Nondisruptive Data Logging: Tools for USJFCOM Large-scale Simulations

Gene Wagenbreth
Dan M. Davis
Dr. Robert F. Lucas
Dr. Ke-Thia Yao
Craig E. Ward
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Ste 1001
Marina del Rey, CA 90292
310 448-8213, 310 448-8434, 310 448-9449
genew@isi.edu, ddavis@isi.edu, rflucas@isi.edu

Keywords:
Data management, logging, HLA, RTI, ABM simulations.

ABSTRACT: *A persistent series of issues and problems in training, simulation and education are those associated with the efficient and effective collection of information. These issues become exacerbated as the data being generated increasingly outstrips the human capacity to sense, remember and analyze the results of the effort. These shortcomings can occur in several dimensions, including volume of information (data glut), speed of transmission to the analyst (input overload), and geographical dispersion of the data (distributed data). The authors have been wrestling with these issues for more than a decade and present both a theoretical overview and a practical set of lessons learned based on those experiences. A review of the problems and a survey of various approaches to their solution will be presented. One of the issues most problematic is the need to reliably collect all of the germane information that is required or desired by the user or analyst without negatively impacting the training, simulation or education that is the primary focus of the activity. Acknowledging theoretical limits to non-intrusive observation, the goal is held to be: maximize the data and minimize the disruption. The authors lay out with specificity their development of the JLogger system to collect needed information out of the Joint Forces Command's experimentation that typically uses Joint Semi-Automated Forces (JSAF) simulations for analysis and evaluation. Examples of problems faced in achieving the goals of the experiments, the approaches used to resolve these and the solutions developed are all presented with a view toward assisting other similarly-tasked professionals with assessing their needs, their problems and their opportunities. The authors conclude by laying out the way the data were collected, then were structured to optimize their usability by the warfighter participants. Then they look to the future of data collection during live, virtual and constructive events.*

1. Introduction

The U.S. Joint Forces Command (USJFCOM) conducts an on-going series of battlespace simulations, typically set in urban environments requiring demographically correct numbers of civilian entities. These events can be used for the traditional uses of computer combat simulations: training, analysis or evaluation. These simulations generate huge amounts of data, but the collection of that data is to some very real degree secondary to the main goal of the event, giving the participant a credible and valid experience.

This activity orientation is counter to the data managers desire to design programs to facilitate data recovery. Therefore, the first rule on USJFCOM events has been to avoid disrupting the live, virtual or constructive simulations. Yet, recording what happened is valuable for a number of reasons, to be discussed later.

The challenges the authors' team faced was coming up with a design that was non-disruptive, provided some real-time queries, minimize transmission of data between the trans-continentally distributed sites, archive the important data so that it can be analyzed later and do all of this during the hectic development spirals building up to the next event.

This paper will first set the stage, then discuss the problems, review the approaches, relate the experiences and discuss the lessons to be learned.

1.1. JFCOM's Mission and Mandate

USJFCOM J9 develops innovative joint concepts and capabilities providing experimentally proven solutions to the most pressing problems facing the joint force. Operationally relevant solutions are rapidly delivered to support current operations and drive changes to better enable the future joint force. JCD&E provides thought leadership and collaborative environments to generate innovative ideas with a range of interagency, multinational, academic and private sector partners. J9 leads and coordinates JCD&E for DoD through an enterprise approach.

The USJFCOM mission is based on strategic guidance within the context of the Joint Operating Environment (JOE) [1] and Capstone Concept for Joint Operations (CCJO) [2]. These are the conceptual blueprints for how the U.S. Armed Forces will leverage technological advances and integrate new operational concepts to foster innovative military approaches for how U.S. forces will conduct future operations. As USJFCOM's experimentation arm, JCD&E has produced a series of major simulation events and experiments in which warfighters in uniform are staffing the consoles during interactive simulations.

In order to simulate the complexities of urban warfare, JCD&E has relied on well-validated entity-level simulations, e.g. Joint Semi-Automated Forces (JSAF) and the Simulation of the Location and Attack of Mobile Enemy Missiles (SLAMEM). These need to be run at a scale (millions of autonomous entities) and resolution (zoomable from global to near photo-realistic street views) adequate in order to engage participants fully and sustain combat realism impact. Some of the growing need for realism is driven by the heightened expectations of current service personnel who grew up playing photo-realistic games.

In order to meet these goals, USJFCOM required the dedicated use of an enhanced Linux cluster of adequate size, power, and configuration to support simulations of more than 2,000,000 entities within high-resolution insets on a global-scale terrain database. If required, this facility can be used to interact with live exercises, but more typically it is engaged interactively to present users and experimenters with virtual or constructive simulations. [3] This technology has proven to be robust and to reliably support hundreds of personnel committed to the experiments. It has been scalable enough to easily handle activities both small and large. These experiments have been used to assess the value of new sensors, tactics and equipment in an environment that allows Humans-In-The-Loop. This capability deepens the analyses by engaging the warfighters themselves and monitoring their reactions.

2. JFCOM and High Performance Computing

The U.S. Joint Forces Command has become one of the premier High Performance Computing (HPC) users in the DoD. Research there has produced some forty papers over the last few years.

