Distributed Interactive Simulation for Synthetic Forces *

P. Messina, S. Brunett, D. Davis, T. Gottschalk
Center for Advanced Computing Research
California Institute of Technology
Pasadena, California 91125

D. Curkendall, L. Ekroot, and H. Siegel
The ALPHA Group
Jet Propulsion Laboratory
Pasadena, California 91109

Abstract

Interactive simulation of battles is a valuable tool for training. The behavior and movement of hundreds or thousands of entities (tanks, trucks, airplanes, missiles, etc.) is currently simulated using dozens or more workstations on geographically distributed LANs connected by WANs. The simulated entities can move, fire weapons, receive "radio" messages, etc. The terrain that they traverse may change dynamically, for example due to rains turning dirt roads into mud or bombs forming craters. Thus the entities need to receive frequent information about the state of the terrain and the location and state of other entities. Typically, information is updated several times a second. As the number of simulated entities grows, the number of messages that need to be sent per unit of time can grow to unmanageable numbers. One approach to reducing the number of messages is to keep track of what entities need to know about which other entities and only send information to the entities that need to know. For example, tanks in Germany need not know about a change of course of a ship in the Pacific. This technique for reducing messages is known as interest management.

Caltech and its Jet Propulsion Laboratory have implemented a simulation of this type on several large-scale parallel computers, exploiting both the compute power and the fast messaging fabric of such systems. The application is implemented using a heterogeneous approach. Some nodes are used to simulate entities, some to manage a database of terrain information, some to provide interest management functions, and some to route messages to the entities that do need to receive the information. Some of these tasks require more memory than others, some require faster processing capability. Thus the application is heterogeneous both in its functional decomposition and to a smaller extent in the characteristics of the hardware that is used to run each function. In addition, workstations are used to run the Graphical User Interface (GUI) that is used to control the simulation and to visualize the simulation as it is running. This approach has been used to run an exercise with over twice the previous record number of vehicles simulated.

A near-term goal is to simulate 50,000 entities. To do so, it will be necessary to run the simulation on several geographically distributed SPPs. For pragmatic reasons (availability of sufficiently large systems), the machines employed will have different architectures.

1 Introduction

Simulation of synthetic environments and activities for training of military personnel is routinely carried out on distributed, homogeneous computing assets. Caltech has undertaken a project whose goal is to increase substantially the size and fidelity of these simulations. Our approach of using large-scale parallel computers has led to a heterogeneous computing strategy. This paper describes our software architecture, our motivation for using a heterogeneous approach, and preliminary experience with the implementation of the simulation program on parallel systems.

2 Background

The United States Department of Defense has found it increasingly useful to train individuals and commands using simulated environments. These simulations have become more realistic and effective with the advent of computer-generated scenarios, visualizations, and battlefield entity behaviors. Of particular importance has been the development and use of Distributed Interactive Simulation (DIS). A large implementation of the DIS was conducted by several units located in Europe in November of 1994. It was called Synthetic Theater of War—Europe (STOW-E). It combined the classic manned simulator entities (as originally developed under SIMNET) with Modular SemiAutomated Forces (ModSAF) simulation soft-

---

*Support for this research was supplied by the Information Technology Office, DARPA, with contract and technical monitoring via Naval Research and Development Laboratory (NRaD)

0-8186-7879-8/97 $10.00 © 1997 IEEE

ware executing on networks of workstations; the individual ethernet networks were themselves interconnected by Wide Area Network (WAN) links. The total number of simulators and ModSAF entities used in this exercise was about 2,000. Stimulated in part by this successful exercise, current simulation initiatives have vehicle count goals in the 10,000–50,000 range. A vehicle is defined in the military argot to be any substantial entity—ground, air vehicles, autonomous personnel, etc. In addition to a desire to simulate more entities, the trainers and the trainees are constantly asking for more resolution, faster refresh rates, higher fidelity, more automatic behaviors, increased training environment responsiveness, and overall improvements in the training environment. Finally, there is the emergent realization that faster than real-time analytic simulations will be required in the future to support the operational use of simulations in the battlefield itself. This latter capability is essential if the simulation software is to be used for planning as opposed to training. It should be noted that this class of simulation has applications in other fields, for example for emergency response to natural disasters.

