High Performance Computing Facilities for Joint Military Simulation Data Management

Dan M. Davis
Information Sciences Institute, USC
Marina del Rey, California
ddavis@isi.edu

Garth D. Baer
George Washington University
Washington, District of Columbia
garthbaer@gmail.com

ABSTRACT

The overwhelming amount of output data inundating many in the simulation user community is a widespread problem. Much of this torrent is generated by current high-end computational capabilities. Joint and combined forces analysts are faced with the major tasks of first validating and then utilizing the data generated by modern techniques. A major part of the solution is an optimized data management software architecture. To enable the analysts to achieve success commensurate with the users' goals, a dedicated and appropriately designed data management facility was required. Taking cognizance of the advances made in the physical sciences' community, such a facility was conceived, designed and is being proposed to the HPCMP. The techniques of identifying, quantifying and implementing important data-handling parameters should be applicable to many large data-set problems in the Test and Evaluation community.

This paper will discuss the general state-of-the-art in data management, the specific problems presented by the U.S. Joint Forces Command simulations of up to a million independent SAF entities on a global-scale terrain, the methods used defining the problems presented thereby, and the path to the decision to standup a new facility. Adopting the successful techniques found effective in basic science, *e.g.* studying approaches used by other scientific research efforts, effective data management schemes have been discovered. Both the design process and the architecture itself will be laid out. Some issues addressed will be the choice of compute platform, the provision of associated communications, the selection of storage peripherals, the analysis of incipient technical advances that are likely to be germane, cost-benefit analyses of competing installations and the approach necessary in order to design for the future. Specific performance, cost and operational issues will be presented and analyzed. Lessons learned from this evolution should be extensible into many fields associated with modeling and simulation, as well as the T&E community in general.

ABOUT THE AUTHORS

Dan M. Davis is the Director, JESPP Project, Information Sciences Institute, University of Southern California, and has been active for more than a decade in large-scale distributed simulations for the DoD. While he was the Assistant Director, Center for Advanced Computing Research, California Institute of Technology, he managed Synthetic Forces Express. He has also served as a Director at the Maui High Performance Computing Center. He served in the USMC on active duty and is a Commander, Cryptologic Specialty, U.S.N.R.-Ret. He has been the Chairman of the Coalition of Academic Supercomputing Centers and received a B.A. and a J.D., University of Colorado, Boulder.

Garth D. Baer is a technical analyst who is studying the impact of policy on technology as well as the changes technology makes on policy formulation and implementation. He currently is a graduate student at The George Washington University, studying policy formation at the Federal level. Previously a Principle Support Engineer at Oracle Software Corporation, he developed new web-based applications and troubleshoots production database issues. Earlier, he was a Mission Control Engineer for Milstar Communications Satellites at Lockheed-Martin. He participated in a multi-university group developing a vision statement for DoD policy on M&S. He received Bachelors in Physics, Univ. of Colorado, Boulder and a Masters in Technology Management, Colorado Technical Univ.

High Performance Computing Facilities for Joint Military Simulation Data Management

Dan M. Davis
Information Sciences Institute, USC
Marina del Rey, California
ddavis@isi.edu

Garth D. Baer
George Washington University
Washington, District of Columbia
garthbaer@gmail.com

BACKGROUND

Since before the advent of written history, military commanders have sought ways to prepare their fighters for upcoming battles. Some of this involved working out plans of attack and some involved assessing the various levels of capability. Though today's commanders use different tools, their goals would be recognized by the commanders of yore. Both would like to observe their fighters in something akin to real combat and make judgments based on those observations. But the advent of industrial power has added to their burdens and our current commanders have had their job skills extended into something more akin to logisticians than their ancient counterparts. The terrible swiftness of modern swords allows little room for contemplation and adjustment.

To help ameliorate this problem, the US Joint Forces Command (JFCOM) has been designated as the transformation laboratory of the US Armed Forces. They have adopted the well-tested, yet powerful and easily modified, Semi-Automatic Forces (SAF) family as one of their major simulation platforms. The version they principally use is Joint SAF, or JSAF. One of the research objectives of their Joint Experimentation Directorate (J9) is to assess the capabilities of various systems when deployed in an urban setting, they thus must "field" an urban population in experiments such as Urban Resolve (UR). This is a complex problem, calling into use all of the capable skills of the simulation programmers, system designers, experiment operators, and data analysts.

Typically, a single IA-32 based PC (like Intel Pentiums) running on a Linux operating system can support a few thousand civilian clutter entities and lashing 20 to 30 of these computers on a Ethernet Local Area Network (LAN) can support a population of around 30K (Ceranowicz, 2002). Looking at any major urban center in the world, one would see that each has nearly two orders of magnitude more entities in action than that, *e.g.* Baghdad has a population of 7.8M with vehicles adding at least another million. In the vernacular of the computational scientists, the LAN solution does not scale to this level. New hardware and software is needed. The JFCOM Joint Experimentation on Scalable Parallel Processors (JESPP) Project was initiated to respond to this problem (Lucas, 2003). Utilizing the readily available capabilities of the High Performance Computing Modernization Program, (HPCMP) the JESPP team was successful in achieving the needed scalability on Linux clusters, also using IA-32 architecture.

