High Performance Distributed Environment with Rapid Prototyping Tools to Integrate Parallel Databases, High Performance Fortran, Simulation and Virtual Reality for Unsteady Flow CFD Visualization

Geoffrey C. Fox, Wojtek Furmanski and Ed Boguczc

Northeast Parallel Architectures Center
111 College Place, Syracuse University
Syracuse, NY 13244–4100

Proposal to the NASA Ames Research Center
September 17, 1992

Project Summary

Northeast Parallel Architectures Center (NPAC) at Syracuse University (SU) will develop a rapid prototyping environment to evaluate and integrate new technologies for advanced CFD visualization. These include High Performance Distributed Processing, Parallel Database Management Systems, High Performance Fortran and Virtual Reality. The project will focus on unsteady flow visualization problems, usually associated with immense space-time data sets. The proposed tasks with be carried out at NPAC using NPAC facilities, precomputed flow data sets from NASA Ames and a suitable mix of NPAC and NASA Ames software tools. The project will be cost-shared with a set of planned NPAC programs, aimed at integrating advanced visualization and VR with the present mainstream software environment at NPAC, centered around the Fortran90D project. MOVIE (Multitasking Object-oriented Visual Interactive Environment) System, also developed at NPAC, will provide the software integration and rapid prototyping platform for the proposed effort. Specific R&D issues to be addressed include: a) evaluation of the nCUBE2 based parallel Oracle DBMS in the context of CFD visualization, b) comparison and possible integration of the NASA Ames software tools such as dlib, PLOT3D and SuperGlue with the NPAC software systems, based on FortranD and MOVIE models, c) inclusion of the NASA Ames CFD visualization challenges into the High Performance Fortran benchmarking suite to drive the development of advanced language constructs such as parallel I/0 models.

Contents

1 Introduction 1
2 High Performance Computing Facilities at NPAC . . 2
3 Current and Planned Software Thrusts at NPAC . . 2
3.1 High Performance Fortran Compiler 2
3.2 MOVIE System 3
3.3 Fortran90D/HPF Interpreter 4
3.4 Virtual Reality 4
4 Project Description 5
5 Statement of Work 7
6 References 8
7 Budget 8

1. Introduction

The aim of this proposal is to build a collaborative effort between NASA Ames NAS Lab and NPAC by addressing the complex CFD visualization problems as posed by NAS in terms of the current and planned HPC hardware and software systems at NPAC.

NPAC major strength is in the area of portable scalable software for high performance computing [FJLOSW88]. The first generation of standards in this domain will be based on the High Performance Fortran (HPFortran), which will provide the base platform for a broad class of computationally intensive problems in science, engineering and industry. NPAC is active participant in the HPFortran Forum, where the first draft of the standard is expected by the end of '92, and it also conducts a set of associated research projects with NSF via the CRPC at Rice University and development projects with DARPA. These efforts are centered on the Fortran90D [FHKKKTW90] compiler development and benchmarking. Fortran90D provided the initial thrust towards the HPFortran standard and it plays the role of a language research, prototyping and extension platform.

Another software thrust at NPAC, funded by the US Army, is represented by the MOVIE System [Furm92][FFHNPS92] for High Performance Distributed Computing. MOVIE model is based on a network of MovieScript interpreters, extending Postscript towards Open Systems visualization standards and high performance computing. New planned projects in this area include developing a family of MOVIE based high performance visualization, multimedia and VR servers for terrain and battlefield visualization. Also funded by US Army is the new NPAC project

starting this fall/winter on evaluating and benchmarking the nCUBE2 based parallel Oracle DBMS.

At the initial design stage is the integration project, combining Fortran90D and MOVIE concepts towards the MovieScript based Fortran90D interpreter for HPDC [FoFu92a]. We plan also to use MOVIE to develop object-oriented database layers on to top of parallel Oracle and to design the corresponding Fortran90D interfaces in the form of suitable parallel I/O language constructs.

In parallel with the R&D projects listed above and funded by the federal agencies, we are also exploring the industry needs in the area of high performance computing through the industrial outreach program NYS ACTION, aimed at integrating parallel processing into the New York State industry.

Finally, at the planning stage is the major education program aimed at parallel processing and virtual reality technology transfer to the precollege education. As part of this long term multidisciplinary program we intend to integrate Fortran90D, MOVIE, parallel Oracle and Virtual Reality components towards the high performance high fidelity and high capacity environment ("World Server") for distributed dynamic multi-user visualization and simulation.