These demands for increased capability and capacity lead one naturally to consider devising a software architecture and computer platform strategy that will support a wide range of requirements. In other words, a scalable approach is needed.

Caltech and its Jet Propulsion Laboratory (JPL) have a long history of using parallel computer architectures for scalability of scientific and engineering simulations, including discrete event simulations. In addition, in the CASA gigabit testbed project [1], we performed experiments with distributed, heterogeneous implementations of several applications executing on parallel supercomputers connected by a high-speed wide-area network. Hence when we decided to tackle the challenge of supporting more ambitious simulations, we quickly decided to apply the large-scale capabilities of High Performance Computing and Communications (HPCC) assets as an alternative to WAN-linked sub networks of workstations in order to develop and demonstrate the software architectures needed to reach these goals.

The Caltech/JPL project, called Synthetic Forces Express (SF Express), selected ModSAF as the base software to enhance and use to carry out scaling experiments. The SF Express project has a two-year goal to achieve a 50,000 vehicle count simulation via:

The efficient operation of the ModSAF software on individual, large, SPP platforms and, the networking of two or more of these large platforms together as a single metacomputer for the largest runs. These WAN's will include connectivity to more conventional ModSAF assets of workstations and simulators.

At present, the SF Express Team has pilot versions of its emerging software architecture operational on Intel Paragon platforms at Caltech and Oak Ridge National Laboratory (ORNL) and on IBM SP2 systems at Caltech and Ames Research Center (ARC). Use of the much larger SP2 at the Cornell Theory Center's SP2 is about to begin. Efforts are also underway to port the SF Express software to the CRAY T3D and T3E class of machines.

At this writing a full 10,000 vehicle scenario, approximately twice the size achieved previously, has been demonstrated on several occasions using the 1,024-node ORNL machine. Indeed, one of these demonstrations took place live during Supercomputing '96 from the floor of the Pittsburgh Convention Center. Software adapted to the SP2 has achieved runs of up to 8,000 vehicles on the 143-node SP2 at ARC.

To date, these simulations have been run using scenarios created by NRAD and executed using the simulated ground environment of that of the Fort Knox Terrain Database. Larger scenarios—up to 50,000 vehicles—are actively being constructed, this time on the much larger playing field afforded by Southwest USA Terrain Database (SWUSA), centered near 29 Palms and spanning much of the surrounding territory of Southern California.

Based on measured performance of our variant of the ModSAF code, we have determined that no single available SPP can execute the full 50,000 vehicle scenario; indeed, the near term 50,000 goal was selected in part so as to require the involvement of two or more supercomputers. Accordingly, our SPP architecture includes provisions for networking several large SPPs together, creating a meta-supercomputing network.

In what follows, we discuss some of the key architectural concepts being explored to make ModSAF suitable for SPP machines and to improve its overall scalability. While ModSAF is the basis for all of our current work, we intend that the applicability of this research to be much broader. ModSAF, then, is the current focus serving both as a convenient tool and as a familiar yardstick for measuring progress familiar to a large community.

3 Interest Management

We take as axiomatic that to enable dramatic scalability of entity level simulations, "interest management" must be central to the software architecture.

Using the language of ModSAF, beyond a certain (rather small) limit, it is necessary to abandon broadcast style inter-entity messaging schemes and insert rather precise interest management techniques. This arises because of two separate but related notions:

An entity's behavior is shaped partly by an awareness of other entities around it (local perceived ground truth). Since not all entities of interest are computed by the same local CPU, the need arises for "remote entities" to signal their presence and activities to that local CPU via messaging. But if each individual CPU attempts to deal with all of these incoming messages (global ground truth), all CPU's will be overwhelmed both in memory and in performing bookkeeping duties. Interest management must be performed more globally to permit scalability.

As the number of entities increases, an all to all protocol eventually overwhelms the physical SPP messaging fabric. The same conclusion is obtained: a global interest management scheme is critical.

Accordingly, the SF Express Team has been experimenting with two variants of global interest management: one a server based notion and a second router based scheme.

Space does not permit their detailed exposition here [2] but the main ideas are easily grasped. See Figure 1.