Thence came the deluge. Like others in the SAF community, JFCOM analysts now faced increasingly large data sets, complex sensor results, convoluted scenarios, and diverse environments. The need to re-run certain actions of interest also heightened the need for data collection, processing, management, storage and retrieval. Literally terabytes of information could be anticipated, and even that assumed much of the specific clutter background activity was not archived and therefore could not be exactly duplicated. This is true because the underlying SAF codes are not deterministic, relying on pseudo random numbers to determine a action's result. Though this deluge of data increases each simulation's reliability, managing it proved a difficult task.

The JFCOM leadership sought the assistance of the Information Sciences Institute of the University of Southern California (ISI/USC) and the Center for Advanced Computing Research of the California Institute of Technology (CACR/Caltech) to resolve these data problems. As reported elsewhere in this conference, (Bunn, 2005) their approach is based upon their experience with High Energy Physics sensor data.

In most high performance computing implementations, the data analysis is performed on platforms designed to facilitate the creation of the data or to interface easily with sensors and other data-producing devices. The JFCOM situation is somewhat different. They bear the pressures of several calls upon their services. These include:

- analysis of battlefields of the 2015 time-frame
- designs intended for use in the next year
- real-time assistance to the warfighter today

These will likely also obviate the use of their clusters for analysis.

That brings about a new opportunity for JFCOM and HPCMPO: not only using custom designed database software, but developing an optimized set of Linux cluster nodes to facilitate the analysts' task and ensure the validity and sanctity of the data. The rest of this paper is directed at establishing the needs of the users, surveying the potential platforms for this use, assessing the various performance characteristics and designing a stable and efficacious system.

THE PROBLEM

Data

The authors do not intend to burden the reader with a surfeit of arcane technical descriptions of the format of the data produced by JSAF and associated sensor federates (see Graebener, 2003 for more details.) That which follows is a higher-level and, the authors contend, more germane general description of the data available for collection and the steps necessary to render it useful to the analysts and experiment controllers. This should allow the reader to better assess similarities between their simulations and the simulation under study. This is intended to provide a basis for analyzing the relevancy of this work to theirs.

For simplicity's sake, we break down the JFCOM simulations into four broad areas:

- Terrain and Environment
- Civilian "clutter"
- Operational entities
- Intelligence sensors

Of these, only the first broad category, Terrain and Environment does not represent a major data task. 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, while the impact of clouds may be more random and 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 are reported to the system on the order of once every 10 to 100 milliseconds. The system itself can tolerate up to 500 millisecond latencies before showing strain. In any case, being able to simulate millions of entities, (Barrett, 2004) 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. The JESPP team has developed a powerful algorithm for saving data so generated (Wagenbreth, 2005) but the volume of data is so high that one very real solution is to simply discard the "clutter" data, treating it much like environmental data.

Next are Operational Entities. These 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 JFCOM military personnel can personally control US and Allied forces and a "Red" team can bring to the experiment all of the creativity and experience they have garnered, frequently after decades of experience in the service. This brings to the surface a significant difference in JFCOM data and, say Bank of America "transaction data" or Caltech "physics sensor data." The solutions sought in both the software and hardware areas must be sufficiently general to support any conceivable data type and load, yet sufficiently specific to optimize both areas for JSAF use. This is not a trivial conundrum.

Finally, there is the data to be gleaned from the intelligence sensors simulation programs. This issue is exacerbated by Intellectual Property issues, occasioned by the interest the programmers have in the intelligence sensor program, which is proprietary. The nature, amount and relevancy of this data is to some very real degree outside the control of the either the authors or the other managers engaged on this project.

The System

One of the great strengths of the JFCOM experimental design is its distributed and dispersed nature. The experiments themselves are housed and controlled by the JFCOM out of its experimental bay near Suffolk, Virginia. Environments and data are managed remotely out of Fort Belvoir in Northern Virginia. The civilian clutter are laid down and managed by a team a continent away in San Diego, at the SPAWAR center on Point Loma. The two 128 node, 256 processor Linux clusters that are provided by the HPCMP, are located in Maui at the Maui High Performance Computing Center (MHPCC) and at Wright Patterson Air Force Base at the Aeronautical Systems Center Major Shared Resource Center (ASC-MSRC) in Ohio.

Communications between the sites are provided by the Defense Research and Engineering Network (DREN). Experiments for Urban Resolve have been unclassified, but work for CENTCOM may require encryption of the communications' links. The Linux operating system is

common across the net (usually Fedora) and most of the programs are written in C++, with a smattering of Java. While the sites at Suffolk and San Diego are entirely devoted to J9, the two HPCMP sites are multi-tasked, including maintaining the dedicated J9 cluster, named *koa* and *glenn*.