2.1. Joint Futures Lab (JFL) at USJFCOM

Creating a standing experimentation environment that can respond immediately to DoD time-critical needs for analysis is one of the major, but difficult to achieve, goals of the Joint Futures Lab in Suffolk. The JFL must operate in a distributed fashion over networks such as the Defense Research and Engineering Network (DREN), at a scale and level of resolution that allowed USJFCOM and its partners to conduct experimentation on issues of concern to combatant commanders, who participated in the experiments themselves. This provided the requisite simulation federations, software, and networks, joined into one common computer infrastructure that was supporting experiments. Quantitative and qualitative analysis, flexible plug-and-play standards, and the opportunity for diverse organizations to participate in experiments were all mandated by DoD needs, and those general mandates remain.

2.2. Joint Advanced Training and Technologies Laboratory

Supporting mission rehearsal, training, operational testing, and analysis remain the goals of the Joint Advanced Training and Technologies Laboratory (JATTL). The principle concerns of the JATTL were developing technologies that support the pre-computed products required for joint training and mission rehearsal. One of the new concepts, the Joint Rapid Scenario Generation (JRSG), as well as others, was seen to be required during its execution. The latter include phenomenology such as environment, cultural assets, civilian populations, and other effects necessary to represent real operations. The JATTL is connected world-wide via networks such as, but not limited to, DREN, Joint Training & Experimentation Network (JTEN) and the National Lambda Rail (NLR) to over forty Joint National Training Capability sites. The DREN represents the "E" in the JTEN.

2.3. USJFCOM's JESPP

The Joint Concept Development and Experimentation (JCD&E) at USJFCOM designed, programmed and operationally implemented a scalable simulation code that quickly was shown capable of modeling more than 1,000,000 entities. Called the Joint Experimentation on Scalable Parallel Processors (JESPP) project [4], this was a continuation of an earlier DARPA/HPCMP project named SF Express. [5] The early JESPP experiments showed that the code was scalable well beyond the 1,000,000 entities actually simulated.[6]

At first, JESPP was successfully fielded and operated using USJFCOM's HPCMP-provided compute assets hosted at Aeronautical Systems Center (ASC) Major Shared Resource Center (MSRC), Wright Patterson AFB, and at the Maui High Performance Computing Center (MHPCC) in Hawai'i. Those dedicated systems suitable and reliable for day-to-day use, both unclassified and classified. The dedicated compute power that was provided did require identification, collection, and analysis of the voluminous data from these experiments, which was the focus of the work of Dr. Ke-Thia Yao and his team. [7]

USJFCOM personnel in Suffolk Virginia worked with a "Red Team" in Fort Belvoir Virginia, a civilian control group at SPAWAR San Diego, California, and players at Fort Knox Kentucky and Fort Leavenworth Kansas, totaling up to 1,500 personnel.. (Figure 1)

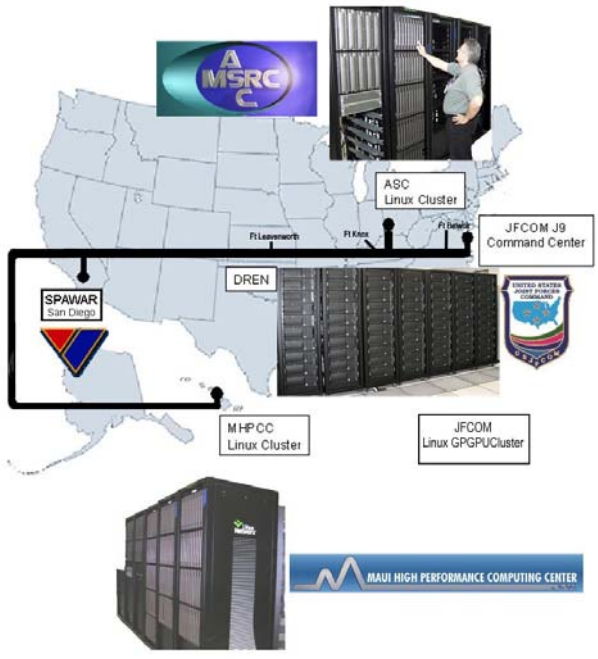

Figure 1. USJFCOM's Experimentation Network

Even using these powerful computers, the USJFCOM experimenters were constrained in a number of dimensions, *e.g.* number of entities, sophistication of behaviors, realism of various environmental phenomenology, etc. While the scalability of the code would have made the use of larger clusters feasible, a more effective, efficient, economical and elegant solution was sought. The end of the Moore's Law speed and power improvements decreed that eventually USJFCOM would need to seek new ways to deliver their services to increasingly sophisticated users with increasingly complex and geographically diverse conflict environments. One approach is the use of more powerful clusters, especially a GPU-enhanced cluster like *Joshua*.

2.4. Value of HPC to the Simulation Community

While precise quantification of the value of the USJFCOM cluster or one like it may be elusive, it is possible to consider the cost savings of the USJFCOM activities, which are, in turn, enabled by the HPC system. In addition, there are incalculable benefits from battlespace simulation when one looks at the need to understand and train for urban conflict, while realizing the impossibility of conducting full-scale military exercises in downtown U.S. cities. The same holds true, perhaps even more so, when the reaction to the loss of life of team members and allied forces needs to be felt in as realistic way as possible, while not even risking the loss of life and injury actually experienced in live combat military exercises.

Not uncommonly USJFCOM needs to simulate large engagements in the constructive mode, utilizing only a few military Subject Matter Experts (SMEs). Clearly, one of the more difficult issues is to parse out how much is actually saved. The personnel who would be used in the live exercise are already incurring some of the costs. No matter how effective a simulation, some live training is needed. On the other hand, tank crews returning from the Gulf reported that the best training they had was the simulator training in the tank mock-ups.

