Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

IMPLEMENTING EMERGING HIGH PERFORMANCE COMPUTING AND DATA MANAGEMENT TECHNOLOGIES IN AGENT BASED SIMULATIONS

Thomas D. Gottschalk

Center for Advanced Computing Research
California Institute of Technology
1200 California Blvd.
Pasadena, California 91125, USA

Ke-Thia Yao, Gene Wagenbreth, & Dan Davis

Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, California 90292, USA

ABSTRACT

Increasing needs for larger and more sophisticated agent based simulations of urban areas prompted the U.S. Joint Forces Command to seek out and apply technologies largely developed for academic research in the physical sciences. The use of these techniques in experimentation is closely tied to the behavioral sciences and has been shown to be effective and stable. The authors set out their decade and a half experiences in implementing high performance computing hardware, software and user interface architectures to enable heretofore unachievable results at JFCOM. They focus on three advances: the use of general purpose graphics processing units as computing accelerators, the efficiencies derived from implementing interest managed routers in distributed systems, and the benefits of effective data management given the surfeit of information produced. Their goal is to inform their simulation colleagues as to the potential improvements and the pitfalls of utilizing these technologies to meet their own challenges.

1 INTRODUCTION

There are new technologies that are emerging that may be of interest to the simulation community. The simulation group at the Information Sciences Institute at USC and the research staff at the Center for Advanced Computing Research at Caltech have been implementing three technologies that fall into this group:

- The use of interest managed routers for a trans-continental test of 10 Gigabit WANs
- The implementation of an optimized distributed data management scheme
- The experimental and production use of a new GPU accelerator-enhanced Linux Cluster

The impetus for design and implementation of these was the need for large-scale, battlefield simulations. The current locus of this activity is the U. S. Joint Forces Command, (JFCOM). The team reports requirements, design considerations, configuration decisions, and early experimental results. This should aid the simulation community in assessing the applicability of these technologies to other simulations.

1.1 Joint Forces Command Mission and Requirements

JFCOM's mission is to lead the transformation of the Armed Forces into the 21st Century via their Joint Concept Development and Experimentation Directorate, J9. This mandate calls for experiments with warfighters staffing the consoles during interactive simulations. They use well-validated entity-level urban combat simulations, *e.g*. Joint Semi-Automated Forces (JSAF). The J9 codes consist of representations of terrain that are populated with intelligent-agent friendly forces, enemy forces and civilian groups. JFCOM required simulations of more than 2,000,000 entities on a global-scale terrain database (Ceranowicz,

2005). The line-of-sight calculations between the entities are an "n-squared" problem (Brunett, 1998). This mandated the use of an innovative interest-managed communication's architecture (Barrett, 2004).

1.2 JFCOM's JESPP

A scalable simulation code capable of 1M entities, known as the Joint Experimentation on Scalable Parallel Processors (JESPP) project (Lucas, 2003), grew out of an earlier project named SF Express. (Messina, 1997) The JFCOM experimenters had been constrained in a number of dimensions, *e.g*. numbers, sophistication, realism, *etc*. The early JESPP experiments showed that the new code could scale beyond the 1,000,000 entities (Wagenbreth, 2005), but required more computing, *e.g*. a GPU-enhanced cluster.

2 INTEREST MANAGED ROUTERS IN WIDELY DISTRIBUTED SYSTEMS

2.1 Introduction

With the geographic distribution of the computers and the human-in-the-loop participants, as shown in the map below (Figure 1), the authors had to reduce long-haul communications as much as possible

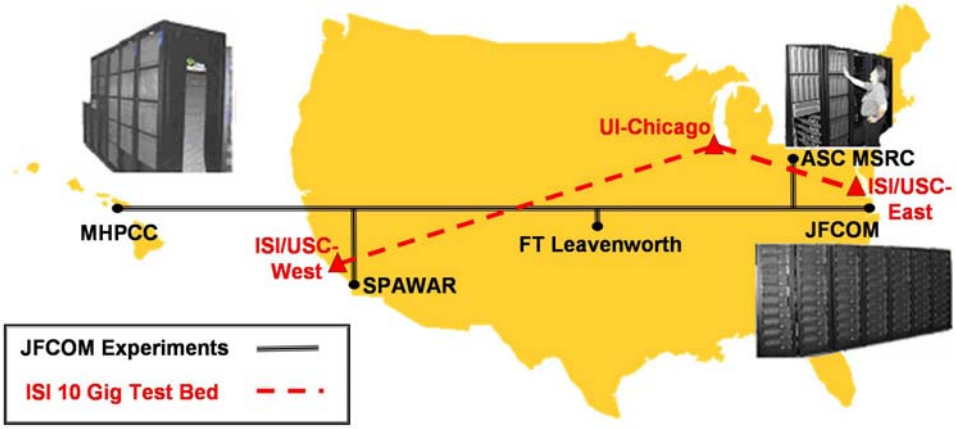

Figure 1
JFCOM Experimentation System and ISI 10 Gig/Sec. Test Bed

2.2 Approach

This section reports the results of bandwidth tests of interest managed message exchange among processors on three clusters. As the intent was to not interrupt ongoing activities at JFCOM, a separate, but comparable, Wide Area Network (WAN) was established. The nodes of this WAN were located at ISI-East in Virginia, ISI-West in California, and the University of Illinois at Chicago. Previous work (Gottschalk, *et al*., 2005) had indicated the utility of interest managed communications on cluster meshes, high-bandwidth Local Area Networks (LANs) and lower bandwidth WANs. The current work was specifically investigating high-bandwidth (10 GigaBit per second) WANs with transcontinental distributions.