Figure 1. Notional Multi Path WAN between, TEC JFCOM, ASC-MSRC, SPAWAR, and MHPCC

Note that there are geographical dispersion issues, Maui being on the order of five thousand miles from Suffolk, as indicated in the notional diagram in Figure 1. This precludes easily and economically meeting with the entire staff. Further, with a five time zone span, the operational synchronization is difficult, most especially in the summer when the mainland sites go to daylight savings time, while Hawai'i does not, thereby creating a six-hour difference.

The Users

The users are nearly as diverse as the system is dispersed. The description that follows will lay down some of these differences. All of these users are potential generators and users of the data to be managed:

- System administrators
- Simulation operators
- Friendly and Red Team controllers
- Analysts

The system administrators must effective manage the distributed hardware and voluminous software necessary to maintain the simulation. As load balancing and fault tolerance are not automatic, they must have real-time access to data during the experiments. After the experiment, they must have unfettered access to the stored data to make assessments of necessary upgrades and remedial repairs. One consistently taxing area for the last decade has been the issue of configuration control across as many as a dozen supercomputers, *e.g.* ensuring the compiler version is in synch.

The simulation operators themselves are tasked with delivering the requisite Forces Modeling and Simulation (FMS) capability to the experiment managers and scenario designers. For this, they need to be able to understand what areas of the simulation are functioning and which areas are producing anomalous data. In addition to this real-time data, they need after-action review to allow reprogramming JSAF's and other federates' code to either correct errors or to enhance existing or add new, capabilities.

The next group, the controllers, need to have carefully limited access to data. They should be given only that data which serves the goal of providing them that which they would have on a real battlefield. Most often, this information will be presented to the controllers in the form of appropriately formatted "messages," emulating real communications.

Finally, we have the analysts, for whom most of this work has been done in the first place. These analysts have varying needs, varying time requirements and varying local compute capabilities. During the experiment, they want to have real-time access to all of the data in the experiment. They may want to discover the exact location, orientation and state of any entity being simulated. They may want to be presented with the output of fairly sophisticated data queries, (Graebener, 2004). They may want to view the real-time Plan View Display (Map) (Figure 2) or Stealth (3-D) to see how the experiment is going. The analysts may order the entire action to be repeated, from the beginning or from some critical point.

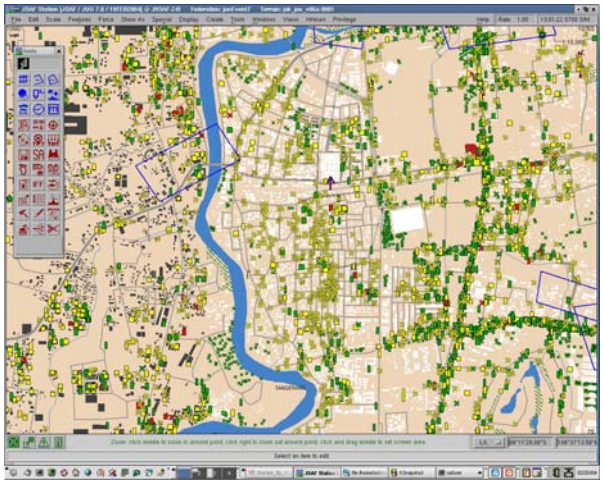

Figure 2. Plan View Display from JSAF

This, however, is only the first stage of the analysts' needs and goals. Every night, during operations stand-downs, the analyst may be reviewing the previous days results with a eye toward altering the experiment or re-running important parts.

After the action is done, the analysts may spend weeks reviewing the data. They will use SQL commands to elicit as much information as possible from the data collected. Again, they may want to repopulate the scenario and run some parts again. They may find entirely new areas of interest or study. They may want to compare this experiment with ones run a year or more before. They may also wish to embark upon extensive and unconstrained data mining, by its very definition, a process that has no preconceived goal, therefor no clear view of the type, nature or extent of the data that will yield up new and vital insights. (Davis, 2004)

Implementation

The JESPP team has developed code that is capable of intercepting, decoding, archiving and organizing all of the data generated (Yao, 2005), and that system is currently under the process of being parallelized to make it more scalable. With the user set distributed across the country (Virginia to Hawai'i), the system is being developed to respond equally to many users and many analysts accessing the data simultaneously.

The system is designed with MySQL as the database engine and Linux as an operating system. Both are open source programs, commonly available and well documented and supported. The use of one of the popular Unix operating systems ensures easy portability to other high performance computing platforms, such as Cray, IBM, SGI, HP and others.