In Figure 1, the top squares represent nodes executing the ModSAF entity behavior codes known as SAFSIMs. As part of this behavior, each vehicle asserts its interest in what in effect are "regions of interest spaces." There are several of these—e.g., a high and a low resolution terrain space, vehicle i.d., signal frequency—but to grasp the basic ideas it suffices to consider interest to be a function of geographic location. In the server interest management scheme, this interest is registered in one of the interest management nodes, nodes which themselves are decomposed over the index of that interest space. Messages (known as PDUs in ModSAF) generated by any vehicle are sent (registered) to the coordinate of that interest space corresponding to the coordinate of the sending vehicle. For example, if a PDU is sent from a vehicle whose location is (x, y), it is sent to the (x, y) coordinate of the Interest Management (IM) nodes. The IM node then forwards the message back to each SAFSIM that has registered an interest in that coordinate.

Looking at the process from the point of view of the IM nodes, each maintains queues of messages to be sent to each SAFSIM, looping over all SAFSIMS, and sending a single bundled message for each traversal of that loop. In this relatively straightforward manner, messages arrive at only the SAFSIMs that have explicitly asserted interest. The remote entities represented at each SAFSIM node and the volume of individual PDUs processed are thus kept to a minimum.

In this IM scheme, communications channels are associated with interest classes, and a single simulator node will generally exchange data with more than one IM node. In the alternative Router model, each simulator node has a single communications channel to the "outside world."

The basic building block of the Router architecture is a fixed collection of SAFSIM nodes associated with a Primary Router, as seen in the bottom of Figure 2. The SAFSIM nodes send data and interest declarations up to their associated Primary Routers, and only the appropriate, interest selected data flow back down. Data communications among the (SAFSIMs+Primary) building blocks are accomplished through additional layers of data collection and data distribution router nodes shown in the top part of Figure 2. Communications within the upper layers occur in parallel with those in the Primary↔SAFSIM layer. This means that there are no significant additional time costs for data messages which take the longer (5 hop) path through the full communications network.

The use of (few) fixed communications channels in the Router architecture allows extremely efficient bundling of data messages. During the communications-intensive initialization phases of ModSAF, individual messages flowing down to the SAFSIM nodes routinely contain 40 or more PDUs, and total data rates through the Primary Routers in excess of 16K PDU/second have been observed. Once initializations are completed, the "steady-state" Primary↔SAFSIM communications account for only about 3% of a SAFSIM's (wall clock) time.

A system-wide evolving picture of interest declarations and payloads can be obtained from the Router architecture. Tracing performance and program behavior, along with general purpose logging capabilities, are facilitated by the very nature of the Router clusters.

4 Functional Decomposition

Vanilla ModSAF normally executes completely within a single workstation, replicating workstations until enough are employed to execute the desired size of the simulation. There are two basic modules in ModSAF: the SAFSIM, already identified, and the GUI which is only activated on a workstation if it is desired to input to the scenario or observe the simulation's progress. In building SF Express, we have already migrated some of the sub elements away from

