

Synthetic Forces Express: A New Initiative in Scalable Computing for Military Simulation

Paul Messina, Sharon Brunett, Dan Davis, and Tom Gottschalk

Center for Advanced Computing Research
California Institute of Technology
Pasadena, California 91125

**Dave Curkendall, Loring Craymer, Laura Ekroot, Charles, Lawson, Craig Miller,
Lucian Plesea, and Herb Siegel**

The ALPHA Group Jet Propulsion Laboratory
Pasadena, California 91109

Kevin Boner, David Fusco and Wallace Owen,

Naval Command, Control and Ocean Surveillance Center
Research, Development, Test, and Evaluation Division
San Diego, California 92152

Peter Brooks

Institute for Defense Analyses
1801 North Beauregard St.
Alexandria, VA 22311

ABSTRACT

The ability to configure simulation exercises employing very large numbers of objects has long been an objective of DARPA's simulation research program. This paper describes a new initiative, the SF Express Project, exploring the use of Scalable Parallel Processors (SPPs) in order to address this objective. The Project is chartered to create and demonstrate a software architecture implemented on SPPs that scales to simulations of 50,000 vehicles or more. ModSAF has been chosen as the underlying software for these developments; entity or vehicle counts achieved with SF Express can be interpreted in terms of the familiar ModSAF standard.

The paper addresses what new architectural elements are needed when adopting ModSAF to SPPs and scaling up to these very large simulations. Particular attention is paid to:

Interest management - systematically restricting the exposure of any part of the simulation to just those elements of interest

Functional decomposition of ModSAF for better load balancing across the SPP and more scalability in the final product.

Portability to different SPP platforms and SPP interoperability in order to enable the configuration of metacomputer platforms for the largest simulations.

Instrumentation to enable after action analysis both of how the simulation and exercise performed. Scalable, distributed

logging and visualization are considered.

Results of current simulations operating at 10,000 vehicles and beyond are presented and discussed.

1.0 INTRODUCTION

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 area of Distributed Interactive Simulation (DIS) using IEEE Standard 1278.x. 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 software 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. Stirred in part by this successful exercise, current simulation initiatives have vehicle count goals (a vehicle is defined in the military argot to be any substantial entity - ground, air vehicles, autonomous personnel, etc.) in the 10,000-50,000 range. In addition to this, the trainers and the trainees are constantly asking for more resolution, faster refresh rates, higher fidelity, more automatic behaviors, increased training environment responsiveness and over all 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.

Accordingly, Caltech/JPL, under the direction of NRD are pursuing an aggressive program to apply the large scale capabilities of High Performance Computing & 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 Synthetic Forces Express (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 the present time, the SF Express Team has pilot versions of our emerging software architecture up and running on the Intel Paragon platform at Caltech and Oak Ridge National Laboratory (ORNL) and on the IBM SP2 platform at Caltech and Ames Research Center (ARC). At this writing, it has been granted access to the Cornell Theory Center's SP2. 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 1,000 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 cpu SP2 at ARC.

To date, these simulations have been run using scenarios created by NRD 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.

We anticipate that no single existing SPP can execute the full 50,000 vehicle scenario and, 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.

2.0 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 [see the related paper, reference 2] but the main ideas are easily grasped. See Figure 1.