While design decisions have not been finalized, it seems likely that some of the MeshRouter communications structure, so successful on the simulations itself, will bring fault tolerance, scalability, and supportability to the distributed user-base. Another open issue along this line is whether to have the major data facility located at one of the existing sites, a new site or distributed amongst all of the current sites. (Gottschalk, 2005)

OPTIONS

High Level Architecture

In looking at these issues, one of the first considerations is that of the general system architecture being considered. One possibility would be to have the entire system designed for the single processor PC used by the analyst. Putting this burden on the analysts' PCs would severely constrain the potential value of the simulation.

Another option would be to utilize some of the nodes of the distributed compute (DC) facility at MHPCC and ASC-MSRC. As discussed before, this would limit operational capabilities and make the administration of *koa* and *glenn* even more difficult. It would, however, have the benefit of being readily available and, for that reason, it may be the interim solution.

The final concept is the establishment of a new analytical computational facility and the development of a scalable program. This would allow the system designers to parallelize, not only separate and independent inquiries, but also to distribute the difficult and time-consuming functions of the data-collection and data-analysis utilities. The location of such a facility is in question. As the issues involved here may be comparable to the reader's??, a modest effort will be made to lay them out. Because some of the data may be classified, at least one classified facility is indicated. Due to the fact that the users may actually come from anywhere in the world, having mirror sites that are widely dispersed may be useful. On the other hand, there arguably is a significant benefit to having the facility under one roof and within easy access of the JFCOM staff.

Compute Platform

Recent developments in CPU architecture may impact the choice of the compute platform. Both AMD and Intel have fielded CPUs capable of 64 bit memory addressing, the Opteron and Pentium 64, respectively. As an aside, it should be noted the previous generation Athlons and Pentiums used 64 bit registers, but 32 bit memory addressing. Both companies still produce pure 32 bit implementations, as well as hybrid implementations, the Hyper-Threaded (HT) configuration. While the 64 bit addressing in large database implementations may seem attractive, current offerings do not allow the user to take much advantage of the larger address space.

Looking at these solutions *seriatim*, the vast majority of FMS compute platforms today are based on the advanced design chip designs of the IA-32 architecture. Some of these, such as the Xeon chips from IBM, are capable of breaking the "4GB" memory addressing

space of typical IA-32 by a margin (up to 64 GB) that is not reflected in available board support, usually restricted by the limited number of RAM slots and board chipsets to 8 GB total. The 64 bit Itaniums (often regarded as an architecture that will not last,) Opteron, Xeons, and Athlon 64s are on the market and available, but again, in ISI's experience, appear on boards limited to 8GB. Memory bus speeds are up and memory prices are down, allowing economical configurations on the PC model that are then distributed across nodes on Linux clusters.

There are other architectures in the offering. The DARPA HPCS program is currently pursuing three advanced designs that will feature shared memory in the PetaScale machines. (Graham, 2004) To adequately use these, there are programs underway to prepare the systems and programming software to support the new architectures. (Kepner, 2003) Not only will these advances reduce latencies and increase bandwidth in data storage, processing and retrieval, the new architectures may revolutionize the code development paradigm now in vogue. The parallel programmer, faced with distributed processing and with distributed memory has long since had to develop very convoluted and sophisticated designs to enable programming for systems that are often distributed across thousands of compute nodes, in dozens of machines, located all across the United States and spanning half a dozen time zones. (Brunnett, 1998.) Many have argued that these advanced architectures would, to some degree, obviate these tortuous paths to useable code.

Database Software

As this article focuses on database installation choices and analysis, only a few major parameters will be discussed here, vis-à-vis the selection of the database software. The first high-level choice is the option to go with one of the major database companies, with their elaborate contractual obligations and cost, but with very professional support staffs and considerable high quality documentation and training. Some of these commercial packages are quite powerful and capable, while others are clearly intended for home use only and have neither the power or the scope to handle serious database burdens, as are found in FMS. The authors have had the experience of working on projects that erroneously began using these "lower-power" databases, only to quickly have to abandon them in favor of something more capable. This, of course, led to much lost training and programming time.

The second possibility is the utilization of the plethora of really substantial and fully capable open source and public license software. Two of these are MySQL and SQL Lite. As with much of the software associated with the Linux revolution, there is an active and vocal (at least electronically) community, supporting and discussing these choices and their uses. Programmer support is easily obtained and new users have, within the author's experience, developed proficiency and sophisticated approaches rapidly.

Quantification of Platform Performance

While it is unlikely to evoke enthusiastic responses and effusive thanks, the prudent user will be well advised to set aside enough time and personnel to rigorously do early testing to ascertain the acceptability of the performance of the target platform. Following the lead of Paul Messina of Caltech, (Messina,1990) the authors accept the tenet that performance testing is always most likely to provide reliable data if the benchmark used is not an artificially generated "toy " program. The most successful performance testing will be the use of the software to be used, processing the data to be studied, in operational conditions as close to real as possible.