Further, putting a value on the cost to society of an accidental death during a live exercise, on an injury to a soldier or even on the wear and tear on very expensive military assets, is really beyond the scope of this paper, yet should resonate with all engaged in this analysis. As new threats emerge, as even more sophisticated equipment needs evaluation and more troops need to be trained, and as public sensitivities about military activities increase, even more burden will be born by the simulation community The technology being researched, developed and

implemented here will play a vital role in enabling that capability.

2.5. Utility of Forces Modeling and Simulation

Forces Modeling and Simulation (FMS) is a field often driving new computer science due to its virtually unlimited needs and open-ended requirements. Nothing short of a fully realistic experience, resting on a global model to obviate the edge effects of a restricted synthetic environment is sought. This field is unique to the DoD, compared to many of the other standard science disciplines, *e.g*. Computational Fluid Dynamics (CFD) and Weather. In a similar way, interactive computing is a new frontier being explored by the JESPP segment of FMS, coordinating with a few other user groups. Along these lines, the newly enhanced Linux Cluster capability has provided significant synergistic possibilities with other computational areas such as signals processing, visualization, advanced numerical analysis techniques, weather modeling and other disciplines or computational sciences such as SIP, CFD, and CSM.

2.6. HPC use at USJFCOM

Research objectives for the USJFCOM system were to develop a platform and system administrator cadre to provide 24x7x365 enhanced, distributed and scalable compute resources to enable joint warfighters at USJFCOM as well as its partners, both U.S. Military Services and International Allies and to advance the technology of GPGPU utilization within the DoD. These enabled the users to develop, explore, test, and validate 21st century battlespace concepts and capabilities in JCD&E's JFL. The specific immediate goal was to enhance global-scale, computer-generated support for experimentation by sustaining more than 2,000,000 entities on appropriate terrain, along with valid phenomenology.

2.7. Implementation Strategy

The use of existing DoD simulation codes on advanced Linux clusters operated by USJFCOM was the implementation method chosen as being most efficient. This effort supplanted the previous USJFCOM JCD&E DC clusters with a new cluster enhanced with 64-bit CPUs and nVidia 8800 graphics processing units (GPUs). Further, the authors were able to modify a few legacy codes. As noted above, the initial driver for the FMS use of accelerator-enhanced nodes was principally the faster processing of line-of-sight calculations. Envisioning other acceleration targets is easy: physics-based phenomenology, CFD plume dispersion, computational atmospheric chemistry, data analysis, *etc*. The computer (Figure 2) was given the name *Joshua*, the biblical Hebrew commander.

Figure 2. *Joshua*

The first experiments were conducted on a smaller code set, to facilitate the programming and accelerate the experimentation. An arithmetic kernel from a Mechanical Computer Aided Engineering (MCAE) "crash code" [8] was used as vehicle for a basic "toy" problem. This early assessment of GPU acceleration focused on a subset of the large space of numerical algorithms, factoring large sparse symmetric indefinite matrices. It made use of the SGEMM (Single precision GEneral Matrix Multiply) algorithm [9] from the BLAS (Basic Linear Algebra Subprograms) routines. [10]

.

As an accelerator for computational hurdles such as sparse matrix factorization, the GPU was seen as a very attractive candidate. Previous generations of accelerators, such as those designed by Floating Point Systems [11] were for the relatively small market of scientific and engineering applications. Contrast this with GPUs that are designed to improve the end-user experience in mass-market arenas such as gaming. Meaningful speed-up was observed, on the order of an overall acceleration of 2X, using the GPUs. It was determined that the data transfer and interaction between the host and the GPU had to be reduced to an acceptable minimum.

2.8. Results of Scaling Effort

While great strides were made in the number of entities to be simulated, the research implications of this achievement, albeit expected, were a dramatic reaffirmation of the value of HPCs, but it does add to the data management burden.

Starting a low level of a few thousand entities, HPC has enabled a steady growth for about a decade. Most notably, on 14 December 2007, the USJFCOM personnel, under the leadership of Rich Williams simulated a full ten million CultureSAF entities on the Baghdad terrain database. This was accomplished. (Figure 3) below shows the expansion over a decade.

