigned to each processor depending on its memory size. For example, it is found that a SGI Origin 3000 processor can handle a maximum grid size of 500K pts for computations using CFD codes such as ENSAERO. The assignment of a zone to processor is started from small zones and progress towards larger zones. In this process any zone that is larger than the maximum size is partitioned. The load balancing scheme used is illustrated in Fig. 7.

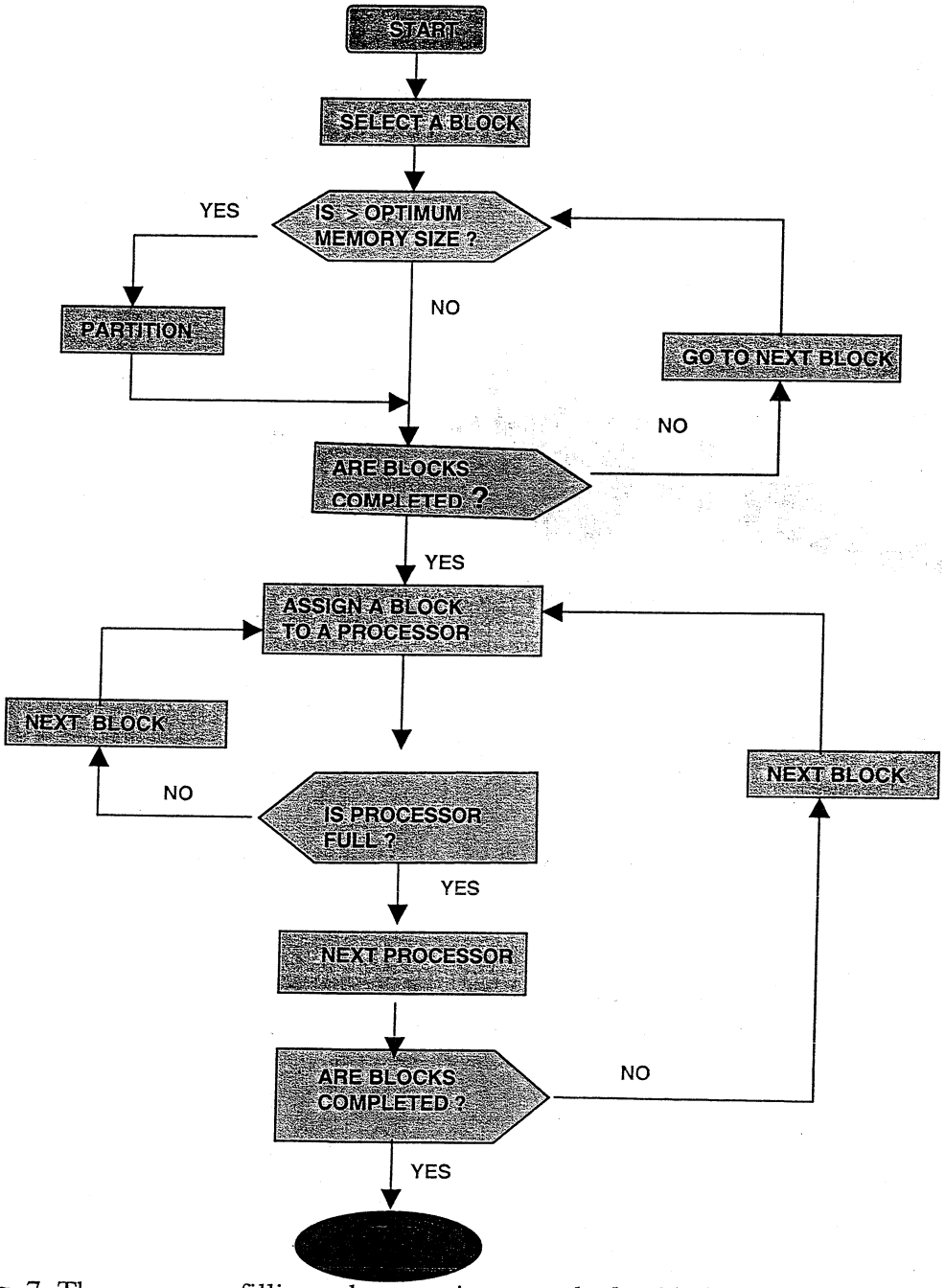

Fig. 7. The processor filling scheme to improve the load balancing.