In summary, our expertise is in scalable high performance computing and we now plan to integrate advanced visualization and VR into our systems. We believe that by addressing the CFD visualization challenges posed by NASA Ames, we will both benefit from the NASA Ames expertise in the field of advanced visualization technologies and we will also contribute new computational paradigms and software design insights to this class of problems, as proposed in this document.

The organization of the material presented here is as follows. Chapters 2 and 3 discuss, correspondingly, the hardware and software components of the HPC environment at NPAC. In Chapter 4 we describe the proposed project with NASA Ames and in Chapter 5 we quantify it in terms of a set of action items and deliverables. Chapter 7 presents the proposed budget.

2. High Performance Computing Facilities at NPAC

NPAC computing environment is currently based on a network of DEC, Sun and IBM workstations and on 3 modern parallel systems: 32–node CM-5, 32–node NCUB2 and 8K DECmpp. Our local MIMD machines are of modest size, suitable for research and prototype development, and we also have access thought CRPC to larger systems installed at Argonne, Caltech, Los Alamos and Rice University. We also expect soon an upgrade of our environment by the 16–node Intel Touchstone system.

UltraNet hardware has been installed recently on a cluster of Sun workstations at NPAC. The near term research project in the gigabit networking area is to develop the high-speed communication software as well as to design and implement a communication processor that would support a high bandwidth point-to-point communication channels between NPAC machines as well as between such high performance distributed cluster and the user front-ends. In the next stage, we plan to address the WAN gigabit networking issues within the planned K-12 education project.

Parallel Oracle relational database management system will be installed on the nCUBE2 system at NPAC within the ongoing research project in collaboration with the TECHBASE program at JPL, sponsored by the US Army agency ASAS. The system will contain: 32 nodes with 32/Mb node, 16 disk controllers with 4Mb cache each, 32 1Gb disks, 16 nChannel VME interfaces for the SGI Crimson front-end and the back-up device (Controller + robotic exabyte). The resulting system will offer high capacity high performance database capability with the estimated throughput of 1000 transactions per second. The Oracle software will include the distributed RDBMS model, the Oracle tools such as SQL*ReportWriter, SQL*Forms etc. as well as support for C and Fortran interfaces to parallel SQL. The SGI Crimson front-end, connected to the nCUBE2 over the VME bridge, can be used for the visualization purposes in the stand-alone model and it also supports both FDDI and HIPPI interfaces which allow to hook up the whole system to a fast network.

3. Current and Planned Software Thrusts at NPAC

3.1. High Performance Fortran Compiler

We develop at NPAC the compiler code for the Fortan90D. This language model is compatible with SIMD designs such as CM-Fortran or MP-Fortran and is also portable across a range of multiprocessors and distributed environments. We intend to position Fortran90D to be compatible with High Performance Fortran but to include experimental extensions which provide additional functionality which may be included in future enhancements to HPFortran. HPFortran is a critically important development for the field as it will allow industry to move into parallel computing with some assurance that their parallel software investment is protected. However, we do not yet have enough experience to define a 'complete' HPFortran and expect that the language will need to be extended. In particular the initial standard (expected Dec '92) will only support well regular synchronous problems. We have started a major project with Kennedy and Saltz aimed at incorporating the techniques of PARTI into FortranD. We expext the PARTI

extended Fortran90D to be particularly important for many NASA Ames applications. Saltz's group has already shown interesting results from PARTI on many NASA fluid flow calculations. As well as support for irregular computations, we expect FortranD to have several other features which will not be present in HPFortran. These include a heterogeneous network as the target 'machine'. We expect on the other hand that most commercial HPFortran compilers will target the particular vendors machines and not a heterogeneous network. As well as straightforward MIMD networks of workstations we consider both multicomputers and SIMD machines, running HPFortran, as components of the network.