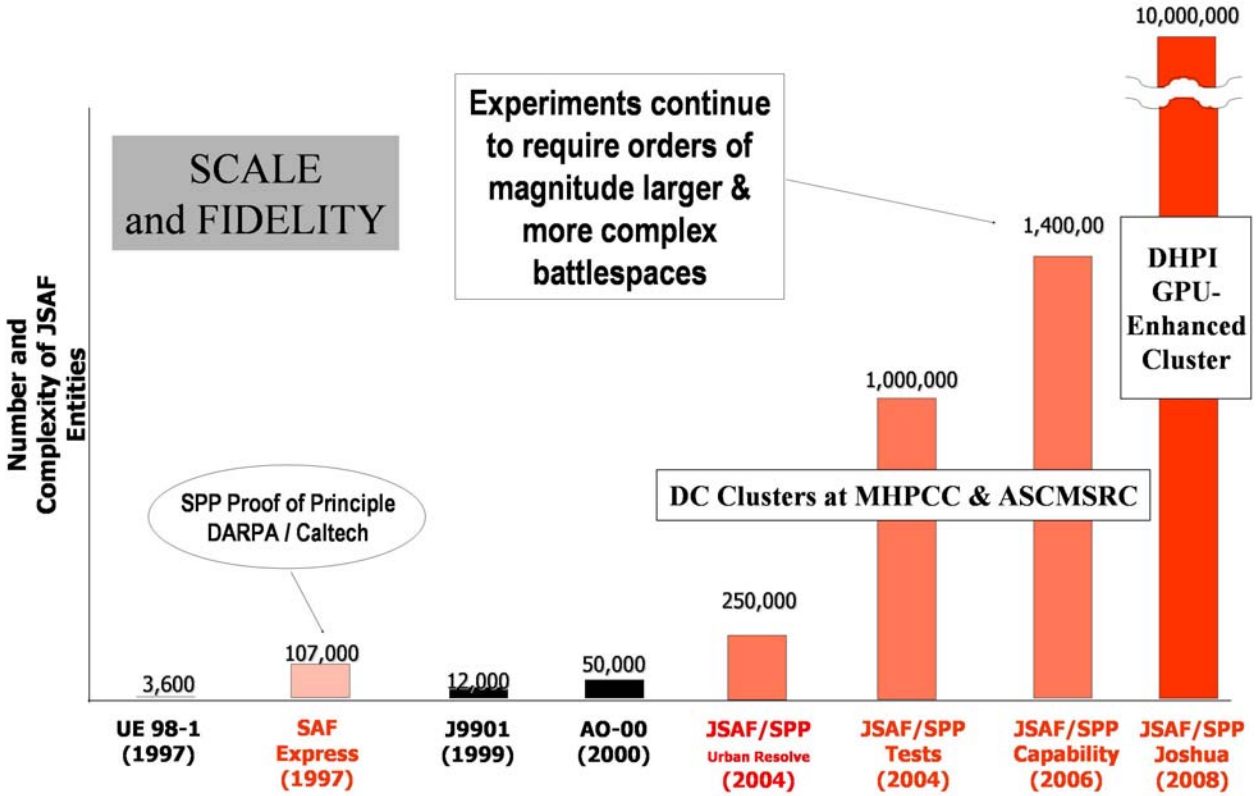

Figure 3. Increase of Number of HPC Enabled SAF Entities 1997 - 2008

3. Logging and Storage

With the increase in the data concomitant with the nearly four orders of magnitude increase in entities, clearly there was a problem. [12] While they limited by using the PC's-on-a-LAN approach, the simulation operations group had been using Microsoft's Access data base which is provided with MS Office. The HLA/RTI interfaces allowed certain data to be extracted from the simulation and stored in Access files. Something larger and more scalable was going to be needed as the capabilities of the simulations were further expanded with more scalable code and using the larger computer clusters which were the HPC assets. The new system generated on the order of ten Terabytes a day..

One option considered was staying with Access. Another was acquiring one of the large proprietary data bases that are on the market. The third option was that of using one of the open source, scalable data bases. The latter one was the option that was chosen, for three basic reasons:

1. Cost
2. Scalability
3. Open access to source code

Security was not a large concern, as all JFCOM exercises are classified and take place on closed systems, using dedicated, encrypted DoD nets.

The team began the new implementation using SQLite, but quickly migrated to MySQL. The entire team experienced the benefits of a very short learning curve and were quickly very comfortable working with this program. To the best of the authors' knowledge, no one has seriously questioned the wisdom of that choice since then.

HLA/RTI has a capability that can be initiated by entries in the RTI Initialization Data (RID) file that can establish a data path for the published data in the publish and subscribe system used in JSAF. Various parameters can be set for filtering the data, which was necessary because the team did not want to "capture"

all of the data, finding it prudent to discard most of the civilian location and movement data, which, while vital to the experiments, was considered as "clutter: by the analysts, so precise location records were of less importance that "red" and "blue" forces and sensor data.

The ISI crew wrote several C++ routines to cull out the data required, always focusing on not burdening local compute nodes with communications or creating storage tasks that would appreciably interfere with the simulation and the users' perception of realism and their interaction with it. The team conceived, implemented and developed a strategy that allowed the information to stay resident on each node's disk, thereby insuring the unfettered use of inter-node communications channels by the simulation and visualization routine data streams.

The analysts required a system that allowed the polling of active nodes for information that was needed during the exercise. This clearly was more problematical, but the team relied on decades of experience working with HPC and with large data bases. They designed a system that allowed simple queries to flow up and down the distribution tree without disrupting the simulation. This initial system had limited ability to do "joins", *i.e.* combinations of parameters in one query.

3.1. JLogger Implementation

JLogger is a general-purpose data logging tool. It is used in conjunction with JSAF to be able to log all HLA object and interaction messages that are output during a simulation run. The logged messages are captured and stored in a relational database management system, in this case, MySQL.

For technical descriptions of the format of the data produced by JSAF, culture, and the associated sensor federates used at USJFCOM, see the work by Graebener *et al.* [13] A higher-level and more germane general description of the data available for collection and the steps necessary to render it useful to the analysts and experiment controllers is given below. It is hoped that this will allow members of the simulation community to better assess similarities between their simulations and the USJFCOM simulations and provide a basis for analyzing the relevancy of this work.

For clarity's sake, the USJFCOM simulations were broken down into four broad areas:

1. Terrain and Environment
2. Civilian or "culture"
3. Operational entities
4. Sensor platforms

Surprisingly, Terrain and Environment does not represent a major data task in comparison to the others. This is due to the fact that the terrain is a largely static entity and the environmental variables are often easily duplicated without storing the actual values during the experiment, *e.g.* the day/night interface is easily recovered; however, the impact of clouds may be more random, *ergo* may need to be recorded.