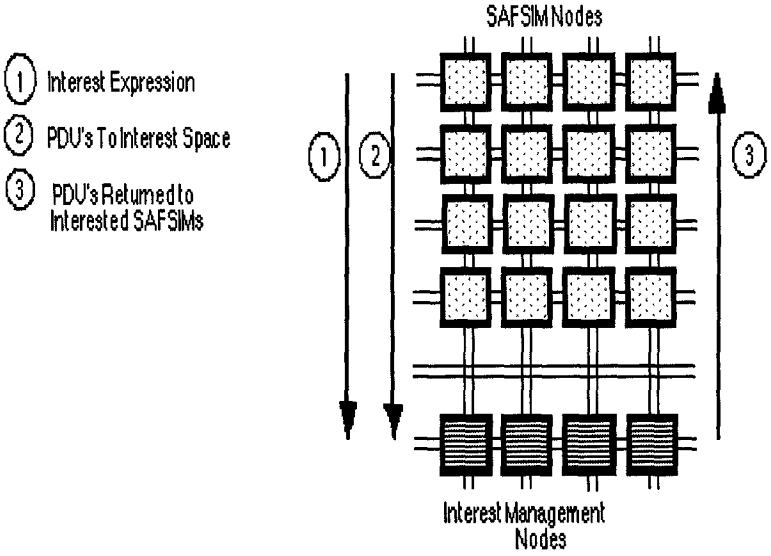

Figure 1: Interest Management Server

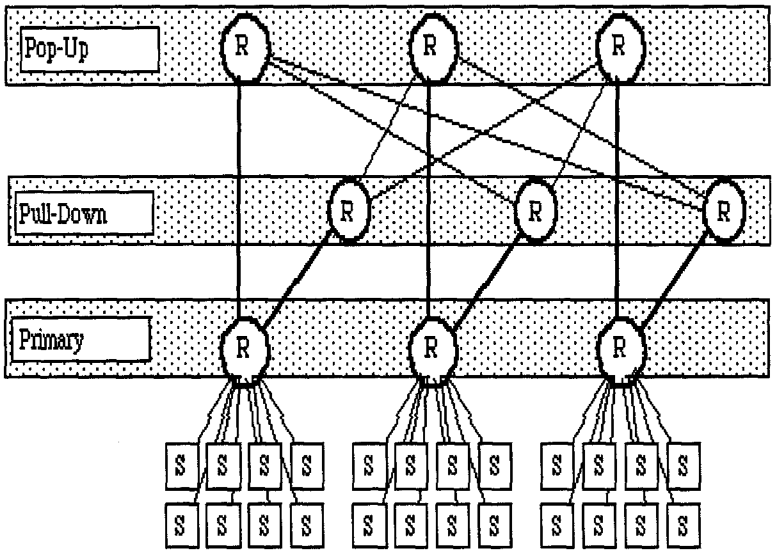

Figure 2: Router-Based Interest Management

the SAFSIM and are planning to migrate others. In addition, we add others, such as the interest management just discussed, as separate and new functions not present in vanilla ModSAF.

Functional decomposition is natural in the quest for scalability. When a resource (such as a terrain data base) must be accessed simultaneously by hundreds or thousands of processors, one replicates it. If the data base is large, computers with large memories should be used. If the computational cost of simulating a complex type of vehicle is high, one spins off that task to separate nodes; if to achieve fidelity the simulation requires a lot of floating-point computation, nodes with suitable CPUs should be chosen. Router nodes on the other hand will do few if any floating-point computations to carry out their role; routers can therefore be hosted on systems that excel at logic and integer operations. Data logging for subsequent replay of the simulation might require processors with ample attached disk storage.

The need to keep up with real time also dictates a functional decomposition. Furthermore, in some simulations sensor data from real instruments must be read and processed to guide parts of the simulation. Visualization of the ongoing simulation is essential and it also requires a different type of computer resource.

An indication of how this approach is used in practice can be gleaned from our experience with a 10,174 Vehicle synthetic forces simulation that was run by on the Oak Ridge National Lab 1024-node Intel Paragon. The run, approximately twice as large as the largest previous such simulation, utilized a scenario set on the Ft. Knox Compact Terrain Data Base, with "blue and red" forces made up of battle tanks, fighting vehicles, armored personnel carriers and trucks. This run employed 784 Paragon processors, of which 640 were devoted to simulating vehicles, 48 processors acted as routers in a communications network that provides the good scalability demonstrated; 90 processors were used as terrain data base servers; and six processors were used as servers to load the program and data. In addition, a GUI proxy node was used, as is described in the next section.

In short, heterogeneous functional decomposition is a natural strategy for coping with the evolving needs of synthetic forces simulations.

4.1 Graphics user interface and visualization

We have experimented with a number of approaches to providing GUI functionality. The most straightforward method on the SP2 is simply to take advantage of its X Windowing system and devote one or more nodes hosting a complete ModSAF with an X Window output being sent to a remote workstation. This is an attractive option, particularly when it is desired to interact with the simulation during its progress: e.g., vehicles can be created and instantiated on the GUI node as in workstation based ModSAF, a function not otherwise readily available with the SPPs. It is also easy to "interest manage" the display, by attaching the GUI node directly to the Interest Management nodes. Interest is geographically expressed by turning the screen display corner coordinates into an interest expression. PDUs only from vehicles within the covered region will be transmitted to the GUI node, a key circumscription if that node is not to be overwhelmed with irrelevant information.

A second technique removes the GUI from the SPP entirely, substituting there instead a GUI Proxy, and executing a workstation GUI as a stand-alone unit on the outside. This workstation then transmits interest declarations to the Proxy, which in turn interfaces with the interest management machinery in a manner similar to a SAFSIM. This technique is less demanding of connection bandwidth but sacrifices some of the portability of the X Windows approach.