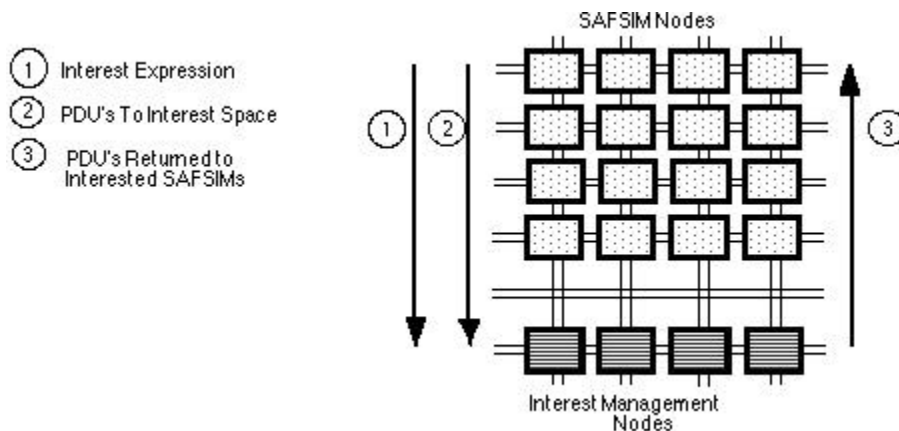


Figure 1. Interest Management Server

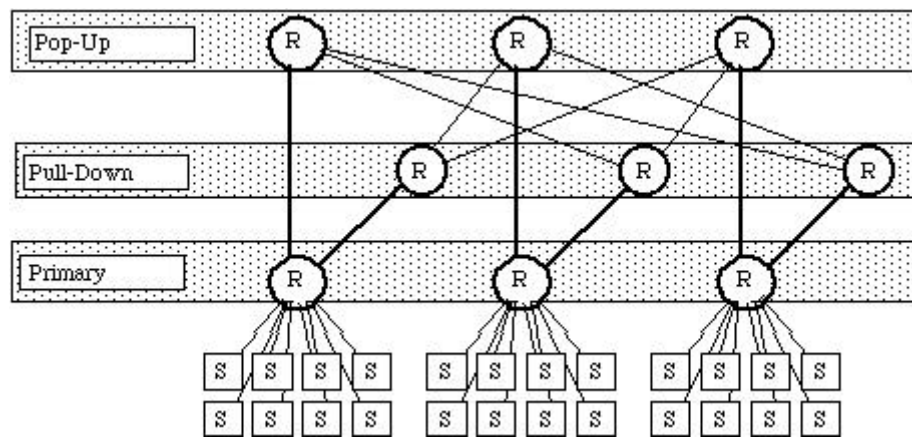


Figure 2. Router-Based Interest Management

In Figure 1, the top squares represent nodes executing the familiar ModSAF entity behavior codes (SAFsims). As part of this behavior, each vehicle asserts his 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 a mental picture of interest as a function of geography is sufficient for the basic ideas. In the server interest management scheme, this interest is registered in one of the interest management nodes, nodes themselves which are decomposed over the index of that interest space. Messages (PDU's) 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 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 PDU's 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 Fig. 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 Fig. 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.

3.0 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 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.

3.1 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 SPP's. 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. PDU's 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 standalone 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 PDU's 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 [reported in detail in an accompanying paper,reference 1] describes progress in processing the PDU stream on external devices either for more scalable real time display or for after action analysis. The postprocessing can subdivide the pdu stream, redirecting the PDU's to multiple processes and to, for example, a matrix of coordinated screens, giving an overall view of the battlefield.

3.2 Replacing Routine Disk Access

Routine appeal to disk storage is not practical on the SPP's. For example, 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.

3.3 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.

4.0 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

By far, the codes on the Paragon and SP2 are the most mature. The major difficulty encountered with the Paragon was the reversed endianness 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.

5.0 Interoperability and Meta-supercomputing

Developing SF Express for 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 network two or more SPP's together over wide area networks to achieve this result.

We are fortunate that the essential information that needs to be shared among the participating SPPs are the familiar ModSAF PDU's and the data structures for these have been designed to interoperate with different machines. Endianness and machine word lengths will not prove difficult problems.

Also, the key to scalability is once again, precise interest management. And this can be accomplished between SPP's 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 meta runs with at best OC3 networks, an approach more parsimonious of bandwidth resources is required.

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

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

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. We are grateful to each organization individually for their cooperation and support, and collectively for set of national assets that makes this program possible.

We especially wish to thank the Peer Review Group which has overseen and guided this work intellectually. They have been composed of: Thomas Sterling, Joshua Smith, James Calvin, and Larry Schuette

References:

- 1) Plesea, Lucian and Ekroot, Laura, "Data logging and visualization of large scale simulations (SF Express)" 1997 Spring Simulation Interoperability Workshop.
- 2) Craymer, L. and Lawson, C. " A Scalable, RTI Compatible Interest Manager for Parallel Processors" 1997 Spring Simulation Interoperability Workshop.