Interest-limited message exchange was done using ISI's MeshRouter formalism (Barrett, *et al.* 2004). The main conclusions of the benchmarking studies are as follows:

- Throughput on a single link (client to router, router to router, *etc*.) is limited to about 320 Mbits/sec. This result does not depend on the location of the involved processors (*e.g*., same or different clusters) but instead appears to reflect limitations of the RTI-s (Ceranowicz, *et al.* 2002) communications primitives used in this study.

- By using multiple routing connections among the participating sites, aggregate bandwidths of 4.8 Gbits/sec were achieved.

The total bandwidth for the aggregate tests represents almost 50% of the nominal WAN bandwidth for the networks used in the tests. While good, it is slightly smaller than rates achieved using simple, network performance tests (*i.e.*, "iperf").

2.3 General Test Configurations

The bandwidth experiments were done using the standard ISI MeshRouter formalism for interest-managed communications. A schematic of the MeshRouter is shown in Figure 2.

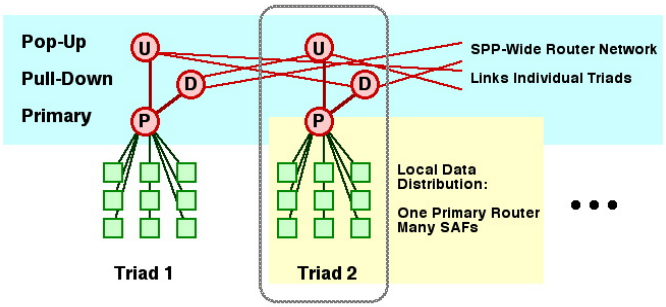

Figure 2. Schematic MeshRouter Topology

The overall communications scheme consists of collections of processors (labeled "SAFs" in Figure 2) each communicating with a specified "Primary" router (P). Interest-limited message exchange among the various basic "Triads" is done using a network of additional "Pop-Up" and "Pull-Down" routers. The three routers on a triad are instanced as separate objects within a single MeshRouter process. Some details of message management along the various links of are needed in order to assess the results of the benchmark study. As described in Barrett, *et al*. (2004) the MeshRouter software is object-oriented (C++) with daughter classes used to implement key, application specific details, including interest state enumeration, basic message interpretation (*i.e.*, "headers"), and "bits on a wire" communications primitives.

The results reported here use an HLA RTI-s (High Level Architecture, Run Time Infrastructure) implementation for both interest enumeration and the lowest-level communications primitives ("dataflow nodes"). While this has enormous advantages, it does have incompletely understood, overheads. Standard RTI-s dataflow implementations exist for both TCP and UDP communications. The results presented here use the TCP implementation.

The application processes for the benchmark tests are of two forms:

- Publish Processors: Send out messages of specified length and interest state. The nominal total publication rate (Mbyte/sec) is controlled by a data file that is re-read periodically (by all publish processors). The nominal experimental data rate can be controlled dynamically.

- Subscribe Processors: Receive messages for a specified interest state, collecting messages from multiple publishers, as appropriate. The subscribe processes are instrumented to measure actual incoming message rates and to detect missed messages.

The routers direct individual messages from publishers to subscribers according to the interest declarations. The router processes were instrumented to determine the fraction of time spent on management.

2.4 Elementary Tests

The first configurations explored involved a publisher and router at one site (ISI-East) and a subscriber and router at a second site (UIC), as shown in Figure 5. The subscripts indicate the associated interest state.

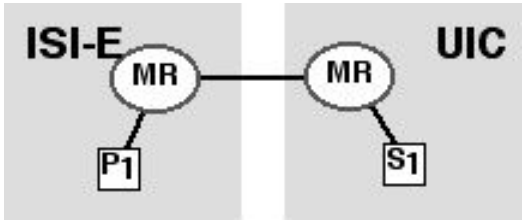

Figure 3
Simple test schematic

This basic configuration was used to explore dependences on various parameters of the basic MeshRouter setup of Figures 2 & 3, in particular: packet size within the standard RTI-s data flows and individual publish message sizes.

Table 1: Maximum rates versus size parameters

Packet Size (bytes)	10 KByte Messages	100 KByte Messages	400 KByte Messages
8192	-	40 Mbyte/sec	-
65536	35 MByte/sec	40 MByte/sec	30 MByte/sec

The first column in Table 1 specifies a buffer size within the RTI-s software. Throughput did not have strong dependence on this parameter. Attempts with a larger packet size (262144 bytes) resulted in software failures within the RTI-s libraries on the ISI-E router. The dependence on individual message size reflects known behavior within the full RTI-s package. Smaller messages mean latency on start-up has higher overhead. Very large message sizes incur an overhead from fragmenting and reassembling

The tests shown in Table 1 were also done for smaller message sizes (1 KByte down to 40 bytes) with throughputs falling precipitously for individual message sizes much below 1 KByte.

The "optimal" maximum inter-site rate for a single communications link was 320 Mbits/second/link, which was found to be remarkably consistent for all benchmarking tests. Somewhat better rates were observed for simple tests will all processes/processors on a single cluster. Schematics for four such tests are shown in Figure 4.