Table 1. Computers Evaluated

Computer	Athlon1800	Xeon	Opteron64
CPU Spd.	1.5GHz	3.02 GHz	2.4 GHz
CPUs/Node	2	2	2
L2 Cache	128	256	256
RAM	2 GB	2 GB	2 GB
Bus Sp	100	133	133
HDD	60 GB	60 GB	240 GB
RPM	7200	7200	7200

For assessing the proposed facility for JFCOM use, the authors had normalization runs conducted on three basic platforms, an Intel Pentium III without multi-threading, a dual processor Intel Xeon with multi-threading and an dual Opteron 64, also exhibiting multi-threading. Other parameters and specifications are given in Table 1. While the runs represented here do not seem adequate to the authors, they are indicative of the types of testing that would be appropriate.

The test scenarios were of a standard analysis of the processing of data from a JSAF run, accomplished for JFCOM. It deals with location, orientation, status and movement of "clutter", that is civilian or non-military pedestrians and vehicles (Ceronowicz, 2002). Varying conditions were input into the data processing evolution and the following results were obtained. The data management program makes several decisions

about whether to convert (decode) the raw data and store it as more easily retrieved ASCII text, to throw it away or to archive it without decoding. A common position is to discard any data for which insufficient processing power is available. This Hobson's choice is unacceptable in the eyes of the authors as presently discarded data may hold key information yet unbeknown to the user (Davis, 2004).

The chart in Figure 3 shows the number of Decoder Delete events that were processed in 15 minutes.

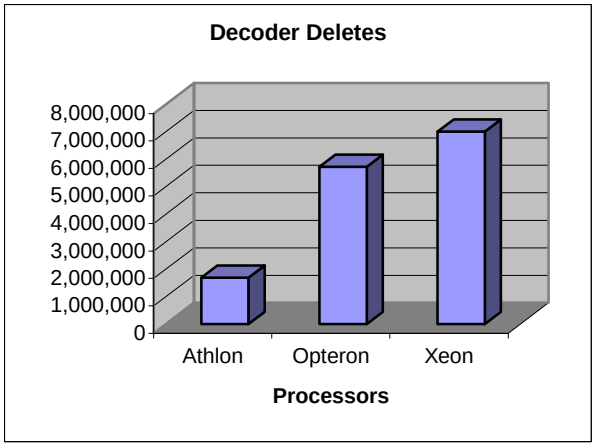

Figure 3. Decoder Deletes – 15 Minutes

This activity consisted of scanning through populated tables looking for the oldest data and leaving everything else in the table. As can be seen the older and slower Athlons are significantly out-performed by the newer, 64 bit oriented processors. However, it should be noted that the Athlon dual boards differed in other ways as well. Perhaps the major insight from this slide is that the H/W configurations do make a marked difference in performance and even simple initial tests can begin to show performance break points.

Next, attention is turned to the truncating of the tables. Often, during JFCOM simulations, when the data tables became full or when data was no longer needed, the truncate command was issued, thereby deleting all data in the table. This tasked another processor consumed with a housekeeping task of consequence in this particular analytical process. Figure 4 shows the truncating results, totaled, for a ten minute run.

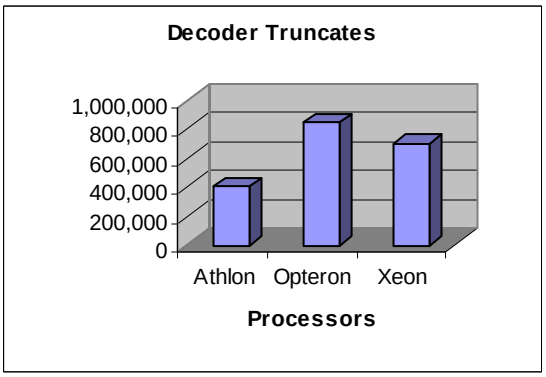

Figure 4. Decoder Truncates – 10 Minutes

Side by side comparisons for any smaller units of time suffered due to the inconsistent rate of the activity over time. One graph is presented here to show the "spikey" quality of the data processing loads. Figure 5 is a graph of a single run of the Athlon machine doing decoder truncates. It demonstrates the necessity of accumulating performance over several minutes time.

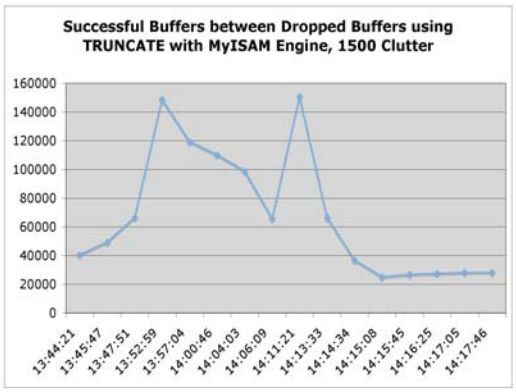

Figure 5. "Spikes" of Athlon doing Truncates

The fact that the application used here as an exemplar of the approach did or did not produce dramatically different performance levels should not be taken as a de-motivating factor in the desirability of performing such tests. This result is supported by a survey of database users who clearly favored software advances. Often, dramatic differences have been observed and it is critical to avoid a trap of procuring millions of dollars in Linux cluster hardware only to find the 64 bit option was either an additional expense with no benefit or, on the other hand, an expense that would have paid for itself in the first few months of operation. Again, the main message here is that benchmarking should be done with the target application, not a general benchmark tool.