On the other hand, the amount of data presented by the clutter of civilians can be huge. Positions of pedestrian entities and vehicle models (numbering in the millions) are reported to the system via "publication" on the order of once every 10 to 100 milliseconds. Being able to simulate millions of entities, 14] now presents the problem of what to do with all of this location data (in three dimensions), orientation data (in three axes) and state data (color, type, damaged, dead, …) that can report as often as 100 times in a second.

As an early approach, the JESPP team developed the first instantiation of a solution for saving the data so generated. [15] However, as will be discussed below, the volume of civilian data was so high that it was simply discarded, treating it much as if it were environmental data, albeit more dynamic.

The Operational Entities are largely armed forces of US, Allied and enemy units. JSAF, like all of the SAFs, presents the possibility of very good Human-In-The-Loop (HITL) intervention. That means that USJFCOM military personnel can personally control US and Allied forces and a "Red" team can bring a hostile presence to the experiment with all of the creativity and experience they have garnered, frequently after decades in the service. This brings to the surface a significant difference between USJFCOM data and, say, Bank of America "transaction data" or high energy physics "sensor data." The solutions sought for DoD use in both the software and hardware areas must be sufficiently general to support any conceivable data type and load, yet sufficiently specific to generate the validity necessary for USJFCOM use. Neither the banker nor the physicist has so little control over their parameters.

Finally, there is the data to be gleaned from the intelligence sensors simulation programs. Currently, in USJFCOM experiments, the sensor simulations generate nearly a terabyte of data per week. Were this and all the rest of the data to be collected at the resolution and sophistication that is possible, the amount would swell to around 80 terabytes per week. Just physically managing that much data is a daunting task. There is also the difficulty in reliably validating

the relationship between simulated sensor output and simulated activity on the terrain database. There is also the issue of facing proprietary data concerns when several DoD contractors are providing code and data..

3.2. Background for Logging Work

The need for a JLogger System was identified during the work above and was based on logging needs by USJFCOM. A design advanced and developed by ISI was adopted by USJFCOM and entered into JSAF's Concurrent Version System (CVS) as JLogger. JLOGGER was developed by the ISI team as a software system to support logging of "published" (as in "publish/subscribe") data by JSAF and other federates that use RTI.

3.3. Approach

JLOGGER code consists of the following segments:

- interceptor - library loaded at runtime by JSAF or other RTI federate to intercept published data and make it available to other JLOGGER software
- decoder - program which reads the interceptor output and saves it in various forms. It can send the data to a mysql which inserts it in a mysql database. It also writes it to disk. decoder can take input from these saved disk files instead of from the interceptor to facilitate offline after-action processing.
- aggregator - program which takes a single user input, sends it to multiple instances of mysql, reads and collates the results from the multiple mysqls, and sends the collated results back to the user. The user may be a program rather than a human user. Useful in distributed simulations.
- jloggerd - a collection of shell scripts to initiate, monitor, control and terminate the programs used in logging

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.

3.5. Logging Data Model Generation

HLA rules state that the FOM shall document the agreement among the federates on data to be exchanged at runtime. The JLogger LDM is automatically generated from a simulation file that 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 one top-level table within the relational schema was created. Simple primitive class attributes mapped to columns in the table. Attributes with complex data types many be either mapped to multiple columns, or to rows in a sub-table. Complex data types with fixed length and with low cardinality were mapped to multiple columns in the top-level table. For example, a location attribute with *x,y,z* coordinates are mapped in three columns. 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. Depending on the data type, it is possible to have sub-table of sub-table tables. 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 is not intended to capture all the information contained provided by the simulation. Uschold [16] points out that the primary purpose of relational schema is *data integrity.* For example, database foreign key constraints can be used to enforce that a row in a sub-table must have a corresponding row in the parent table. When deleting the parent row all the corresponding sub-table rows must also be deleted. However, such data integrity constraints are expensive to enforce. In practice the JLogger and other ISI data management tools identify these constraints, but do not explicitly enforce them on the collected data. However, foreign key constraints were used on the Analysis Data Model.

3.6. Intercepting/Decoding HLA Messages

The system is guided by HLA rules which state that all exchange of 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, which is a highly-scalable implementation of the RTI, [17] provides an interceptor plug-in framework that exposes calls to this HLA interface to registered plug-ins. JLogger exploited this plug-in to intercept and log messages federate attribute updates and interaction sends. With respect to the RTI, the contents of these messages are raw binary strings. RTI provides publish/subscribe facilities to exchanges these messages, but not decode their contents. To provide query and analysis capabilities, these messages must be decoded with respect to simulation data formats. The following diagram shows these flows. (Figure 4)

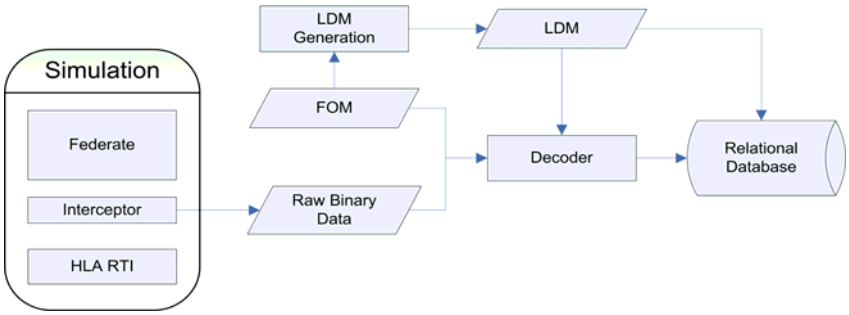

Figure 4. JLogger logging data flow

3.7. Mapping between LDM and ADM

The data intercepted from the simulation federates is stored according to the Logging Data Model (LDM). In order to provide useful information to the analysts, the data has to be abstracted and aggregated by JLogger and translated into the Analysis Data Model (ADM). See the diagram below. (Figure 5)