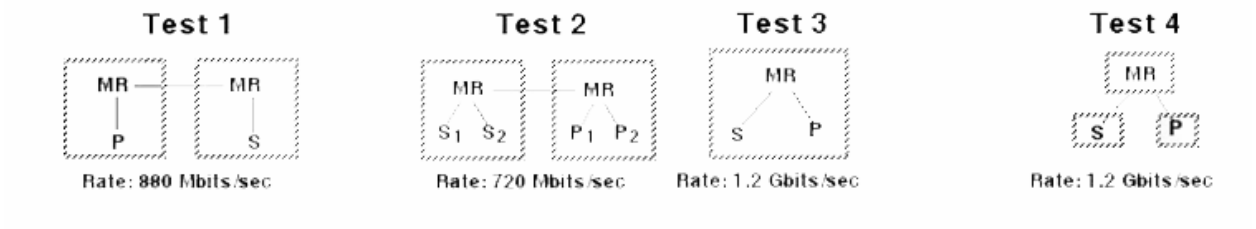

Figure 4
Test configurations on a single machine.

The rectangles represent individual compute notes within the clusters, with the symbols representing individual processes within the nodes. The numbers quoted below each diagram are the maximum achieved total message throughput. The intra-processor rates are higher than the inter-processor rates by factors of 2 to 4, the "three-hop" rates (tests 1,2) are about 2/3 of that for the "two hop" tests, and performance is comparable for the inter- and intra-processor configurations. It is helpful to compare these results with a more direct comparisons of TCP communications rates using iperf, with observed a rate of 1.9 Gbits/sec, for Node A to Node B and a rate of4.0 Gbits/sec, for Node A to Node A

These comparisons demonstrate that the full MeshRouter implementation does currently involve considerable overhead, particularly concerning the dataflow communications primitives within the RTI-s library. Significant improvements on the results described in this note can be achieved, but at the cost of implementing a leaner set of communications primitives. It should be noted that the MeshRouter itself does not appear to be the primary limiting factor for performance. In the simple tests of Figure 5, the measured fraction of time that the router is busy with its management services is O(40%). The primary performance limitations come from the lower level RTI-s communications.

In preparation for the wide area tests described in the next section, a number of generalizations of the "Test 4" configurations were done using the ISI-W cluster. These tests involved configurations with a single router process on its own node, a single publish process on its own node and multiple subscriber processes on either one or two nodes Performance results for these configurations are summarized in Table 2, with the bandwidth numbers reflecting only data rates into the subscriber processes.

Table 2: Bandwidth results for various test configurations within the ISI-W cluster.

Number of Subscribers	Subscriber Nodes	Per Node Bandwidth	Total Bandwidth	Router Load
1	1	680 Mbit/sec	690 Mbit/sec	41% Busy
2	1	672 Mbit/sec	1.3 Gbit/sec	54% Busy
4	1	504 Mbit/sec	2.0 Gbit/sec	59% Busy
6	2	344 Mbit/sec	2.1 Gbit/sec	55% Busy
8	2	288 Mbit/sec	2.3 Gbit/sec	51% Busy
16	2	160 Mbit/sec	2.6 Gbit/sec	44% Busy

The bandwidth into individual subscribers falls somewhat steadily as the number of subscribers increases. The total bandwidth out of the router into the collection of subscribers is seen to rise although the increases for the later rows of Table 2 are fall smaller than those seen in the first two rows. This behavior is better illustrated in the plot of aggregate bandwidth versus number of subscribers shown in Figure 5.

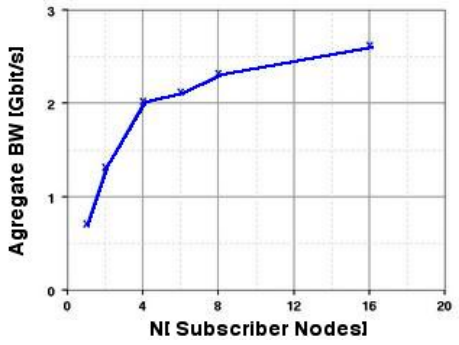

Figure 5
Aggregate bandwidth versus number of subscribers

Strong rises for small numbers of subscribers suggests a simulation has not yet reached the inherent capabilities of the general MeshRouter framework while the significant flattening for larger problems is likely related to competition for outgoing communications capabilities from single router processes.

2.5 Performance Tests Over The Wide Area Network

The results from the previous sections suggest two general observations: Point to point rates for RTI-s communications are significantly smaller than those within a single cluster and aggregate bandwidth can be increased by exploiting multiple communications paths into and out of individual router processes. These observations suggest that tests across the WAN should involve a rather rich, multiple-router mesh configuration. The key elements in this configuration are as follows

- All publish processes are located at one site (ISI-E)
- Each ISI-E router node services a separate publish process for each interest state in the exercise (five different states for the configuration shown in the diagram)
- The router process and its associated publishers are instanced as separate processes within a single node of the ISI-E cluster
- Multiple nodes within the ISI-E cluster run separate copies of the basic publish "building block" of the previous three points.
- Subscriber configurations on the UIC cluster are "mirror images" of the ISE-E publishers.
- The MeshRouter communications are "fully connected" with explicit point-to-point links between a pair of routers.

A number of variants of the basic configuration were explored, such as the number of distinct interest states (*i.e.*, number of processors associated with a single router processes) the number of replicas of the basic "Router plus Associated Pub/Sub" nodes ate each site.