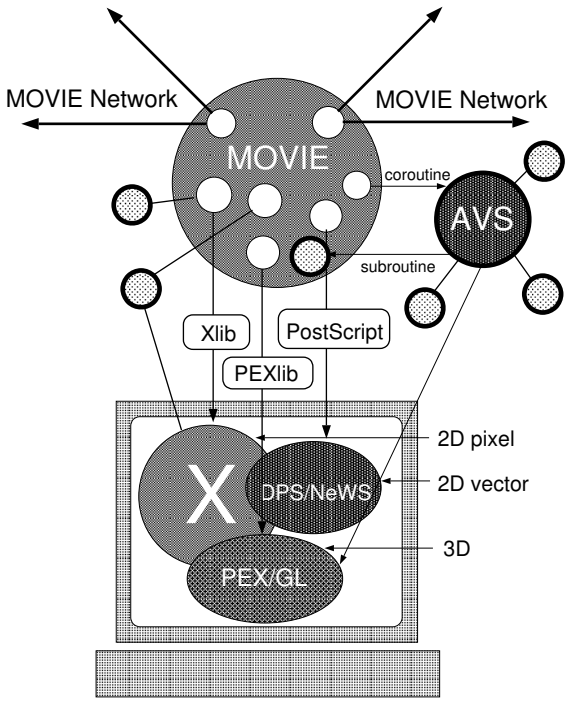

FIG. 1: Integrated graphics model in MovieScript. Uniform interface in terms of primitive types is constructed to X, DPS/NeWS and PEX/GL components of the Open Systems software and implemented, correspondingly, in terms of the Xlib, GL/PEXlib and PostScript communication protocols. Additionally, an interface to the AVS server is constructed, supporting both the subroutine and coroutine operation modes. The AVS-style extensibility model in terms of the UNIX dataflow processes is illustrated both for AVS and MOVIE servers. Within this model, one can also import other graphics models and applications to the MOVIE environment. This is illustrated on the example of the 'third party' X Window application which is configured as a MovieScript object or operator.

We also will be experimenting with new compile time optimization methods for collective communications and for automatic generation of alignment and distribution.

In a larger term project, we are currently designing a HPFortran interpreter [FoFu92a], discussed in Section 3.3

3.2. MOVIE System

MOVIE System is a network of MOVIE servers. MOVIE Server is an interpreter of MovieScript. MovieScript is a high level object-oriented programming language derived from PostScript. PostScript is embedded in the larger language model of MovieScript. This includes new types and operators as well as syntax extension towards the C++-style object-oriented model with dynamic binding and multiple inheritance. MOVIE Server is based on custom made high performance MovieScript interpreter. Some design concepts of MovieScript are inherited from the NeWS model, developed by Sun. C-shell based CASE tools are constructed for automated server language extension. MOVIE 1.0, to be released in fall '92, will offer uniform MovieScript interface to all major components of the Open Systems software such as X/Motif/OpenLook, DPS/NeWS, PEX/GL, UNIX socket library based networking and Fortran90–style index-free matrix algebra. Subsequent releases will build on top of these standards and extend the model by more advanced modules such as database management, expert systems and virtual reality. The language extensibility model is based on the concept of inheritance forest which allows to enlarge both the functional and object-oriented components of MovieScript, both in the system and application sector, both at the compiled and interpreted level. The default development model for MOVIE applications is based on MovieScript programming. System integration tools are also provided which allow to incorporate 'third party' software into the system and to structure it as suitable language extensions. Integrated visualization model is provided, unifying 2D pixel and vector graphics, 3D graphics and GUI toolkits. Interfaces to AVS-style dataflow based visualization servers are also provided. MOVIE Server is a single C program, single UNIX process and a single X client. The server dynamics is governed by preemptive multithreading with the real-time support. Threads, which are MovieScript light-weighted processes, compute by interpreting MovieScript and communicate by sending/receiving MovieScript. Uniform model for networking and message passing is provided. Various forms of concurrency can be naturally implemented in MOVIE such as single server multitasking or multiserver networks for MIMD-parallel or heterogeneous distributed computing. Multiserver systems of multhreading language interpreters offer novel approach to parallel processing, integrating data parallel and dynamic, irregular components. Due to the system features such as rapid prototyping, extensibility, modularity and 'in large' programming model, MOVIE lends itself for building large modern software applications.

3.3. Fortran90D/HPF Interpreter

We are currently designing an integrated language model, including concepts form FortranD and MOVIE and structured in the form of a Fortran90D interpreter. The central component of Fortran90 in the HPC domain is represented by the index-free matrix constructs. The performance penalty, typical to interpreted models, is negligible in this domain since all relevant operators refer to large memory blocks. In consequence, the concept of High Performance Fortran Interpreter is conceivable, and in fact very attractive due to the natural rapid prototyping capabilities of interpreted models.