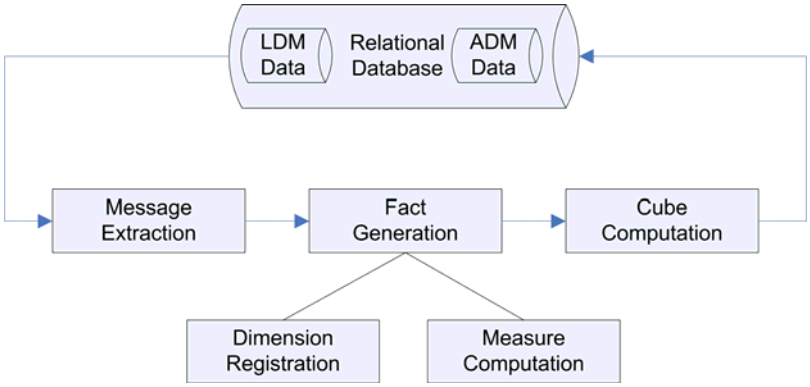

Figure 5. JLogger's mapping data stored in Logging Data Model then flowing to the Analysis Data Model

3.8. Message Extraction

The purpose of the message source extraction is to provide an interface layer that hides simulation data variations from the rest of mapping process. Message extraction performs include: filtering to keep on the columns needed by the analysis data model, and partitioning values. Currently, message extraction is implemented using SQL select statements.

As described in the previous section log data is stored within a relational database using a LDM schema. In practice, the information within the LDM schema is a super set of the information needed by analysis data model. Message extraction prunes away columns that are not needed. The rationale for partitioning is two-folds. One is the numerical precision that simulators are capable of is often not needed by the analysts who are interested in summary and aggregate information. Message extraction take the initial processing step needed to map the logged toward analyst's notion of dimension of interest. Second is such high precision generates large cubes that takes huge storage space. The analysts have to decide the minimum granularity he wishes to examine the data.

Either single numerical values or vector values can be used as the basis for partitioning. A common example of single numerical value is time. JSAF typically keeps time at the resolution of seconds. SDG logs message on the order of milliseconds. An example of vector values is geographical location. JSAF is capable of sub-meter location resolution.

Neither uniformity nor linearity are required of the partitions. In order to emphasize particular time periods

and locales where actions are taking place, analysts should be able to define smaller grain partitions, *e.g*. for an urban combat setting, the grid size can be set as relatively fine. Then the grid size can progressively be increased as the locus of action moves towards more rural environments

4. Scalable Data Grid

In order to better use all of this information, a better organization and taxonomy was needed. All of the data had to be retrieved and organized for archiving. It had to be filtered for unnecessecary duplication.

So far the paper has focused on creating a relational data base. A better approach was seen to be the addition of a capability to create multidimensional cubes. To work in distributed environments, an additional layer needed to be defined to aggregate multidimensional cubes distributed across different machines. The left-hand side of Figure 6 depicts a single three-dimensional sensor/target/detection status score-cube. This cube is generated from data logged from the local simulation federate. 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 6 depicts a tree summing together all the distributed cubes. Again, the associative and commutative properties of the aggregation operator can be used. The raw facts, which can be potentially bandwidth-intensive, do not have to be communicated across busy intermodal links. Only the cubes themselves have to be transmitted.

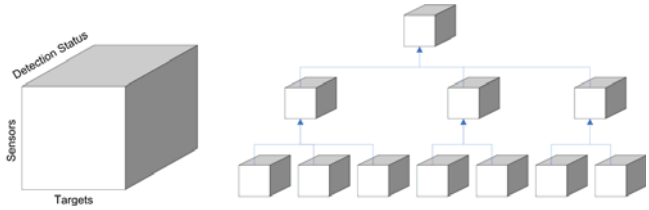

Figure 6. Distributed data analysis

4.1. Similar Efforts

Previous research in the simulation community has focused on creating data frameworks to facilitate simulation federates participating in federations using different definition files. Agile FOM Framework (AFF) describes a mapping methodology for translating back and forth between the internal federation SOM representation and the external FOM representation. [18][19] Types of AFF mappings include naming convention mappings (*i.e.*, Position-to-Location), attribute group mappings (three attributes representing location to one vector attribute), unit conversions, byte swapping and so on. In the work described in this paper, these types of mappings are not addressed. Potentially, an additional AFF framework layer could be defined. Instead storing the logging data in the FOM representation, the data could be stored after AFF mapping by means of the Agile FOM Interface.

Analysts may prefer one unit measurement representation over another. The other is that some of the low-level processing tools, such SQL scripts, do not have to be rewritten to adapt different naming and grouping conventions. Another interoperability approach is to provide a common Object Model Data Dictionary from which SOM and FOM developers can choose to incorporate into their models. [20] This approach can potentially work in tightly knit communities where developers can agree upon a common representation, and in domains where the modeled elements are relatively static and do not change frequently. Conceptually similar approach to the AFF is to use more formal knowledge presentations to describe the FOM and SOM models. [21] This work defines a world ontology in which FOM and SOM ontologies can be expressed. Then, a formal mapping is defined between the SOM and FOM ontologies.