LARGE SCALE APPLICATIONS

The method presented here is suitable for large scale multidisciplinary analysis. It has been tested using the Euler/Navier-Stokes based flow solver modules such as ENSAERO[9], USM3D[18] and finite element based structures modules such as NASTRAN[10,19]. The method has been demonstrated for large scale aeroelastic applications that required 16 million fluid grid points and 20,000 structural finite elements. Cases have been demonstrated using up to 228 processors on IBM SP2 and 256

processors on SGI Origin2000 computers. Typical configurations analyzed are full subsonic and supersonic aircraft.

An example of a complex multi-block grid system is shown for a typical transport aircraft in Fig. 8. The grid system is made up of 34 blocks and the number of grid points varies from 30K pts to 427K pts per block. If each block is assigned to a processor, the load efficiency of the processor assigned to the smallest block will be about 7%. The load balancing scheme is applied to improve efficiency.

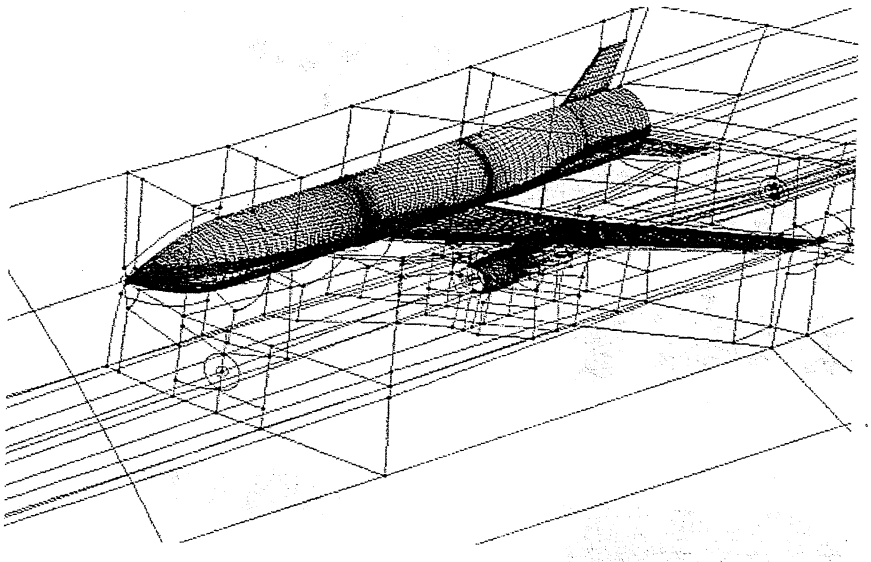

Fig. 8. Complex grid arrangement for a typical transport aircraft

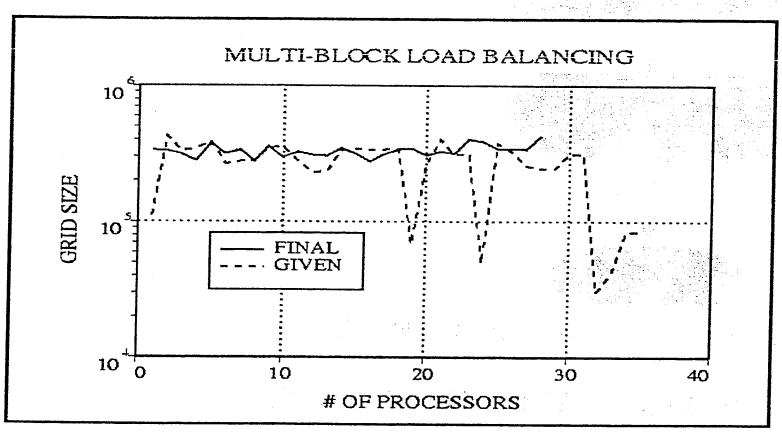

Fig. 9. Grid points per processor with and without processor filling scheme.