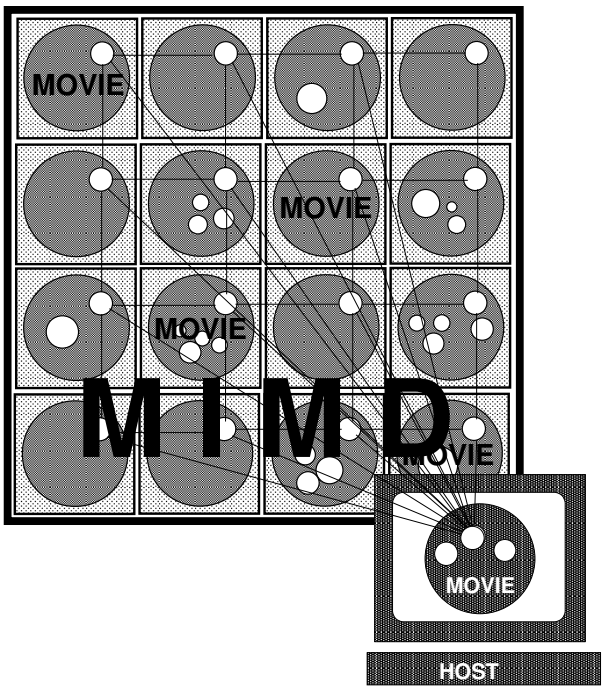

FIG. 2: Elements of the MIMD MOVIE model. Each node runs asynchronously an identical, independent copy of the multithreading MOVIE Server, interpreting (a priori distinct) node MovieScript programs and communicating with other modes via MovieScript messages. Regular and irregular components can be time-shared as illustrated in the figure. A single unique thread has been selected in each node (the one in the upper right corner) for regular processing and the other threads are participating in some independent or related irregular tasks. The regular thread processing is based on the 'MovieScript + message passing' model, i.e. all node programs are given by the same code which depends only parametrically on the node number. The mesh of regular threads is mapped on a single host thread which can be considered for example as a matrix algebra accelerator 'board' within some sequential or distributed Virtual Machine model, involving the host server.

MovieScript contains support for Fortran90–type or APL-type matrix constructs and operators and hence it lends itself as the implementation language for the Fortran90D interpreter. Additional functionalities of MovieScript in the area of graphics, networking etc. can be promoted to user layer of Fortran in the form of suitable runtime libraries. Object-oriented constructs can be expressed in terms of derived types of Fortran90.

Looking from another perspective, one can consider MovieScript as an intermediate representation language, a construct often used in a variety of compiling techniques to represent intermediate stages between subsequent compiler passes. Such intermediate languages were so far based on custom designs, driven by performance constraints and language specific internal bookkeeping requirements. More recent tendencies are towards open design and standardizing this layer to accelerate the compiler development process. PostScript plays a special singular role in this context as postfix syntax based and hence providing natural 'assembler' functionality for intermediate representations and internal language transforms, while at the same time it is also an industrial standard for electronic printing. It is conceivable that a successful design of MovieScript support for Fortran interpreter can be then naturally exploited for interpreting other HPC models such as for example dialects of concurrent C++.

3.4. Virtual Reality

The best choice of the human interface front-end depends typically on the problem at hand. In many scientific visualization problems, all one needs is a good quality graphics workstation, perhaps augmented by the 3D graphics accelerator boards. For more diversified interfaces, involving many information sources, the multimedia front-ends are most suitable. Finally, for advanced interactive systems with natural 3D embedding that require tight human-machine integration and user 'presence' in the modeled environment, Virtual Reality is the technology of choice. We intend to explore at NPAC a broad spectrum of such interface techniques, using initially the application set comprising the HPFortran benchmarking suite and then moving towards more advanced systems, developed in the individual research projects. In the area of conventional scientific visualization, we have recently evaluated the AVS system by developing visual 3D interfaces to various computational domains such as terrain imaging [Furm92] or high energy physics [Haupt92].

There is a strong interest in multimedia/VR at Syracuse University, recently enhanced by the small scope VR Lab, sponsored by Gentile from AGE, Inc. who is SU alumnus and partner in this successful NYC based start-up, developing the low-end VR peripherals on the entertainment market. We collaborate with AGE, Inc. within the NYS ACTION program and SU is expected to be the beta site of the new generation of low cost 'consumer VR' systems, now developed by AGE, Inc. and available in mid '93. We also explored more broadly the current status of the emerging VR industry [Furm91] and we are planning new initiatives including the use of VR and parallel processing

in K-12 education. We have to modify Grand Challenge applications such as a NASA fluid flow simulation to become an attractive environment to be used in precollege science education. We extend the HPFortran interpreter to be particularly useful in providing an environment where the student can experiment with and learn about the physics.