4.2. SDG Analysis

The ability to capture and log detail message traffic from very large scale simulations is exceeding the ability to analyze and comprehend that data. The sections above described a framework for quickly translating these operational-level log data into analyst-level data that are capable of supporting decision makers. The framework 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.

5. Experience at USJFCOM

A typical experiment might include sites participating from across the country. An example would be one in which the TEC site at Fort Belvoir, Virginia, had 30+ workstations and Saber, a quad-CPU machine with four TeraBytes of disk space that were used for after-event storage. The SPAWAR site at San Diego, California, had 20+ workstations. The J9 Joint Futures Lab at Suffolk, Virginia, had 50+ workstations and a 16-node mini-cluster. The ASC Wright Patterson Air Force Base at Dayton, Ohio, had the Glenn cluster with 128 dual CPU nodes. The MHPCC site at Maui,

Hawaii, had the Koa cluster with 128 dual CPU nodes. This work was conducted prior to the acceptance of Joshua, the GPGPU-enhanced cluster.

These experiments typically ran five days a week, ten hours a day. Simulators might run all night, but with little activity and usually with logging disabled. Depending on availability and requirements, one or both of Glenn and Koa were used. Up to two hundred thousand clutter entities were simulated on the large clusters. (In this simulation, civilian entities are termed clutter, in that they serve to mask military entities.) Several thousand non-clutter entities were simulated on the other sites. A single node on the large clusters simulated 1000-2000 clutter entities.

Civilian data from the Glenn and Koa clusters was not entered into the Saber database, due to size limitations. Data from 100 nodes on Glenn for a ten-day event would have been close to a TeraByte. Data from TEC, SPAWAR, J9 and J9 mini-cluster for non-clutter entities were entered into the MySQL database. Urban Resolve Phase I exercise generated about a TeraByte of data in the MySQL database.

The nightly data transfer was about 15 gigabytes of compressed data. Network transfer rate to Saber was approximately ten megabits per second. Three or four hours was required to do the transfer. Decoding and indexing the data into the MySQL database took 12 hours if everything worked perfectly. Human error and other factors usually prevented a day's data from being entered into the database before the next day's event started. It was usually at least several days after an event before the complete after action database was ready on Saber.

The logging routines used for the four exercises in the old configuration were adequate. It was the first attempt at logging data from hundreds of processors distributed geographically around the country simulating thousands of non-clutter entities. SDG is intended to remove deficiencies in the old methodology and upgrade what was essentially an experimental system into a production system. The design parameters for SDG specifically address the following list of deficiencies in the old system:

1. Near real time and after action data logging are implemented differently. Near real time queries are restricted by the use of simple aggregators.
2. The use of a single database on Saber does not have the capacity to include clutter data from the Glenn and Koa clusters.
3. Data transfers, decoding and indexing are time consuming and error prone, delaying the availability of the database. A goal is to have the complete database kept up to data continuously.
4. Retrieval of data and database generation for multiple exercises is inconvenient.
5. Expansion to more compute nodes, more entities per compute node and more data per entity is impossible. Disk storage, compute power, and network bandwidth all impose serious limitations.
6. The system does not respond gracefully to hardware and network problems. Saber is a single point of failure that makes all data unavailable.
7. Complex queries that may be useful to analysts are slow or impossible.

6. Analysis and Conclusions

6.1. Analysis

When faced with seemingly intractable amounts of data and immutable demands for its effective use, exacerbated by the wide geographical distribution of data and users, the approach of implementing a system of non-disruptive data logging such as JLogger and a data management system such as SDG has been shown to possess four useful qualities:

1. Modest development time
2. Operationally stable
3. Future flexibility
4. Low cost

Most of these benefits were available to USJFCOM due to the simulation community experience of the DoD contracting personnel from Lockheed Martin and Alion, coupled with the HPC experience of the ISI team. It could be argued that even a small decrement of experience level on either team would not have produced a system that was nearly so capable. This would seem to suggest that careful selection and retention of the highest quality programming and system analyst personnel is still advisable. [22]

6.2. Conclusions

HPC capabilities are expanding data generation at very high rates. Interactive simulations using HPC can generate huge amounts of data in any given unit of time. Non-disruptive data logging is feasible and can be implemented by journeyman programmers in reasonable periods of time. Even after logged, effective use must provide for real time probes, archival storage, efficient organization, and easy retrieval. Distributed data management has been shown to be an effective approach for meeting all of these goals. Further research into several area is indicated. HPC can create data problems, but it can also answer those problems.

7. Acknowledgements

None of this research could have been successful without the support of USJFCOM leadership like Tony Cerri and Jim Blank, and the collaboration of Lockheed Martin personnel like Mark Torpey and Bill Helfinstine and Alion personnel like Col. Ray Dehncke and Dr. Andy Ceranowicz. This material is based on research sponsored by the U.S. Joint Forces Command via a contract with the Lockheed Martin Corporation and SimIS, Inc., and on research sponsored by the Air Force Research Laboratory under agreement numbers F30602-02-C-0213 and FA8750-05-2-0204. 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 the U.S. Government.