DISTRIBUTED DATA PROCESSING

The next topic to discuss is the design characteristics of the JFCOM usage that militated in favor of a distributed data facility. (Yao, 2005) A quick review of the JFCOM operational paradigm may be well advised here. JFCOM runs large scale simulations that are supported by and of interest to personnel at facilities that are literally world-wide, but are typically distributed from the Peninsula of Virginia to the slopes of Haleakela in Maui. This use is dynamic in both its load and its subject matter. During simulations, data is produced at vastly different rates and data is accessed and analyzed in similarly unpredictable ways. Original, non-scalable, designs for data handling were structured around discarding data of too great a volume to be managed, then returning data assumed to be of interest to a data facility at JFCOM in Suffolk and processing it there. This virtually precluded data analysis and use during the operations period of the simulation experiment. Figure 6 below is a notional representation of the operations phase of the experiment, with the data facility's (SABER's) grey tint indicating its idle state.

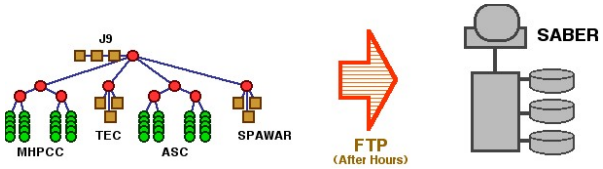

Figure 6. Current JFCOM Data Design - Ops

The converse becomes true during the data management cycle of the experiment, typically during the night and on weekends and then the entire period of analysis between the experiments. As seen in Figure 7 below, now the operations platforms are idle (at least as far as JFCOM is concerned) and the data facility is fully utilized.

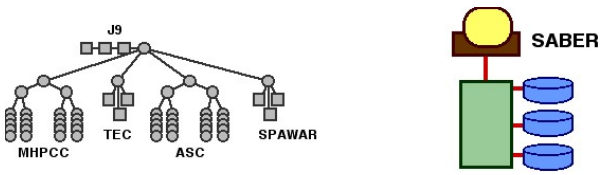

Figure 7. Current JFCOM Data Design - Analysis

Some other weaknesses of this system may be obvious, but merit attention. Firstly, a single data location is not fault tolerant for the diverse and dispersed user community, as equipment failures anywhere between the user analyst and the data results in total downtime. Secondly, this concentrated asset does not lend itself to scaling, as intervening tree architectures put huge loads on the "root" node, making it a likely (and observed) choke point (Barrett, 2004). Thirdly, the current facility processes everything serially at that point, not taking advantage of the benefits afforded by parallel processing Linux Clusters.

One of the design characteristics being studied at this time is the possibility and desirability of using:

- dispersed Linux clusters
- partitions of the operational cluster
- second processors on cluster nodes
- other variants

These would allow assessment of the optimal dispersion of the data and data handling.

The concept of using dispersed supercomputers is often called met-computing, a close relative of the currently popular term grid computing (Foster, 1997). The meta-computing configurations most favored by the JFCOM team is one in which the major compute sites, Maui and Ohio, would be likely locations for much of the data logging, decoding, storing, organizing, and archiving, thereby minimizing wide area network (WAN) communications. This could be significantly augmented by a separate analytical Linux cluster at either a new location, at one of the compute sites or at JFCOM in Virginia. Figure 8 is a notional view of the operations phase of such a design. Note that the user (or any number of users) can access the data during the simulation experiment. Using the scalable communications hierarchy developed for operational scalability, (Gottschalk, 2005), the number of users accessing the three data facilities should not be impeded by communications bottlenecks.

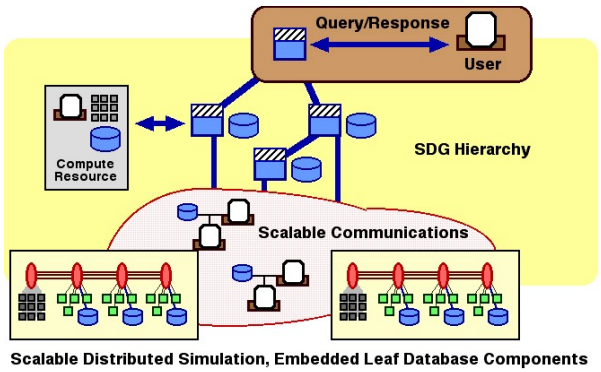

Figure 8. Planned JFCOM Data Design – Ops

Further, as additional Linux cluster nodes have been set aside or new cluster nodes provided at each site, the data logging, decoding and entry into the MySQL database will occur real time, allowing for near real time accesses by analysts. This is, of course, a huge improvement, allowing the analyst to retrieve and internalize insights from on-going operations with an eye toward real time changes in the experiments operation.