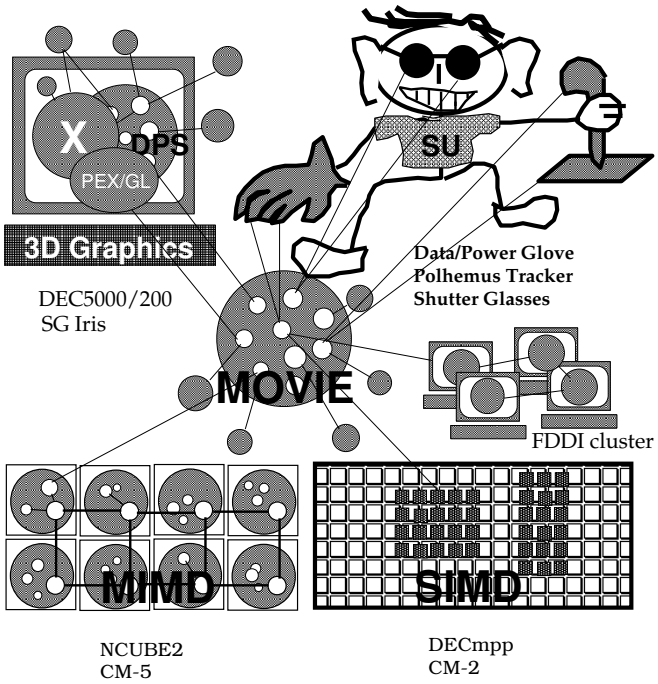

FIG. 3: Planned high-end Virtual reality environment at NPAC. New parallel systems: CM-5, NCUBE2 and DECmpp are connected by fast HIPPI network and integrated with distributed FDDI clusters, high-end graphics machines and VR peripherals by mapping all these components on individual threads of the VR MOVIE Server. Overall synchronization is achieved by the real-time support within the MOVIE scheduling model. Although the figure presents only one 'human in the loop', the model can also support in a natural way the multiuser shared Virtual Worlds with remote access capabilities and with variety of interaction patterns among the participants.

The planned VR equipment at NPAC will include BOOM, DataGloves and SGI, which is the front–end prototyped and currently used by NASA Ames for CFD visualization [BrLe91] [BrGY92]. We expect to benefit from NASA experience when building out VR Lab, in a similar way as NCSA and Army Research Labs did. We want also to evaluate high-end systems from Division, Inc. for the research purposes and low-end systems from Sense8, Inc. for educational purposes within our K-12 program.

4. Project Description

We propose here a collaborative effort between NPAC and NASA Ames towards an integrated high performance and high capacity environment for interactive visualization of unsteady flow CFD data sets. This class of complex visualization problems is characterized by immense data sets which do not fit the main memory on any existing computer. In consequence, the need arises for the high performance virtual memory technologies, coupled with the appropriate processing units to transform raw data fields to geometries and finally to images. Custom systems of this type such as *Tempus Fugit/Interview* [GeYa92] or *Distributed Virtual Windtunnel* [BrGY92] are currently developed at NASA Ames and based on custom distributed I/O library *dlib* [GeYa90].

One alternative solution, proposed in this document, is to use the emerging commercial parallel database management systems such as the nCUBE2 based parallel Oracle, discussed in Chapter 2. Automatic virtual memory management represents one of the critical components of each DBMS model and the associated software support was mastered by the relational database companies over the last two decades. One of the primary targets of the new family of parallel database systems are advanced multimedia servers so that one may expect these new systems to offer optimized performance not only for conventional small grain size transactions but also when handling large homogeneous data volumes. Furthermore, models such as parallel Oracle, based on a mesh of SQL interpreters, admit smooth integration of the data acquisition and the data processing modules. A simple integrated MIMD model of this type, when expressed in the hypercube terminology, would consist of even and odd subcubes, coupled in parallel through the 0th channel and dedicated respectively to the Oracle database sector and the user provided data processing sector.

MOVIE Server, discussed in Section 3.2, provides a natural building block to develop the processing sector, cooperating with the parallel database. In the even-odd model described above, the even/odd subcube is constructed as a mesh of SQL/MovieScript interpreters. The coupling between both components is realized by a suitable MovieScript interface to SQL. The most natural solution is constructed in terms of MovieScript *field* and *record* objects. The record design interpolates between the concepts of C structure and PostScript dictionary, i.e. it is accessed as a dictionary from the interpreted MovieScript layer and it has memory layout as determined by the corresponding C structure. Field is an n-dimensional array of numbers or records. Fields of numbers, together with the associated family of APL-style index-free matrix operators, form the base platform for the Fortran90D interpreter design. Field of records represents a natural model for a table or relation in a relational database. Due to this natural object-oriented mapping, the usual "impedance mismatch" , typical when embedding SQL in imperative languages, is absent here and moreover we also preserve high performance on the link between MovieScript and SQL servers since fields of records can be also manipulated by suitable vector operators, in-