A third approach, and one which ultimately may prove more powerful, is to send the PDUs themselves out of the SPP to external devices. These data can be compressed and limited in various ways, but current experience indicates that the entire PDU stream can be issued by the SPP and assimilated by a high performance workstation in real time. A current experiment [3] describes progress in processing the PDU stream on external devices either for more scalable real time display or for after action analysis. The post processing can subdivide the PDU stream, redirecting the PDUs to multiple processes and to, for example, a matrix of coordinated screens, giving an overall view of the battlefield.

4.2 Replacing routine disk access

Frequent retrieval of data from disk storage is too slow to be practical on the SPPs. Instead, reader files common to all SAFSIMs are held in RAM in one or more file server nodes. Supplying each SAFSIM with its required information then takes place at RAM access and SPP messaging rates, greatly reducing initialization time. We are currently experimenting with compiling these reader files into binary prior to any single simulation. This compacts the files and further speeds up their delivery to the individual SAFSIM nodes.

The simulation terrain in ModSAF is represented through a fairly elaborate, memory-efficient scheme built from small terrain elements ("pages" and "patches"). Arbitrarily large terrains are supported through a caching scheme in which a SAFSIM maintains only a modest fraction of the full terrain in memory, requesting new pages and patches as they are needed.

In the parallel implementation, the disk-read data retrievals of conventional ModSAF are replaced by message exchanges with database server partitions. Each partition consists of a sufficient number of nodes to hold the entire terrain database in memory. Multiple replicas of the database partition are used for runs with large numbers of SAFSIM nodes.

4.3 Some future possibilities

While not currently implemented, the above terrain serving scheme is consistent with ultimately providing for dynamic terrain. Since only a few terrain servers are needed, it is practical to keep these synchronously updated with terrain changes and, via cache coherence methods, ensure that the SAFSIMs receive cached updates as well.

In the future we expect to migrate more functionality away from the individual SAFSIMs. Terrain reasoning is a good candidate. High level and complex functions such as path planning are currently handled within the SAFSIMs on a lower priority basis than the fundamental activity loops. The computation takes many cycles to complete and its performance is hard to predict. Migrating that function to the terrain server nodes has great appeal.

It may even be helpful to migrate lower level functions like intervisibility calculations there as well. In workstation based ModSAF many intervisibility calculations are unnecessarily duplicated. Vehicle A calculates its visibility to remote vehicle B, while in B's local workstation, the reciprocal calculation is being made to its remote vehicle A. Doing this calculation once in a server can gain important economies.

Finally, decomposing the ModSAF functionalities and switching to a server perspective paves the way for higher fidelity reasoning and environmental calculations, since more CPU power can be deployed to any one function when it is needed without interfering with the tightly controlled and repetitive tasks within each SAFSIM.

5 SPP Portability

SF Express has been built around MPI messaging libraries, a necessary but by no means sufficient condition to ensure portability. Machines that have been addressed so far with various degrees of completeness are:

Intel Paragon
IBM SP2
Cray T3D
SGI Origin 2000
SGI Power Onyx/Challenge Series
Beowulf

The codes on the Paragon and SP2 are by far the most mature. The major difficulty encountered with the Paragon was the reversed endianess as compared to all other machines on the list, save Beowulf. The port to the SP2 was smooth and uneventful. Unfortunately the Cray T3D has proved the most difficult of all, almost entirely because of the lack of a 32 bit Cray C compiler. ModSAF was definitely not written with portability to 64 bit machines in mind. Our current approach is to work with the AC compiler authored by Bill Carlson and available on both the T3D and the T3E. Success here would give the Project access to this important class of machines.

An informal port to the SGI Origin 2000 was performed and demonstrated during the Supercomputing '96 Convention in Pittsburgh. The Power Onyx/Challenge Series of machines are listed, even though they are shared memory machines, because they offer an MPI library. The shared memory machines, then, emulate the message passing architectures and the SF Express concepts port without difficulty. Since ModSAF itself is native to the SGI's, the port was uneventful.