Typical performance numbers for a test with eight participating nodes at each of the UIC and ISI-E site are summarized in Table 3.

Message Length	Client BW (bytes/sec)	Single MR BW (bytes/sec)	Aggregate BW (bites/sec)
0.4 KByte	3.2 M	16.0 M	1.0 G
0.8 KByte	6.4 M	32.0 M	2.1 G
1.6 KByte	12.8 M	64.0 M	4.1 G
2 KByte	14.3 M	71.5 M	4.6 G
100 KByte	0.8 M	4.0 M	0.3 G

Table 3: Performance measures for a typical WAN test.

Aggregate throughput is rather poorer for individual message sizes that are either too small or too large. This is consistent with the simpler, single SPP results of the previous section and are largely due to the nature of the RTI-s communications primitives (which have been carefully tuned to perform best for 2-4 KByte message buffers).As long as one stays near the optimal message size, the aggregate bandwidth for the WAN test is about 4.6 Gbits/sec. WAN tests with variations in configuration gives similar results:

Max Total WAN BW = 4.6 – 4.9 Gbit/sec

This is almost half of the nominal 10 Gbit/sec bandwidth of the ISI-E to UIC network and compares well with a rate of roughly 7 Gbit/sec achieved using multiple parallel executions of iperf (without any of the overhead of RTI-s communications).

3 DISTRIBUTED DATA MANAGEMENT FOR LARGE SCALE SIMULATIONS

3.1 Introduction

These simulations generate terabytes of data that must be effectively managed to be useful to the analyst. The High Level Architecture Object Model Template (HLA OMT) supports simulation interoperability by providing a Federation Object Model (FOM) to formally describe the information interchange (objects, object attributes, interactions, and interaction parameters) within a federation among the federates. Information used by a single federate is defined by the Simulation Object Model (SOM). Often the individual SOMs are mutually incompatible, so standing up a federation typically requires a tedious process modifying the simulation federates to conform to the purposed FOM. A variety of agile FOM techniques have been proposed to facilitate this integration process. From the simulation data logging and analysis perspective, there is an analogous problem of adapting the analysis tools to particular federations. Data analysis tools are designed in accordance with the analysts' notion of Measures of Effectiveness (MOE) and Measures of Performance (MOP). Often these measures are invariant with respect to the underlying federation object model. This is especially true for the lower-level MOP. This section presents a two-layered framework that supports the agile adaptation of analysis tools to specific federations. The top semantic layer provides a modeling framework to capture concepts that analysts tend to use. The concepts include measurements and dimensions, such as object classification, time, and geographic containment. The lower syntactic layer describes how to map the particular federation object models to more abstract semantic concepts. In addition, we show how this approach supports reuse by taking advantage of the hierarchical nature of the object models. These concepts have been successfully implemented by JFCOM.

3.2 Data Management Framework

The type of Measure Of Effectiveness/Measure Of Performance questions of interest to analysts are typically not directly captured by simulation loggers. In general analysts are interested in how well higher level mission tasks and objects are satisfied. A MOE is a question or measure, designed to show how well particular tasks are satisfied with respect to a system (Gentner *et. al.*, 1996). A MOP is typically quantitative measure of a system characteristic used to support a MOE. For example, sample MOE questions are can the red forces be pinned, or can the sensors detect red force movement within urban environment. MOP supporting these MOEs may include percentage of red forces killed/damaged, percentage of blue forces kill/damaged, time take to cross terrain, percentage of forces detected within sensor foot-print, percent of forces detected total, and percentage of detection by sensor type by terrain type by time of day.

Simulation loggers do extremely well at capturing detailed operational data, such entity state changes of individual entities and interactions among the entities. Depending on the type of entity, entity state changes may include location, orientation, and velocity. For vehicles, additional attributes may include external lights on and engine on. Interactions may include collision, damage assessment, sensor detection, and contact report. After appropriate processing these types of operational data are potentially useful to the analysts. The level of the raw logging data collected from the simulation is too low to be of direct use to the analysts. Information has to be abstracted from the logged data by collation, aggregation and summarization. In particular for the JSAF simulation federate for the Urban Resolve exercises as it was used by JFCOM.

3.3 Defining the Analysis Data Model

One of the key focus areas of Urban Resolve exercises is to study the effectiveness of future Intelligence, Surveillance and Reconnaissance (ISR) sensors in helping soldiers operate in complex urban environments. The Sensor/Target Scoreboard provides a visual way of quickly comparing the relative effectiveness of individual sensor platforms and sensor modes against different types of targets (Graebener *et. al*., 2003; Graebener *et. al*., 2004). Sensor/Target Scoreboard is a specific instance of the more general multidimensional analysis (Kimball *et. al.*, 1998). We use the Sensor/Target Scoreboard to motivate the discussion. In a 2005 I/ITSEC paper, we described the data management and analysis tool Scalable Data Grid that uses multidimensional analysis (Yao and Wagenbreth, 2005).

Simulated sensor entities lay down sensor footprints to delimit coverage. For targets, a contact report holds the result. The contact report includes information about the sensor, sensor platform, sensor mode, target, detection status, target type, perceived location and velocity, *etc*. Sensor/Target scoreboards have the capability of providing summary views by aggregating individual sensor platforms into sensor platform types, such as high altitude, medium altitude, and low altitude. It aggregates individual target entity objects into target classes, which can range from the generic (Large Truck) to the specific (Russian MAZ-543).The Analysis Data Model (ADM) is defined in terms of multidimensional analysis. Two key concepts in the ADM are Dimensions of Interest and Measures Of Performance as shown in Figure 6.