cluding the base SQL command set. This design can be mirrored in a natural way in the MovieScript based Fortran90D interpreter, where MovieScript records translate to *derived types* of Fortran90 and, hence, a database table is represented by a Fortran90 array of derived types and the query language such as SQL can be embedded in the form of an appropriate intrinsic runtime library. Since, on the other hand, SQL provides an abstraction of virtual memory, or, equivalently, a disk storage model, one can also view such library as a Fortran90D model for parallel I/0.

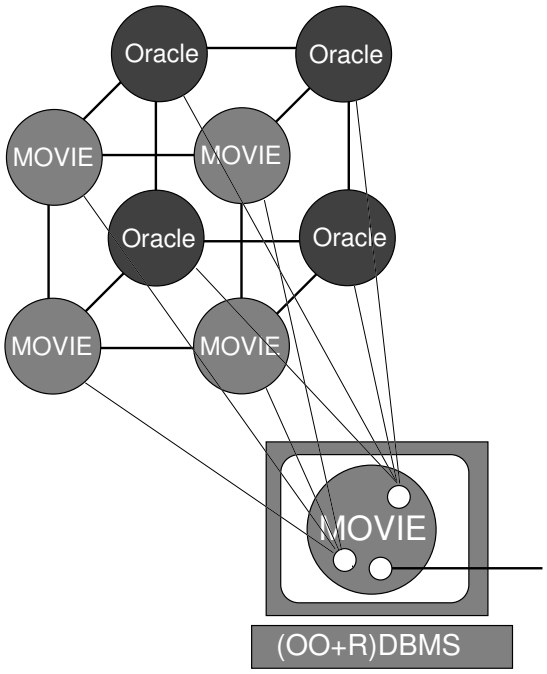

FIG. 4:: The even-odd hypercube model for integrating MIMD MOVIE model with parallel Oracle. Data acquisition layer represented by a mesh of SQL servers is coupled in parallel with the computational layer of MovieScript interpreters. Both regular and irregular computation patterns can coexist for the same data set, as discussed in Fig. 2. MovieScript can be used to build object-oriented constructs on top of the relational environment. From the network perspective, the MOVIE server running on the nCUBE2 host (an SGI workstation) can be viewed effectively as a single powerful object-oriented or/and relational database and visualization (see Fig. 1) server.

To summarize, the even-odd model above can be illustrated as a set of data parallel constructs, starting from the physical disk array, through an array of SQL interpreters and MovieScript interpreters, to end up at the programmer side in the form of an abstract portable data parallel I/O model of Fortran90D.

The construction, described above, represents a typical set of experiments with the Fortran90D I/0 design, planned within the interpreter project. The data retrieval and processing tasks associated with unsteady flow CFD visualization will provide an ideal testbest for such experiments. The detailed algorithms for data transformations from raw to geometry to image will be inspected and the appropriate MovieScript operators will be constructed for a selected visualization tasks. Regular loosely synchronous components of these algorithms will be transformed directly into appropriate Fortran90D constructs. Irregular components will be either converted to appropriate Fortran90D library routines or retained at the MovieScript level.

The MIMD model discussed above can be also realized in the distributed computing mode after replacing the parallel Oracle component by a set of distributed database servers. In this form, it becames closer to the Tempus Fugit/Interview system at NASA Ames. 'Supercomputer' is represented here by a distributed mesh of MovieScript servers and the 'dlib' corresponds to a mesh of SQL servers or is to be retained as is, i.e. as a custom database model.

Analogies exist also between the dlib [GeTa90] and SuperGlue [HuRa92] models at NASA Ames and the MOVIE model at NPAC. The context preserving remote connection in dlib corresponds in MOVIE to the programmable multiserver model. Each server can build remotely a suitable context by sending appropriate MovieScript message and then refer to it in subsequent connections. However, MOVIE communication is lower level at the C implementation layer (since based directly on the socket library) and higher level at the programmer layer (since based on an appropriate set of interpreted object-oriented message passing primitives).

SuperGlue shares many common aspects with MovieScript. Both models offer interpreted object-oriented interfaces and C-level extensibility. The difference is that MOVIE develops its own multithreading model, based on preemptive real-time scheduling and integrated message passing and networking services, whereas SuperGlue uses the external thread library and is a stand-alone server rather than distributed multiserver model. In a sense, one can therefore say that MOVIE integrates the concepts of dlib and SuperGlue.