Figure 9 illustrates the results of applying the load-balancing scheme to the multi-block grid system shown in Fig 8. The dashed line shows a plot of grid size against the block number. The solid line shows the plot of modified grid size against the regrouped blocks. The number of blocks is reduced from 34 to 28. The ratio of the minimum to maximum block size increased from 7% to 81%. Thus a maximum factor of increase in efficiency per processor equal to 11.6 can be achieved. An efficiency factor E = 1.60 can be computed as a ratio of (average grid size per processor x number of processors) to (average grid size per block x number of blocks).

Parallel computations were made on SGI's Origin 2000 computer. Fig. 10 shows one of the 5 structural modes from the finite element computations of a transport aircraft. Each mode was represented by 2100 degrees of freedom. One O2000 processor was assigned to the modal data. Solutions from HiMAP were obtained using an ENSAERO module[9] along with parallel a MBMG (MultiBlock Moving Grid) [20] moving grid module. A typical aeroelastic solution is shown in Fig. 11. The colors represent pressure coefficients map. The stability and convergence of the GO3D[21] upwind algorithm in ENSAERO module was not affected by re-distribution of patched grids to different processors

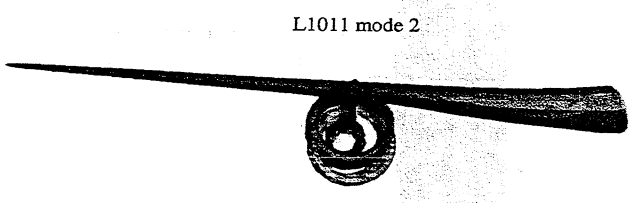

Fig. 10. Typical structural twist mode of an aircraft. (black = original, green = deformed)

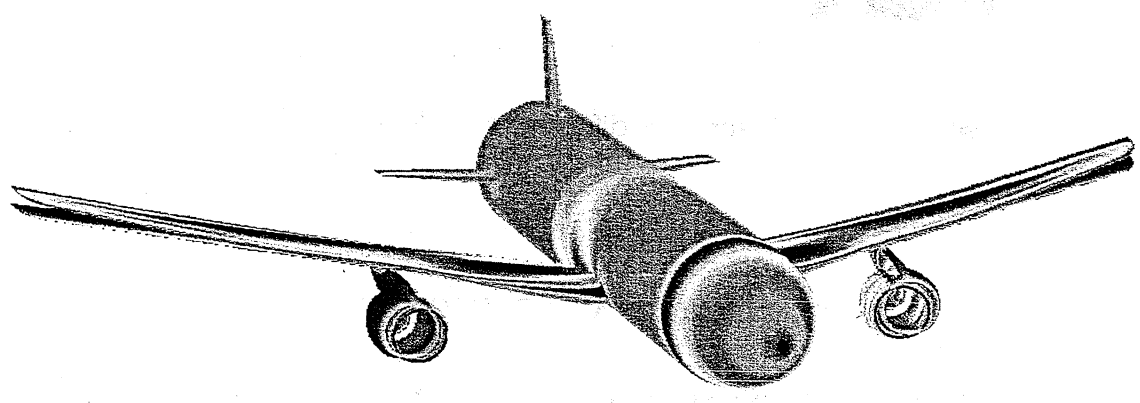

Fig. 11. Pressure coefficient map of a deformed aircraft.

In large scale aerospace problems, grid topologies of the configurations are predetermined based on design needs. Grid size and number of blocks are directly related to the complexity of configuration and fidelity of equations solved. Quite often, parallel efficiency can be addressed only after the grids are designed. The procedure presented here will help for cost effective computations.

Some of the results from applying methods developed here to several large aerospace problems are summarized in Fig. 12. The complexity of the problems increases significantly from a simple wing-body model to full configuration as shown by increase in grid size and number of blocks. The present approach shows a better improvement in efficiency factor E as the complexity of the configuration increases.