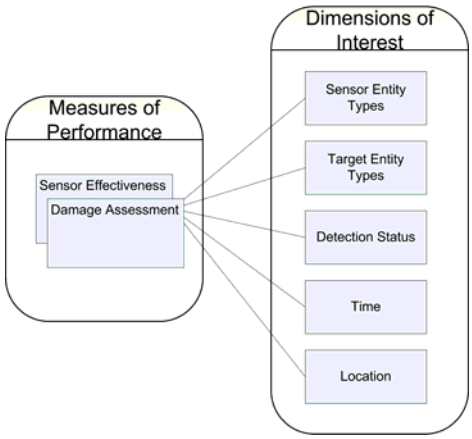

Figure 6
Dimensions of Interest and Measures of Performance

For large simulations, the magnitude of data collected ranges in the terabytes. Dimensions categorize and partition the data along lines of interest to the analysts. Defining multiple crosscutting dimensions aids in breaking the data into smaller orthogonal subsets associated measurement units. Choosing the granularity of these units aids in determining the size of the subsets. Another dimension example is in terms of simulation entity groups. In this sensor/target scoreboard case the analysts want to define two dimensions: one for the sensors and the other for the targets. For the sensor dimension, the analyst may want group together sensors by the type of platform: high-flying UAVs (unmanned aerial vehicles), mid-altitude UAVs, OAVs (organic air vehicles) and UGSs (unattended ground sensors). The targets may be grouped together, for example, by transportation mode: air, ground and sea.

Hierarchical dimensional units are also possible. For example, the analysts may want to subdivide the sensor platform category into the sensor modes: MTI (moving target indicators), SAR (synthetic aperture radar), images, video, and acoustic. Multiple unit decompositions of the same dimension are allowed. The analysts may want to verify how each federate response to the sensor contacts, so they define the unit category to be the type of federates. After the data have been partitioned along lines of interest, the data

subsets may still be large. Measures provide quantitative ways of characterizing the data subsets and can be aggregated. The hierarchical crosscutting dimensions partition the data into a hierarchy of subsets in order to provide a meaningful summarization. To be computationally efficient the operator for aggregating measure must satisfy the associative property—the measure of a set must be computable from the measures its subsets.

In the case of the sensor/target scoreboard the MOP is an integer count of the number of times a sensor has detected a target. The aggregation operator is the addition operator. Sometimes mean and variance performance measures are of interest to analysts. For example, instead of integer detection counts, the sensor/target could be extended to maintain a floating point number indicating degree of uncertainty. Then, in this case it makes sense to measure the mean and the variance of uncertainty. Mean and variance operators do satisfy the associative property. For example, given only the means of two subsets, it is not possible to compute the mean of the union of the two subsets. However, these measures are directly computable from associative measure. Mean is computable from two associative measures: the count of number of detections (or uncertainty), and the sum of uncertainty. Variance requires an additional sum of squares of uncertainty. Let X be the uncertainty, and n be the count of number of detections:

$$Mean = \frac{\sum X}{n}; \quad Var = \frac{n\sum X^2 - \left(\sum X\right)^2)}{n^2}$$

Typically MOE decomposes into multiple performance measures. If it is possible to decompose these measures along the same dimensions, or sometimes called con-forming dimensions, then it is possible to compare these measures. The type of question analysts may ask is “How more likely are damages to enemy target entities if they have been detected by sensors?”. Let X be sensor effectiveness and Y be damage assessment. If define an additional measure that is the sum of X times Y, then we can determine covariance between damage and detection by using the “mean” operator:

$$Cov(X,Y) = Mean(XY) - Mean(X)Mean(Y)$$

3.4 Logging Data Model

The Logging Data Model (LDM) describes the content and format of data being logged by SDG. SDG uses relational databases to store the logged data. In this case, the LDM is a basically a relational schema. HLA rules state that the FOM shall document the agreement among the federates on data to be exchanged at runtime. The ISI LDM is automatically generated from the FOM description. FOM describes objection classes and interactions. Classes have attributes, and interactions have parameters. The attributes and interactions are defined in terms of primitive data types and complex data types. For each object class, there is one top-level table in the relational schema. Attributes are mapped to multiple columns, to rows in a sub-table or to multiple columns in the top-level table. Complex data types with high or unbounded cardinality, such as arrays, are mapped to sub-tables. The sub-table contains keys referencing the parent table, sequence column indicating the order in the array and data columns representing the actual data. Interactions and their corresponding parameters are handled similarly to the object classes and their attributes. The purpose of the relational schema is for the efficient storage of log data intercepted during the federation execution. It has been pointed out that the primary purpose of relational schema is data integrity (Uschold ,2004). Foreign key constraints are used on the ADM. (Figure 7).