In turn, integrating functionalities of MOVIE and SuperGlue seems to be an attractive project as both systems address rapid prototyping from different perspectives: SuperGlue offers a set of advanced graphics tools whereas MOVIE provides more generic portable support based on Open Systems standards. While following the FortranD + MOVIE based computational paradigm in the proposed project, we would like at the same time to explore the design and functionality of graphics objects in SuperGlue or in the PLOT3D package, used in Ames. Depending on source availability of these systems, we would adopt appropriate integration techniques so that the computing environment, prototyped in the course of the proposed effort would be usable and acceptable at Ames at the front-end level.

Yet another domain where we expect to benefit from the NASA Ames expertise and leadership is Virtual Reality. Following the example of NCSA and Army Research Center, we intend to start our VR Lab by cloning the NASA Ames VR environment for the Wind Tunnel. The planned VR equipment at NPAC, currently at the level of pending proposals [FoFu92b][BCFNV92], include color BOOM, DataGloves and SGI, which matches the peripheral set used at NASA Ames. We would also like to port the base VR software support such as drivers, geometry and control libraries etc.

As the first step in this direction, our graduate student, Diglio Simoni, spent this summer in NASA Ames, working in Bryson's group on software tools for the Windtunnel. He also developed an interesting dynamic algorithm based on content-addressable memory for real–time recognition of the DataGlove gestures. We would like to continue this work at NPAC within the proposed project of CFD visualization and to compare the CAM algorithm with other pattern recognition techniques such as the geometry invariants classifiers or the method of moments.

In the next stage, we plan to evaluate MOVIE model as a candidate for the Virtual Reality operating shell. We would like to build appropriate MovieScript interfaces to VR peripherals and to test the server dynamics in the context of VR challenges such as dynamic multi-user access, collision support, frame rate synchronization in the distributed environment etc. To our knowledge, there is yet no systematic attempt at constructing a 'VR language', i.e. a set of objects and operators, suitable for portable, high-level VR programming. MOVIE system, with its rapid prototyping capabilities, language extensibility tools and the dynamical threading model, suitable for VR, seems to be a natural candidate to address such design problems. We also intend to compare MOVIE is this domain with other VR operating shells in order to extract the already accumulated experience in the existing and functioning designs. This include systems such as VEOS at HITLab, Trix at Autodesk and also the SuperGlue at NASA Ames which shares common features both with MOVIE and with VEOS.

5. Statement of Work

We quantify here the proposed project in terms of a set of action items and deliverables. The scope of potential collaboration between NASA Ames and NPAC within the general outline in Chapter 4 can in fact turn out broader than the 1 year effort, suggested by this NRA. Below, we list the proposed tasks for the first year and within the proposed budget and we also indicate possible directions of continued or extended collaboration.

The four major components of the proposed effort and the associated tasks are:

- **evaluation of parallel Oracle for the unsteady flow CFD visualization**
  The unsteady flow data sets will be transferred to NPAC and ported to the parallel Oracle environment. A set of benchmarks will be performed to measure the retrieval performance of this system as compared with the distributed collection of SQL servers (we will use INGRES, available on all DECstations) and with the custom database or I/O library such as dlib at NASA Ames. The full comparison with system such as Tempus Fugit will require MIMD-parallel implementation of the transformation filters, converting raw data to geometries and to images. A small selected set of such algorithms will be implemented as part of the 1 year project.
- **comparison and integration of NASA Ames and NPAC software systems**
  The detailed task specification depends on the availability of technical information and/or sources for the software systems used at NASA Ames. Ideally, we would like to install at NPAC systems such as dlib, PLOT3D, SuperGlue and Tempus Fugit/Interview and to compare them with NPAC systems. From the MOVIE perspective, of particular interest is SuperGlue which follows a similar design. In the first year, we would evaluate the concept of integrating both systems and we would design appropriate software tools such as Schema <-> MovieScript translators.
- **CFD visualization as a benchmark application to drive the HPFortran I/O design**
  A sample CFD visualization code, developed at NPAC, would be included into the HPFortran benchamrk suite to drive the design of the parallel I/O language construct. An attractive option, suggested in Chapter 4 and to be evaluated as part of this project is to embed the retrieval functionality of relational database in the HPFortran I/O design. In consequence, the visualization code developed on top of parallel Oracle could be then easily ported to a custom I/O library.
- **evaluate MOVIE model for VR Operating Shell**
  The detailed task specification depends on availability of VR equipment at NPAC within the first year project. We intend to clone NASA Ames VR environment, integrate it with MOVIE and develop VR operating shell capability in MovieScript. We would like to formulate it as a join project with NASA Ames in which NASA Ames experience in VR would provide the driving force in shaping up the appropriate language structures of MovieScript. We expect that

during the first year project we will be able only to evaluate and articulate this concept and perhaps port the low level VR software support, whereas the MovieScript VR support would require an extended, larger scope project.