As in real live-fire actions, the amount of information available to the soldier and commander alike is growing far beyond their ability to make optimal use of it. This is also reflected in the simulation environment. Originally, subject matter experts reviewed, discussed and opined about the validity of and insights from the simulations. Now, the data is just too vast and the action too complex to make best use of this type of analysis.

Both SME and the analysts have a need for all the machine processing assistance that can be provided to them. This, again, will help model future assistance to the warfighter in combat. It should be remembered that machine processing and analysis are important when more than a million entities may be simulated across terrain databases that are literally global, but may also be engaged in high fidelity urban simulations, such as the snap shot in Figure 9.

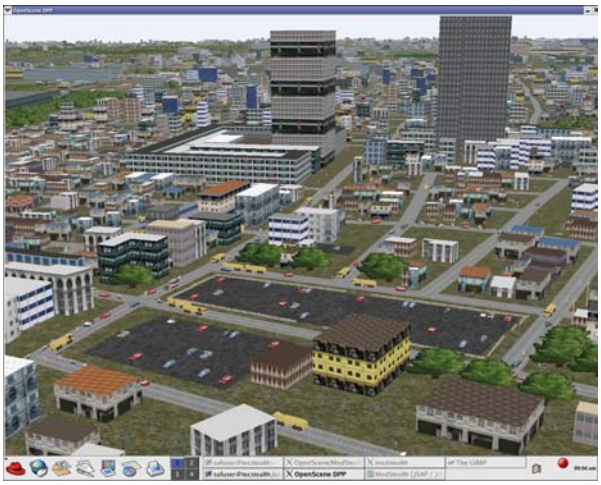

Figure 9. JFCOM Experiment – One Block of City

The last configuration diagram shows the usage during the analytical phase of the work. As can be seen in Figure 10, there is a significant amount of activity, with only the simulation operational nodes being idle. In actuality, these nodes are turned back over to the HPCMP community for other batch or interactive users. This is represented here in by the absence of the nodes in the two lower boxes, which represent the compute sites. Of course, as analytical capabilities are implemented, even these "operational" nodes may be pressed into service.

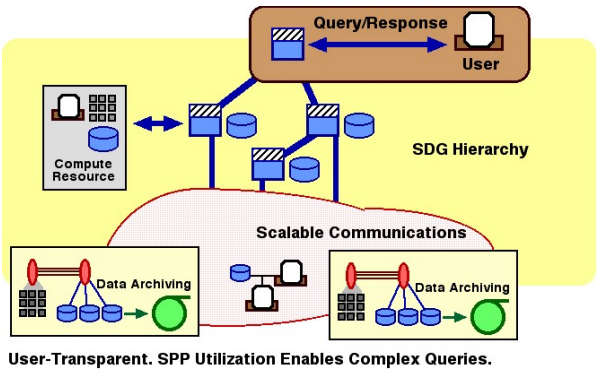

Figure 10. Planned JFCOM Data Design –- Analysis

It is not suggested that the efficacy of this design for JFCOM makes this design appropriate for all situations, but the process described above will serve the simulation professional well. Careful planning begins with establishing and maintaining good communications with all of the experimenters and all of the potential users. Repeated "white board" planning sessions and pilot trials are very useful in determining the correct configuration. Fortunately, one of the real benefits of the Linux cluster revolution is that its component parts are nearly universally useful, so even design decisions that later prove less than optimal can be recovered, due to the ease with which the cluster nodes can be put to alternative, beneficial uses.

CONCLUSIONS

The authors started from what they consider to be a virtually unassailable set of premises:

- Modeling and simulation can utilize the benefits from high performance computing
- This benefit produces so much data that an unaided user will find it difficult to extract, process and analyze the results
- Users and computing facilities are often dispersed geographically
- Failure to recover all of the possible insights will likely be an undesirable result for the analytical teams and the warfighters

The authors' beginning thesis could be stated as follows: the curse of the surfeit of data can best be met by the very creator of that curse, *i.e.* scalable, dispersed, parallel meta-computing. Pursuant to that thesis, the JESPP team has created and initially tested a distributed data-handling platform that is both scalable and responsive.

Part of that design process is the scoping, evaluating and implementing the facilities themselves. While ideally these issues are completely outside the ken of

the user, they are of significant concern to the simulation team and the computational scientists supporting them.

Lessons Learned

The first lesson learned is the desirability of seeking advice, counsel and support from as diverse a group as possible. Resisting the temptation to do it *de novo*, it has proven to be very useful to seek as broad an input as could be arranged. The benefits of the University community came to the fore here, as major research Universities have on their staffs experts in every field.