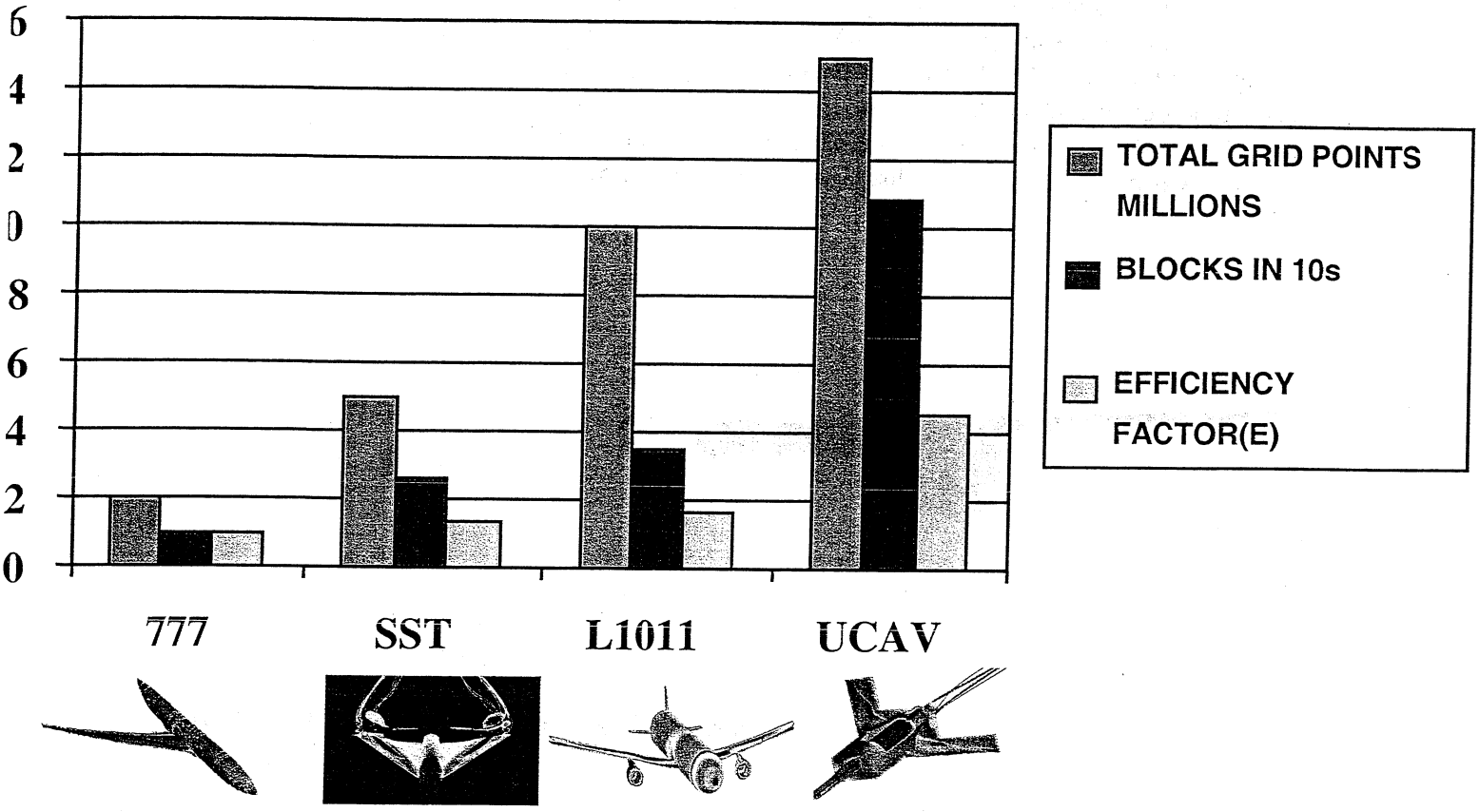

Fig. 12. Parallel efficiency factor for different complexity configurations.