A Beowulf "pile of PCs" cluster, has been built by the California Institute of Technology and the Jet Propulsion Laboratory. The cluster consists of 16 Intel Pentium Pro (200MHz) processors running Parallel Linux connected via a 100Mb/sec ethernet switch. Out of the box ModSAF has been ported to Beowulf. We are experimenting various MPI extensions and profiling libraries to maximize efficiency and properly characterize the performance of the SF Express port.This kind of cluster shows very good price-performance ratios and may be a viable platform for future uses of SF Express.

In summary, we are pleased with the considerable—but incomplete—progress made towards our portability goals. We believe that offering options to be an important aspect of enabling the continuing applicability of this research.

6 Interoperability and Meta-supercomputing

Implementing SF Express on multiple machines is additionally important to achieving the project goal of 50,000 entities. As mentioned in the introduction, no single SPP is likely to be able to achieve this goal and it will be necessary to utilize two or more SPPs together connected by wide area networks to achieve this result.

Fortunately, the essential information that needs to be shared among the participating SPPs is exchanged using ModSAF PDUs and their data structures were designed to interoperate with different machines. Endianess and machine word lengths will not pose difficult problems.

Also, the key to scalability is once again, precise interest management. And this can be accomplished between SPPs as an extension of the interest schemes already described.

In an unconstrained world, a uniform messaging structure would be established across the whole meta-supercomputer and the structures we have been discussing would need no modifications at all—a node on a distant machine would be different only in that it had a unique node identification. Unfortunately, this would require the WAN network to be as high in bandwidth and message handling capabilities as the SPP messaging fabrics themselves. Since we will attempt the metacomputing runs with at best OC-3 networks, an approach more parsimonious of bandwidth resources is required.

Referring to Figure 2, one can think of the interface between the geographically-distributed SPPs as being done by connecting the Pop-Up routers with WAN connections. The time delays for PDUs sent through the upper router layer are modest (e.g., less than 50 msec) and thus likely to be small compared to the delays introduced by WAN access.

This approach has not been fully implemented but its broad outlines are clear. To establish a global interest manager, each SPP would need to create periodically (once every ~1–5 sec) a complete interest expression across the entire range of interest coordinates. The remote SPP returns only the PDUs responsive to those interests.

7 Conclusions and Plans for '97

At this writing, the project is consolidating the progress made thus far which culminated in the 10,000 and 8,000 vehicle runs at ORNL and ARC respectively. Implementations are being cleaned up and more comprehensive attention paid to instrumentation and measurement.

Near term developments include the design of the meta-supercomputing interfaces to enable the employment of two or more SPPs in a single exercise.

In addition, little attention has been paid thus far to how to make the large simulations thus enabled available to conventional ModSAF cluster workstation networks and simulators. In the sense that everyone speaks DIS protocol, the interface is easy and assured. But once again, interest management must be enabled as a two way interface between the parties, else the workstations will be overwhelmed and the influence of the entities modeled within the conventional workstations will not be properly represented to the SF Express Forces within the SPP. There are several choices available; perhaps the best is to treat the SF Express as an HLA federate and implement a standard HLA/RTI interface to the outside world.

We are being asked to reach the 50,000 goal this year and in pursuit of this are setting up the necessary cooperations between several major national SPP assets. In addition to the assets at JPL/CIT, we are enlisting support in pursuit of the meta-supercomputing goals from ORNL, ARC, CTC, and the San Diego Supercomputing Center (SDSC).

Acknowledgments

Access to various computational facilities was essential to the work performed. The Intel Paragon and SP2 at Caltech were made available by the Caltech Center for Advanced Computing Research and the Oak Ridge Intel Paragon by the Oak Ridge Center for Computational Sciences. The Cray T3D at JPL and the JPL Dual SGI Power Onyx Visualization Laboratory were made available by the JPL/Caltech Supercomputing Project. Finally, access to the IBM SP2 at ARC was provided by the Numerical Aerodynamic Simulation Systems Division at NASA Ames Research Center.

References

[1] Messina, P., and Mihaly-Pauna, T., "CASA Gigabit Network Testbed: Final Report." California Institute of Technology Technical Report CACR-123, July 1996.

[2] Craymer, L. and Lawson, C. "A Scalable, RTI Compatible Interest Manager for Parallel Processors." 1997 Spring Simulation Interoperability Workshop.