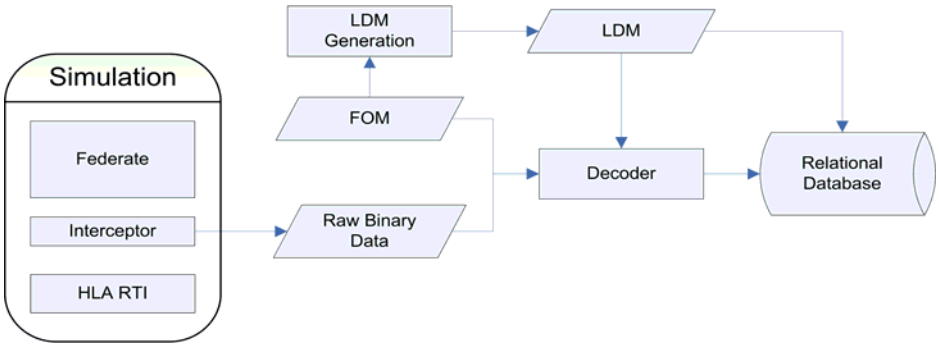

Figure 7
Logging data flow

HLA rules state that all exchanges of FOM data among federates shall occur via the RTI, and that federates shall interact with the RTI in accordance with the HLA interface specification. RTI-s is a highly-scalable implementation of the RTI (Helfinstine *et. al.*, 2003). SDG exploit RTI's plug-in to intercept and log messages federate attribute updates and interaction sends.

3.5 Distributed Logging and Analysis

The sensors should be able to detect enemy forces and simulating such urban environments requires tremendous amount of distributed computer resources (Lucas & Davis, 2003). To work in distributed environments an additional layer is needed to define on top to aggregate multidimensional cubes distributed across different machines. The left-hand side of Figure 10 depicts a single three-dimensional sensor/target/detection status score-cube. It represents only a partial, incomplete view. To generate a complete view, cubes from other simulation federates have to be aggregated. The right-hand side of Figure 8 depicts a tree summing together all the distributed cubes. Again, the associative and commutative properties of the aggregation operator are used, while the raw data is not sent.

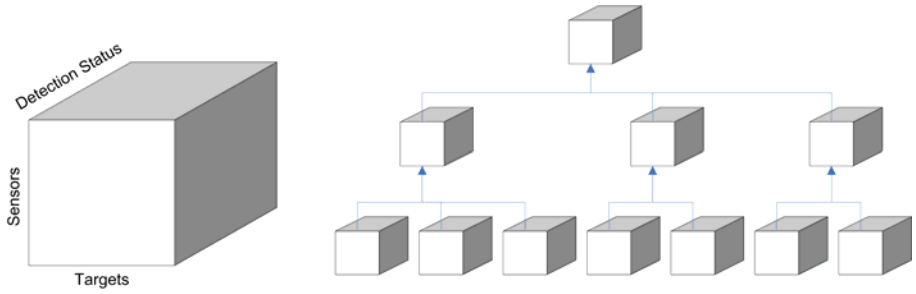

Figure 8
Distributed Data Analysis

3.6 Data Management Analysis

The ability to capture and log detail message traffic from very large scale simulations has exceeded the ability of humans to analyze and comprehend that data. A framework for quickly translating these operational-level log data into analyst-level data has been implemented. The frame-work explicitly defines a two-level data model that separates the operational logging data model from the analysis data model. The agility of the framework results from being able to isolate changes to the logging data model as a result of

changes to the federation object model, and from being able to quickly define analysis data model that match analyst notion of measure of effectiveness and of performance.

4 GPGPU ACCELERATION IN SIMULATION OPERATIONAL SETTINGS

4.1 Research Objectives and Methods

The objective of the effort was to provide stable, distributed and scalable compute resources to JFCOM. Existing DOD simulation codes were implemented on a new cluster, enhanced with nVidia 8800 GPUs. Acceleration targets include: line-of-sight calculations, physics-based phenomenology, CFD plume dispersion, data analysis, *etc*. The GPU has long been a very attractive candidate as an accelerator for computational hurdles, but previous generations of accelerators, *e.g.* Floating Point Systems (Charlesworth 1986), were for the small market of science and engineering, as opposed to current GPUs that are mass-marketed for gaming. However, in order to get any meaningful speed-up using the GPU, the data transfer and host-GPU interaction had to be minimized. The analyst/programmer must carefully isolate those subroutines that are amenable to GPU acceleration and are limiting the performance of the overall program.

4.2 Results

The initial year of research on the GPGPU-enhanced cluster at JFCOM, Joshua, was marked with typical issues of stability, O/S modifications, optimization and experience. Yet, even with incomplete utilization of the GPUs, the goal of enabling larger global scale experimentation was exceeded when more than 10M entities were sustained on appropriate terrain with valid phenomenology. (Figure 9)