8. References

[1] USJFCOM, (2010). *Joint Operating Environment, 2010*. United States Joint Forces Command, Norfolk, Virginia, retrieved on 30 June 2010 from, http://www.jfcom.mil/newslink/storyarchive/2010/JOE_2010_o.pdf

[2] CJCS, (2009). *Capstone Concept for Joint Operations, V 3.0*. Chairman of the Joint Chiefs of Staff, Washington, DC, retrieved on 30 June 2010 from; http://www.jfcom.mil/newslink/storyarchive/2009/CCJO_2009.pdf

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

[4] Lucas, R., & Davis, D., (2003). Joint Experimentation on Scalable Parallel Processors, *2003 I/ITSEC Conference*, Orlando, FL

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

[6] Gottschalk, T., Amburn, P. &Davis, D., (2005), Advanced Message Routing for Scalable Distributed Simulations, *The Journal of Defense Modeling and Simulation*, San Diego, California

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

[8] Diniz, P., Lee, Y.-J., Hall, M. & Lucas, R., (2004). A Case Study Using Empirical Optimization for a large, *Engineering Application, in the Proceedings of the 18th International Parallel and Distributed Processing Symposium*, April 2004, pp: 200-208

[9] Whaley, R.C. & Dongarra, J.J., (1998), Automatically Tuned Linear Algebra Software, *IEEE/ACM Conference on Supercomputing*, SC98., pp.:38 - 38, Orlando, Florida

[10] 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

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

[12] Yao, K-T., Davis, D., & Lucas, R., (2006). Supercomputing's Role in Data Problems and Its Contribution to Solutions, *The ITEA Journal of Test and Evaluation* Vol 27, No 3

[13] Graebener, R., Rafuse, G., Miller, R. & Yao, K-T. (2003). The Road to Successful Joint Experimentation Starts at the Data Collection Trail *Proceedings of the Interservice/Industry Training, Simulation and Education Conference*, Orlando, Florida

[14] Barrett, B. & Gottschalk, T., (2004). Advanced Message Routing for Scalable Distributed Simulations. *Proceedings of the Interservice / Industry Training, Simulation and Education Conference* Orlando, FL.

[15] Wagenbreth, G., Yao, K-T., Davis, D., Lucas, R., and Gottschalk, T., (2005). Enabling 1,000,000-Entity Simulations on Distributed Linux Clusters, *WSC05-The Winter Simulation Conference*, Orlando, Florida

[16] Uschold, M. & Gruninger, M., (2004), Ontologies and semantics for seamless connectivity. ACM SIGMod Record, Vol 33, Issue 4, pp 58-64, New York, New York

[17] Helfinstine, B., Torpey, M., & Wagenbreth, G. (2003). Experimental Interest Management Architecture for DCEE. *Interservice/Industry Training, Simulation, and Education Conference*. Orlando, Florida

[18] Macannuco, D., Dufault, B., & Ingraham, L. (1998). An Agile FOM Framework. *Simulation Interoperability Workshop*, Orlando, Florida.

[19] Wilbert, D., Macannuco, D. & Civinskas, W. (1999). A Tool for Configuration FOM Agility. *Simulation Interoperability Workshop*. 99F-SIW-116.

[20] Hammond, J., Dey, M., Scrudder, R., & Merkin, F., (1998). Populating the HLA Object Model Data Dictionary—A Bottom-Up Approach. *Simulation*

Interoperability Workshop. 98S-SIW-075, Orlando, Florida.

[21] Rathnam, T., & Paredis, C.J.J. (2004) Developing Federation Object Models Using Ontologies. in the *Proceedings of the 2004 Winter Simulation Conference*. Washington, D.C.

Author Biographies

GENE WAGENBRETH is a Systems Analyst for Parallel Processing at the Information Sciences Institute at the University of Southern California, doing research in the Computational Sciences Division. Prior positions have included Vice President and Chief Architect of Applied Parallel Research and Lead Programmer of Pacific Sierra Research, where he specialized in tools for distributed and shared memory parallelization. He has also been active in benchmarking, optimization and porting of software. He received a BS from the University of Illinois.

DAN M. DAVIS is the Director, JESPP Project, Information Sciences Institute (ISI), University of Southern California, and has been active in large-scale distributed simulations for the DoD through two decades. As Assistant Director of the Center for Advanced Computing Research at Caltech, he managed Synthetic Forces Express. Earlier, he was a Software Engineer at JPL and Martin Marietta. An active duty Marine Cryptologist, he retired as a Commander, USNR. He received a B.A. and a J.D., both from the University of Colorado in Boulder.

ROBERT F. LUCAS is the Director of the Computational Sciences Division of the University of Southern California's Information Sciences Institute (ISI) and a Research Associate Professor in the Department of Computer Science at the same University. There he manages research in computer architecture, VLSI, compilers and other software tools.

[22] Brooks, F.P. (1995). *The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition* (2nd Edition), Addison-Wesley Professional, Reading, Massachusetts

He has been the principal investigator on the JESPP project since its inception in 2002. Prior to joining ISI, he was the Head of the HPC Research at NERSC, LBNL, the Deputy Director of DARPA's ITO, and a researcher at IDA. Dr. Lucas received his BS, MS, and PhD degrees in Electrical Engineering from Stanford University.

KE-THIA YAO is a Research Scientist and Project Leader in the University of Southern California Information Sciences Institute, where he leads a section of the JESPP project. That project has the goal of supporting very large-scale distributed military simulation involving millions of entities. He is a widely published author and lecturer on data management and teaches an undergraduate course on the topic at USC. Within the JESPP project he is developing a distributed data management system, the Scalable Data Grid, and a suite of monitoring/logging/analysis tools to help users better understand the computational and behavioral properties of large-scale simulations. 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.

CRAIG E. WARD is a Parallel Computer Systems Analyst at the Information Sciences Institute. Much of his recent research has focused on large-scale data management. Previously he has worked as a systems analysts for law enforcement organizations. He has a B.A. in History from the University of California, Irvine, and an M.S. in Computer Science from Loyola Marymount University.