XMSF Workshop

Monterey, CA

Position Paper

Kalyan S. Perumalla, Ph.D.
College of Computing, Georgia Tech
http://www.cc.gatech.edu/~kalyan

August 19, 2002

Introduction

The following are some of the authors' areas of expertise that bear relevance to the XMSF:

- Analytic simulation domains: Telecommunication network simulation, Transportation simulation, Air traffic control simulation
- High-performance Runtime Infrastructure (RTI) implementations: Time management algorithms, Data distribution management, Execution on supercomputers, workstations, PCs, PDAs
- On-line simulation applications: Fixed network and mobile ad-hoc network optimization.

In the rest of the document, we ruminate about the relevance and impact of the XMSF approach on the preceding areas.

Analytic simulation

Currently, there is a great need in communities, such as telecommunication network simulation and transportation systems simulation, to be able to publish/pull models to/off the Web and assemble and simulate a virtual configuration of choice by the end-user. For example, in Internet simulations, users would like to be able to search for the most efficient/relevant traffic source models, network protocol models and router models for their specific network studies. After discovering and gathering all the relevant components, the models should be assembled and simulated. Unfortunately, no such Web-access facility exists presently, and the Web-oriented nature of XMSF is appealing in such simulation applications.

Moreover, the Web-based access portion of the XMSF approach appears to be well suited towards a flexible and automated execution of *multiple* simulation runs in an experiment series, with the following features[1, 2]:

- Support for session management and control for *replicated* experiments – going beyond simple start/stop control, possibly supporting cloning, and backward/reverse execution.
- Support for integration with optimizers or other higher-level design applications.

With such support, XMSF will fill the void in integrated execution of federations in the overall design/optimization loop of simulation usage.

Time Management

The XMSF style of software organization and execution present research challenges with respect to efficient time management (TM), due the following factors:

  - Wide network link latency spectrum
  - Highly dynamic joins and leaves (transient messages problematic)
  - "Always on" nature of presence and access.

These factors together may warrant the development of new TM services (e.g. introduction of the concept of *time-to-live* on time-stamped events) that complement the existing HLA RTI TM services[3].

It also raises the issue of what is the best way to map implementation of XMSF services to web technology – variety of CGI-bin or ActiveX realizations, etc. – that is most efficient with respect to TM.

Online Simulation

Lately, a new usage pattern of simulations is emerging, namely, on-line simulation. In this approach, simulation is used in a closed-loop fashion with the real system that is being simulated. The current state of the real system is gathered and fed into a controller/optimizer. The controller/optimizer then launches multiple simulation runs corresponding to potential future scenarios and/or events, and either proactively steers the real system with the simulated knowledge, or quickly reacts to future system states based on the pre-simulated information.

This on-line simulation pattern is appearing in many places, including dynamic optimization of networks, and intelligent model-based control of transportation systems. On-line simulation of telecommunication networks is described later in related work.

Existing standards such as the HLA and DIS are primarily concerned with inter-simulator interoperability, and do not directly provide standardized support for multi-simulation frameworks. The XMSF can fill in this need, especially due its inherently distributed, client-server nature.

Perhaps, a *simulation-server* paradigm could be built into the XMSF set of services from the start, and these services can be utilized to achieve new simulation usage patterns such as on-line simulation.

Related Work

Here, we describe some of our past and current work that is related to the goals and approach of the XMSF.

Jane

It is clear that Web-based, and Internet-based approaches to simulation control generally have been found quite useful and successful (e.g. [4]). First hand, we have been having very favorable experiences with remote execution and control facilities using a system we developed called Jane[5].