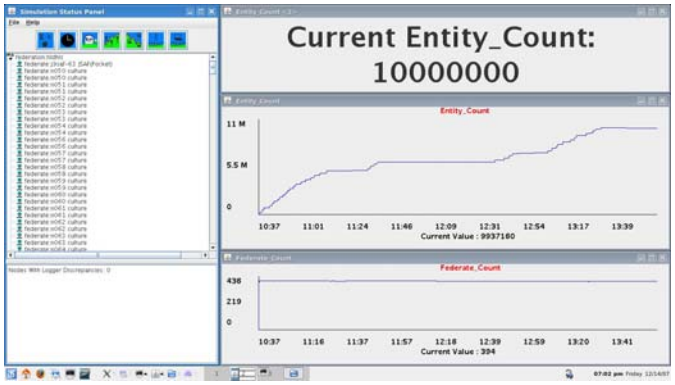

Figure 9
Screen Capture of Ten Million Entity Run

4.3 Analysis of Additional Potential Uses

The need for higher fidelity simulations is on the rise, *e.g.* complex experiments in dense urban environments. There is a general consensus that there are two common ways to improve simulation: (a) by increasing entity counts (quantitatively) and (b) by increasing realism (qualitatively). Numerous efforts have been made to increase the former, *e.g.* SF Express (Brunnet, *et al.* 1998) and Noble Resolve (USJFCOM 2006). These included the use of the Scalable Parallel Processors (SPP) or Linux clusters (Wagenbreth, *et al*. 2005). JFCOM teams have made great strides in improving entity behavior models (Ceranowicz, *et al*. 2002 and 2006) by adding more realistic entity behaviors. GPUs can be employed to address these issues. The authors found that CUDA was easily implemented by journeyman programmers and that, while there were exceptions, the general speedup expected was in the 2X to 3X range for J9

codes. JFCOM recognized the potentials behind the general-purpose graphics-processing unit (GPGPU) computing paradigm and, with ISI assistance, obtained a GPU-enhanced cluster (Figure 10).

Figure 10
Joshua cluster at JFCOM, 256 nodes with NVIDIA GTS8800 GPUs

In the quest to advance the broader use of GPUs (Lastra 2004), the new Compute Unified Device Architecture (CUDA) programming language has made GPUs more accessible to programmers(Buck, 2007). Some potential areas of improvements to the JSAF simulation were identified, *e.g.* the use of GPU for the route-planning. During that stage there is a huge computational load that can be offloaded to the GPUs. Another area of interest is the combination of GPUs and others accelerators such as FPGAs. The successful use of FPGAs as accelerators has been reported (Linderman, 2005) and they are installed on compute nodes of some Linux clusters. Ideally, some future CPU-GPU-FPGA configuration would allow a designer to take advantage of the strengths of FPGA and GPU accelerators. The raw integer power and reconfigurability of an FPGA is key to cryptography and to fast folding algorithms (Frigo, 203). Clusters could be GPU-enhanced for linear algebra (Fatahalian, 2004) and FFT operations (Sumanaweera, 2005).

5 HIGH PERFORMANCE COMPUTING AND ANALYSIS CONCLUSIONS

Agent-based simulations have more and more requirements in several dimensions: size, speed, resolution, and behaviors. Technology continues to deliver faster, more detailed, more sophisticated and more exploitable products. This paper set out three emerging technologies that will likely become mainstays of the simulation community's tool box in the next decade. Many simulations call out for the capabilities set forth above and many more can make use of these to improve their already useful products. The technologies were set forth is what is hope to be sufficient detail to allow the researcher to understand their power, analyze their limitations, assess their costs in dollars, manpower and disruption. Being academics, the authors seek no recompense for their previous efforts and gladly offer up any future assistance they could provide. The work shows new abilities to bring new compute power to bear in order to generate the simulation, to use that power to better analyze the data, to more effectively move the data around the country and more efficiently store the data in such a way as to make it more accessible and more useful.

ACKNOWLEDGEMENTS

Thanks are due to the excellent staffs at JFCOM, ASC-MSRC and MHPCC. Some of this material is based on research sponsored by the Air Force Research Laboratory under agreement number FA8750-05-2-0204. Other work is based on research sponsored by the U.S. Joint Forces Command via a contract with the Lockheed Martin Corporation and SimIS, Inc. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views

and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of these organizations.

REFERENCES

Barrett, B. & T.D. Gottschalk. 2004. Advanced Message Routing for Scalable Distributed Simulations, in the *Proceedings of the 2004 Interservice/Industry Training, Simulation and Education Conference,* Orlando, Florida.

Brunett, S., & T.D. Gottschalk. 1998. A Large-scale Meta-computing Framework for the ModSAF Real-time Simulation, *Parallel Computing*:, V24:1873-1900, Amsterdam

Buck, I., 2007. GPU Computing: Programming a Massively Parallel Processor, *International Symposium on Code Generation and Optimization,* San José, California

Ceranowicz, A. & M. Torpey. 2005. Adapting to Urban Warfare, *Journal of Defense Modeling and Simulation*, 2:1, January 2005, San Diego, Ca

Ceranowicz, A., M. Torpey, B. Helfinstine, J. Evans, & J. Hines. 2002. Reflections on Building the Joint Experimental Federation, in the *Proceedings of the 2002 Interservice/Industry Training, Simulation and Education Conference,* Orlando, Florida.

Charlesworth, A., & J. Gustafson, J. 1986. Introducing Replicated VLSI to Supercomputing: the FPS-164/MAX Scientific Computer, in *IEEE Computer*, 19:3, pp 10-23, March 1986

Dongarra, J., 1993. Linear algebra libraries for high-performance computers: a personal perspective, *Parallel & Distributed Technology: Systems & Applications*, IEEE, Feb. 1993, Volume: 1, Issue: 1, pp: 17 – 24

Fatahalian, K., Sugerman, .J. & Hanrahan, P., 2004. Understanding the efficiency of GPU algorithms for matrix-matrix multiplication, *Workshop on Graphics Hardware*, Eurographics/SIGGRAPH

Frigo., J., Palmer, D., Gokhale, M., and M. Popkin-Paine, M. 2003, Gamma-ray pulsar detection using reconfigurable computing hardware, *11th Annual IEEE Symposium on Field-Programmable Custom Computing Machines FCCM*.