PORTABILITY AND PERFORMANCE

The process developed here is successfully ported to massively parallel processor (MPP) platforms of SGI, SUN and IBM. The optimized flow solver performs at a rate of 120

MFLOPS per processor on Origin 3000 MPP platform. The supermodular capability of HiMAP is demonstrated by plugging in the USM3D unstructured grid solver in place of the patched structured-grid solver and computing aeroelastic responses with minimal effort [18]. In Ref. 18 portability of this software to workstation cluster is also demonstrated. HiMAP can also be used for uncoupled aeroelastic analysis which is embarrassingly parallel [22]. A summary of results on different parallel computer systems is shown in Fig. 13. Almost linear scalability in performance of 3-level parallel HiMAP process was demonstrated on a 256 node IBM SP2 MPP system [23]. Recently the performance and portability of HiMAP is further improved for shared memory configurations by implementing Open_MP communication [24].

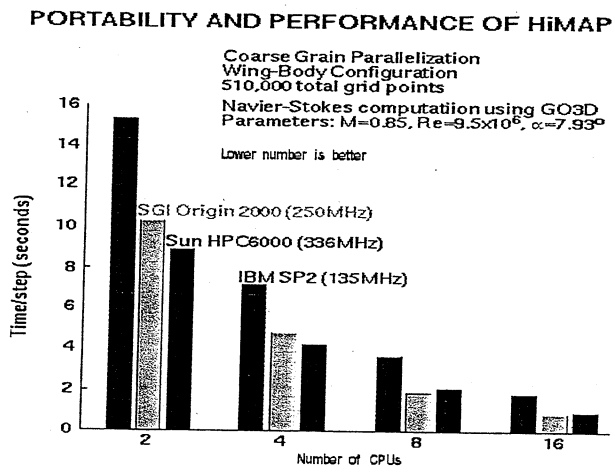

Fig. 14. Demonstration of Portability and Scalability

CONCLUSIONS

An efficient parallel process needed for computationally intensive analysis and design of aerospace vehicles is presented. The process can simulate aeroelasticity of aerospace vehicles using high-fidelity equations such as the Navier-Stokes equations for flows and finite-elements for structures. The process is suitable for both tightly coupled and uncoupled analyses. The process is designed to execute on massively parallel processors (MPP) and work-station clusters based on a multiple-instruction, multiple-data (MIMD) architecture. The fluids discipline is parallelized using a zonal approach while the structures discipline is parallelized using the sub-structures concept. Provision is also made to include controls domain. Computations of each discipline are spread across processors using computer standard message passing interface (MPI) for inter processor communications. MPI based Application Program Interface(API) is developed to run disciplines in parallel. In addition to inter and intra discipline parallelizations, an embarrassingly parallel capability to run multiple parameter cases is implemented using a script system. The combined effect of three levels of parallelization is an almost linear scalability for multiple concurrent analyses that perform efficiently on MPP. Finally this paper demonstrates a first-of-its-kind unique use of the latest parallel computer

technology to the multidisciplinary analysis needed for the design of large aerospace vehicles. The scalable modular approach developed here can be extended for other fields such as bio-engineering and civil engineering.

ACKNOWLEDGEMENTS

Author would like to thank Dr. Chansup Byun of Sun Microsystems and Mr. Mark Potsdam of US Army Aeroflightdynamics Directorate for the consultations provided in writing this paper.