[3] Plesea, L. and Ekroot, L., "Data Logging and Visualization of Large Scale Simulations (SF Express)." 1997 Spring Simulation Interoperability Workshop.

Sharon Brunett is a computing analyst for Caltech's Center for Advanced Computing Research. She is currently involved with optimization and characterization of applications on shared and distributed memory

MPPs. She is also involved with modifying application programs to exploit new systems software and language features. Current interests include analyzing the various I/O requirements and interfaces necessary for applications communicating across multiple MPPs. She received her B.S. in Computer Science and Applied Mathematics from the University of California Riverside in 1983.

David W. Curkendall received a BS in electrical engineering from Cornell University and a PhD in engineering and science from University of California, Los Angeles. He is manager of the Advanced Parallel Processing program at the Jet Propulsion Laboratory. His early technical specialization were in interplanetary navigation and the development of radiometric and optical tracking system technology. For the past several years, he has been active in all areas of computational science, including the leadership of the technology teams that developed the early Hypercube Concurrent Computers, their software and applications. He is currently active in several applications of MPP machines including parallel rendering, military simulation, and leads a NASA multidisciplinary Grand Challenge Science team in spaceborne Synthetic Aperture Radar processing.

Dan Davis received a B.A. in Quantitative Psychology from the University of Colorado in Boulder in 1973. He entered graduate study there in Business Administration, but transferred to the University of Colorado School of Law. He received his Juris Doctors degree in December of 1975 and was admitted to the bar of the Supreme Court of Colorado and the U.S. Supreme Court, among others. He is currently the Assistant Director of The Center for Advanced Computing Research at Caltech. He is currently involved in implementing large scale simulations on Scalable Parallel Processing computers. These simulations range from military battlefields to VLSI designs. His computer research activities have run from his early work in computational analyses for the behavioral sciences at C.U., Boulder through defense work in the U.S. Navy, in which he holds a reserve commission as a Commander, Cryptologic Specialty.

Laura L. Ekroot received the B.S. degree in electrical engineering from the California Institute of Technology, Pasadena, in 1986, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1988 and 1991 respectively. She is currently working as a Member of the Technical Staff at the Jet Propulsion Laboratory in Pasadena, CA. Her research interests include Shannon theory, coding theory, image compression, and parallel simulation.

Tom Gottschalk is a Lecturer in Physics and a Member of the Professional Staff at the California Institute of Technology. In 1978 he received a Ph.D. in Theoretical Physics from the University of Wisconsin. In 1974, he received a BS. in Astrophysics from Michigan State University. He is currently researching new architectures and algorithms for parallel processing of various simulations. His work is directed at applying the power of large computers, capable of computing in the range of tens of GigaFLOPS. In the electronics area, he has developed a new tool for analyzing VLSI design verification on very large circuits, VLSIs of 5 million transistors or more. In the defense area, he has worked on battlefield simulators and missile defense tracking algorithms. His early work was in simulating sub-atomic particles using Monte Carlo models. He is a frequent speaker on the subject of computer and internet usage at primary and secondary schools.

Paul Messina is Assistant Vice President for Scientific Computing at Caltech, Faculty Associate in Scientific Computing, Director of Caltech's Center for Advanced Computing Research, and serves on the executive committee of the Center for Research on Parallel Computation. His recent interests focus on advanced computer architectures, especially their application to large-scale computations in science and engineering. He also is interested in high-speed networks and computer performance evaluation. He heads the Scalable I/O Initiative, a multi-institution, multi-agency project aimed at making I/O scalable for high-performance computing environments. Messina has a joint appointment at the Jet Propulsion Laboratory as Manager of High-Performance Computing. Messina received his PhD in mathematics in 1972 and his MS in applied mathematics in 1967, both from the University of Cincinnati, and his BA in mathematics in 1965 from the College of Wooster.

Herb Siegel received a B.A. in mathematics from the University of California at Berkeley in 1962 and an MA in mathematics from the California State University at Los Angeles in 1965. His early work at the Jet Propulsion Laboratory included the calibration of interplanetary radiometric range and development of high performance software correlators for VLBI. He led the development of a commercial multiprocessor for Action Computer Enterprise. Currently he is the computational leader for a NASA Grand Challenge effort in high performance computing and Earth-Space science to processes interferometric SAR over high performance networks and using national assets of MPP machines.