Gottschalk, T., P. Amburn, & D. Davis. 2005. Advanced Message Routing for Scalable Distributed Simulations, *The Journal of Defense Modeling and Simulation,* Volume 2. Issue 1: 17-28, San Diego, California

Graebener, R., G. Rafuse, , R. Miller., & L-T. Yao. 2003. The Road to Successful Joint Experimentation Starts at the Data Collection Trail—Part II in the *Proceedings of the 2003 Interservice/Industry Training, Simulation and Education Conference,* Orlando, Florida.

Helfinstine, B., M. Torpey, & G. Wagenbreth. 2003. Experimental Interest Management Architecture for DCEE. In the *Proceedings of the 2003 Interservice/Industry Training, Simulation and Education Conference,* Orlando, Florida.

Kimbal, R., L. M. Reeves, M. Ross, & W. Thornwaite. 1998. *The Data Warehouse Lifecycle Toolkit*. Hoboken, New Jersey: Wiley.

Lastra, A., M. Lin, and D. Minocha, 2004. *ACM Workshop on General Purpose Computations on Graphics Processors*, Los Angeles, California

Linderman, R. W., Linderman, M. H. and Lin, C-S., 2005. FPGA Acceleration of Information Management Services, *2005 MAPLD International Conference,* Washington, DC

Lucas, R., & Davis, D., 2003. Joint Experimentation on Scalable Parallel Processors, in the *Proceedings of the 2003 Interservice/Industry Training, Simulation and Education Conference,* Orlando, Florida.

Messina, P. C., S. D. Brunett,., M. Davis, and T. D. Gottschalk. 1997. Distributed Interactive Simulation for Synthetic Forces, In *Mapping and Scheduling Systems, International Parallel Processing Symposium*, Geneva

Sumanaweera, T. and D. Liu. 2005. *Medical Image Reconstruction with the FFT, in GPU Gems 2*, M. Pharr, Ed. Boston: Addison-Wesley

Tran, J.J , R.F. Lucas, D.M. Davis, G. Wagenbreth, K-T Yao & D.J. Bakeman. 2008. A High Performance Route-Planning Technique for Dense Urban Simulations, in the *Proceedings of the 2008 Interservice/Industry Training, Simulation and Education Conference*, Orlando, Florida.

Yao, K.-T., & Wagenbreth, G. 2005. Simulation Data Grid: Joint Experimentation Data Management and Analysis. In the *Proceedings of the 2005 Interservice/Industry Training, Simulation and Education Conference*, Orlando, Florida.

Yao, K-T., C. Ward & G. Wagenbreth. 2006. Agile Data Logging and Analysis, in the *Proceedings of the 2006 Interservice/Industry Training, Simulation and Education Conference*, Orlando, Florida.

AUTHORS

THOMAS D. GOTTSCHALK is a Member of the Professional Staff, a Senior Research Scientist at the Center for Advanced Computing Research (CACR), and Lecturer in Physics all at the California Institute of Technology. He has worked at CACR for more than a decade advancing the use of massive parallel computers for simulation. His instructional duties include Statistics and Experimental Design for Caltech Physics Graduate students. Dr. Gottschalk has been active in parallel programming for nearly twenty years, with efforts spanning integrated circuit design, intelligent agent simulations, theater missile defense, and physics modeling. He consults for a number of other organizations, including his work on space-based systems for the Aerospace Corporation. He received a B.S. in Physics from Michigan State University and a Ph.D. in Theoretical Physics from the University of Wisconsin. His email address is <tdg@cacr.caltech.edu>.

KE-THIA YAO is a research scientist in the Information Sciences Institute (ISI) and a lecturer at the Viterbi School of Engineering, both at the University of Southern California (USC). He teaches a popular course in Data Mining at USC and has been working on the JESPP project at ISI. Within the JESPP project he is developing a suite of monitoring/logging/analysis tools to help users better understand the computational and behavioral properties of large-scale simulations and he has developed the Scalable Data Grid to manage the immense volume of data distributed from Virginia to Hawai'i. He received his B.S. degree in EECS from UC Berkeley, and his M.S. and Ph.D. degrees in Computer Science from Rutgers University. His email address is <ktyao@isi.edu>.

Gene Wagenbreth is a Systems Analyst for Parallel Processing at USC's ISI, doing research at the Computational Sciences Division. He has been a guest lecturer in Computer Sciences at the Viterbi School of Engineering at USC. He has extensive experience in agent based modeling and discrete element simulations. He has nearly four decades of research management experience. He has a BS in Math/Computer Science from the University of Illinois, 1971 His email address is <genew@isi.edu>.

DAN M. DAVIS is Director, JESPP Project, ISI at USC and has been active in large-scale distributed simulations for the DoD since 1988 at Caltech, the Maui High Performance Computing Center and at ISI. He was the Assistant Director of the CACR Research at Caltech, managing the Synthetic Forces Express simulation project. Prior to that, he was a Software Engineer on the All Source Analysis System project at the Jet Propulsion Laboratory and worked on a classified project at Martin Marietta, Denver. An active duty Marine Cryptologist, he retired as a Commander, USNR, Cryptologic Specialty. He has served as the Chairman of the Coalition of Academic Supercomputing Centers and the Coalition for Academic Scientific Computation. He received a B.A. and a J.D., both from the University of Colorado in Boulder.. His email address is <ddavis@isi.edu>.