A second lesson, flowing from the first, was that the experience and insights from the High Energy Physics community were more germane than were the commercial transaction-processing programs or the Internet recreational data-search designs. The reason for this is assumed to be the closer relation to the types of data, the technical literacy of the users, the more uniform access to high-bandwidth, and the lack of the need for elaborate inter-user security, but very high external security. However, there are differences from the HEP community, *e.g*. the simulation community's broader and more dynamic analytical interest. The HEP community tends to work with output from a single, well-defined experiment, looking for a specific set of data. The military Forces Modeling and Simulation community has many user interests and changing goals.

The third lesson is that working with open-source, public licensed software has many advantages for the developer. The Linux community is active and involved. Source code is available for scrutiny, modification, and implementation. Hundreds of hours that would be expended in the procurement process are now available for more productive endeavors. While the authors recognize the value added by major commercial vendors and their support staffs, the JFCOM experience indicates that open source software should be seriously considered.

The last lesson learned to be offered here is the desirability of simultaneously pursuing both a "bottoms-up" and "top-down" approach. The exigencies of operational necessity dictate that current but evolving code bases, be pursued and the authors' experience indicates that these efforts often outstrip and outperform new "improved" designs. This bottoms-up approach keeps the simulation going and serves to be a real time laboratory to continuously assess the applicability of new concepts. The top-down approach is useful in supplying a more theoretically founded view of the ultimate design. This approach keeps the design from growing without focus, necessary infrastructure or control. This is also very important in insuring scalability of the final product. Pending arrival of new computer architectures, parallel processing offers the best hope of increasing compute power. If these implementations do not scale, however, all of the additional processors will be of little use.

In closing, it should be restated that High Performance Computing brings FMS new capabilities, which bring new floods of information, for which HPC facilities can be designed. The combination of careful planning and openness to others skills are a *sin qua non* of success.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the members of the ISI JESPP team who have contributed to this paper through their efforts and their intellectual stimulation. Much of the success reported here came from the Joint Experimentation on Scalable Parallel Processor project, initiated, directed and funded by the Joint Forces Command and to a very large degree conducted on the compute assets of the Maui High performance Computing Center, ASC-MSRC at Wright Patterson Air Force Base and other members of the High Performance Computing Modernization Program. Dr. Thomas D. Gottschalk and Dr. Ke-Thia Yao were very generous in sharing their designs for distributed data management. Craig Ward of ISI generated the performance characterization data. Without the support and encouragement from all of the above, none of this would have been possible. This material is based on research sponsored by the Air Force Research Laboratory under agreements 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

REFERENCES

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

Brunett, S., Davis, D., Gottschalk, T., & Messina, P. (1998) Implementing Distributed Synthetic Forces Simulations in Metacomputing Environments, *The Seventh Heterogeneous Computing Workshop*, Orlando, FL.

Brunn, J. & Gottschalk, T. (2005, publication pending), Incorporating High Energy Physics Data Capabilities into Joint Forces Simulations . *Proceedings of the Interservice / Industry Training, Simulation and Education Conference*, Orlando, FL.

Ceranowicz, A., Torpey, M., Hellfinstine, W., Evans, J. & Hines, J. (2002) Reflections on Building the Joint Experimental Federation. *Proceedings of the Interservice/Industry Training, Simulation and Education Conference*, Orlando, FL.

Foster, I. & Kesselman C. (1997) Globus: A Metacomputing Infrastructure Toolkit. *Intl J. Supercomputer Applications*, 11(2), 115 –128.

Gottschalk, T. & Amburn, P., (2005, publication pending), Extending The MeshRouter Framework for Distributed Simulations. *Proceedings of the Interservice / Industry Training, Simulation and Education Conference*, Orlando, FL.

Graham, S., Snir, M., and Patterson, C., Editors, (2004) *Getting up to Speed, the Future of Supercomputing*, The National Academy Press, Washington, D.C.

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, FL.

Graebener, R. & Rafuse, G., Miller, R., & Yao, K-T., (2004) The Road to Successful Joint Experimentation Starts at the Data Collection Trail, Part II. *Proceedings of the Interservice / Industry Training, Simulation and Education Conference*, Orlando, FL.

Kepner, J., (2003), HPC Productivity: An Overarching View, *International Journal of High Performance Computing Applications*, London, UK

Lucas, R. & Davis, D. (2004) Joint Experimentation in Scalable Parallel Processors, *Proceedings of the Interservice / Industry Training, Simulation and Education Conference*, Orlando, FL.

Messina, P., Baillie, C. F., Felten, E. W., Hipes, P. G., Walker, D. W., Williams, R. D., et al, (1990), Benchmarking Advanced Architecture Computers, *Concurrency*, 2, 195.

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

Wagenbreth, G., Davis, D., Gottschalk, T., Lucas, R. & Yao, K-T. (2005, publication pending), Operational Experience, Distributed Simulations, Data Management and Analysis , *Proceedings of the 2005 Winter Simulation Conference*, Orlando, FL.