6. References

[BrLe91] Bryson, S. and Levit, C., "The Virtual Wind Tunnel: An Environment for the Exploration of Three-Dimensional Unsteady Flows", RNR Technical Report RNR-92-013, NASA Ames Research Center, Moffett Field, CA, 1991.

[BrGY92] Bryson, S. and Gerald–Yamaski, M., "The Distributed Virtual Windtunnel", RNR Technical Report RNR-92-0130, NASA Ames Research Center, Moffett Field, CA, 1992.

[BCFNV92] Berra, B.P., Chen, R.C., Furmanski, W., Nilan, M.S. and Venkatesh, M, "User and System Models for the Design and Evaluation of Systems for the Sensing and Interpretation of Complex Virtual Environments", Syracuse University Proposal to the Air Force Office of Scientific Research, July 1992.

[FoFu92a] Fox, G.C. and Furmanski, W., "Towards Interpreted Run-Time Fortran90D Environment", Syracuse Center for Computational Science, Internal Report, 1992.

[FoFu92b] Fox, G.C. and Furmanski, W., "CISE Research Instrumentation for Integration of Virtual Reality into High Performance Computing Environment", Syracuse University Proposal to the National Science Foundation, Aug 1992.

[Furm91] Furmanski, W., "Report from the Virtual Reality Tour", Syracuse Center for Computational Science, Technical Report, SCCS, 1991.

[Furm92] Furmanski, W., "MOVIE – Multitasking Object-oriented Visual Interactive Environment", in Fox, G.C., Messina, P. and Williams, R., editors, *Parallel Computing Works*, Morgan and Kaufman, 1992 (in preparation).

[FFHNPS92] Faigle, C., Furmanski, W., Haupt, T., Niemiec, J., Podgorny, M. and Simoni, D., "MOVIE Model for Open Systems based High Performance Distributed Computing", in *Proceedings of the First International Symposium on High Performance Distributed Computing, HPDC-1, Sept 9–10, 1992, Syracuse, NY*, IEEE Computer Society Press.

[FHKKKTW90] Fox, G.C., Hiranadani, S., Kennedy, K., Koelbel, Ch., Kremer, U., Chau-Wen Tseng and Min-You Wu, "Fortran D Language Specification", SCCS/42C, 1990.

[FJLOSW88] Fox, G.C., Johnson, M.A., Lyzenga, G.A., Otto, S.W., Salmon, J.K. and Walker, D.W., *Solving Problems on Concurrent Processors*, Prentice-Hall, 1988.

[Haupt92] Haupt, T., "Visualization of High energy Physics Data using AVS", Syracuse Center for Computational Science Technical Report SCCS-243 (1992).

[HuRa92] Hultquist, J.P.M and Raible, E.L., "SuperGlue: A Programming Environment for Scientific Visualization", RNR Technical Report RNR-90-014, NASA Ames Research Center, Moffett Field, CA, 1992..

[GeYa90] Gerald–Yamaski, M., "Distributed Library", RNR Technical Report RNR-90-008, NASA Ames Research Center, Moffett Field, CA, 1990.

[GeYa92] Gerald–Yamaski, M., "Interactive and Cooperative Visualization of Unsteady Fluid Flow", RNR Technical Report RNR-92-018, NASA Ames Research Center, Moffett Field, CA, 1992.

7. Budget

Salaries	**% Effort**	**Year 1 Total**
Researcher	0.50	16,000
2 Graduate Students	1.00	28,000
Fringe Benefits @ 33%		12,804
Travel		5,000
Material & Supplies		0,442
Tuition 24 credit hours @ $406/hour		9,744
Subtotal		71,990
IDC @ 45%		28,011
Total		100,001

Budget Explanation

- **Salaries:** The salaries are based upon standard rates for the categories listed for a period of twelve months.
- **Travel:** This is a projected figure based upon two round trips to NASA Ames from Syracuse for the Researcher and one student.

- **Material & Supplies:** This category will cover expenses for day to day operation for research team described above. The cost of publishing papers and required reports is also included.