REFERENCES

1.Blades, E. L., Ruth, M., and Fuhrmann, H.D: Aeroelastic Analysis of the X-34 Launch Vehicle. AIAA-99-135, April 1999.
2. Holst, T. L., Flores, J., Kaynak, U., and Chaderjian, N.: Navier-Stokes Computations, Including a Complete F-16 Aircraft. Chapter 21, Applied Computational Aerodynamics, edited by P. A. Henne, vol. 125, Progress in Astronautics and Aeronautics, AIAA, ISBN 0-930403-69-X, 1990.
3. Yang, T. Y.: *Finite Element Structural Analysis*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1986
4. Lohner, R. : Three-Dimensional Fluid-Structure Interaction Using A Finite Element Solver and Adaptive Remeshing. Computing System in Engineering. Vol. 1, Nos 2-4 pp 257-272, 1990.
5.Zienkiewicz, O. C. *The Finite Element Method.* Third edition, McGraw Hill Book Company (UK) Limited, Maindenhead, Berkeshire, England, 1977.
6. Felker, F.F: A new Method For Transonic Aeroelastic Problems. AIAA-92-2123, 1992.
7 Guruswamy, G.P and Yang T.Y. : Aeroelastic Time Response Analysis of Thin Airfoils by Transonic Code LTRAN2. Computers and Fluids, Vol. 9, No. 4 Dec. 1981, pp. 409-425.
8. Guruswamy, G. P.: Vortical Flow Computations on Swept Flexible Wings Using Navier-Stokes Equations. AIAA Jl. , Vol. 28, No. 12, Dec 1990, pp. 2077-2133.
9. Guruswamy, G. P: ENSAERO - A Multidisciplinary Program for Fluid/Structural Interaction Studies of Aerospace Vehicles. Computing Systems Engineering, Val. 1 nos. 2-4, 1990, pp 237-256.
10. NASTRAN User's Manual: NASA SP-222 (08), June 1986.
11. Slotnick, J. P. Kandula, M. and Buning P.G.: Navier-Stokes Simulation of the Space Shuttle Launch Vehicle Flight Transonic Flowfield Using a Large Scale Chimera Grid System. AIAA-94-1860, 1994.
12 Schuster, D., Vadyak, J. and Atta, E.: Static Aeroelastic Analysis of Fighter Aircraft Using a three-dimensional Navier-Stokes Algorithm. AIAA Paper 90-0435, 1990.
13 Gupta, K.K.: Development of Aeroelastic Capability. Jl of Aircraft, Vol 33, No. 5, Sept-Oct 1996.
14. Message Passing Interface Forum, MPI: A Message-Passing Interface Standard, University of Tennessee, May 1994.

15. Fineberg, S.: MPIRUN: A Loader for Multidisciplinary and Multizonal MPI Applications. NAS News, Vol. 2, No. 6, Nov. - Dec. 1994.
16. Guruswamy, G.P, Byun, C., Potsdam, M and Farhangnia, M.: User's Manual for HiMAP - A Portable Super Modular Multilevel Parallel Multidisciplinary Analysis Process. NASA TM-1999-209578, Sept 1999.
17. Hatay, F., Byun, C., Gee, K., Jespersen, D., Rizk, Y. and Guruswamy, G. P.: A multi-level parallelization Concept for High-Fidelity multi-block Solvers, Supercomputer 97, San Jose Nov. 1997.
18. Raj, P., Goodwin, S. A., Sankar, L.N., Weed, R.A., Parikh, P. C. and Bhat, M. K.: Development and Validation of a Fast, Accurate and Cost-Effective Aeroservoelastic Method on Advanced Parallel Computing System. LG97ER0106, Lockheed Martin Report, July 1997.
19. Eldred, L and Guruswamy, G.P.: User's Guide for ENSAERO_FE Parallel Finite Element Solver. NASA TM-1999-208781
20. Potsdam, M and Guruswamy, G.P.: A Parallel Multiblock Mesh Movement Scheme For Complex Aeroelastic Applications. AIAA Paper No. 2001-0716.
21. Obayashi, S., Guruswamy, G. P., and Goorjian P. M: Streamwise Upwind Algorithm for Computing Unsteady Transonic Flows Past Oscillating Wings. AIAA Jl., Vol. 29, No. 10, October 1991, pp 1668-1677.
22. Byun, C. and Guruswamy, G. P: An Efficient Procedure for Flutter Boundary Prediction on Parallel Computer. AIAA-95-1484, 36th AIAA, April 95.
23. Farhangnia, M, Guruswamy, G, and Byun, C: Performance and Applications of HiMAP on Scalable Computers. NASA/CP-1999 208757 Jan 1999.
24. Byun, C. : Porting Experience With HiMAP On Sun HPC Environment. SUN Users meeting, Hawai, April 2002.