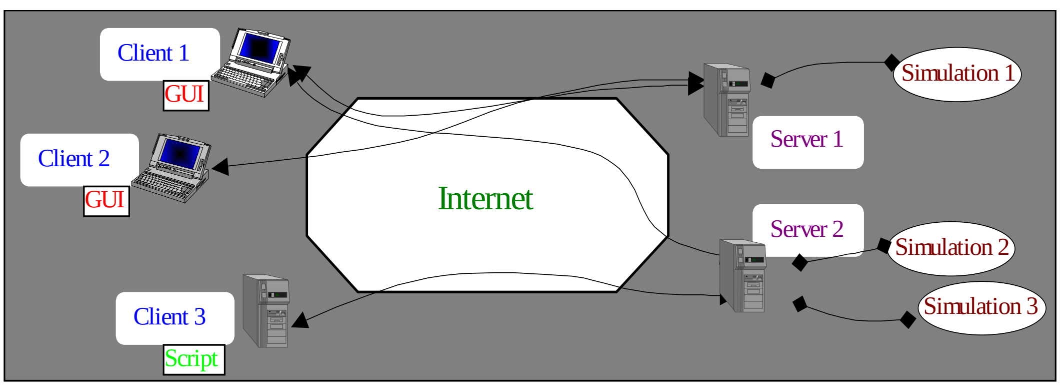

Figure 1: Functional architecture of the Jane simulation control system. One or more clients can control one or more simulations, with arbitrary overlap. In this example, clients 1 and 2 are controlling simulation 1, while client 3 is controlling simulatoins 2 and 3. Clients can behuman users or programmed automata.

Jane is based on a client-server model, as illustrated in Figure 1. Client implementations exist in Java, and server implementations are in C. The Jane system has served an extremely useful role of remote simulation execution, primarily focusing on multiple as-fast-as-possible simulations.

FDK High-Performance Runtime Infrastructure

We have expended considerable effort in high-performance implementation of simulation middleware such as the HLA RTI. Traditional approaches to high-performance time management in RTI implementations assumed relatively static configuration of federates, with few dynamic changes[6-8]. These approaches resulted in highly efficient implementations, at the cost of certain amount of usability from the user's point of view. On the other hand, the XMSF, due to its highly dynamic, distributed and fault-tolerant nature, can greatly enhance the usability, but at some potentially increase runtime cost. This tradeoff opens up challenging issues with respect to efficient realization of simulation services such as time management and data distribution management.

Online Network Simulation

In some of our recent on-going projects, we are successfully applied the on-line simulation usage pattern in the context of defense network optimization[9].

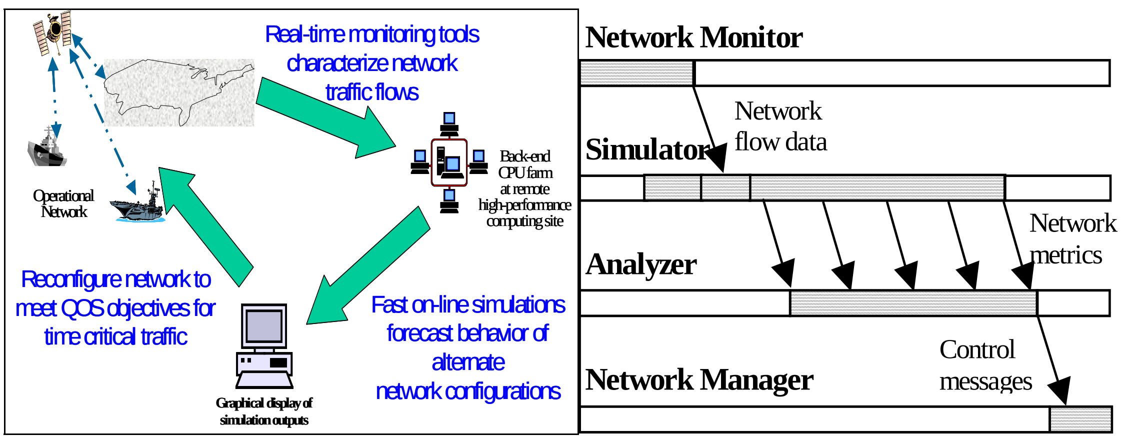

Figure 2: Functional and logical views of the on-line network management loop.

Wide-Area Time Management

The problem of efficient time management over wide area links is a challenging one, because of high latencies involved. The levels of latency and jitter typically experienced in Web accesses exacerbate the problem of accounting for transient messages in time synchronization. A few algorithms, such as synchronization over unreliable network transport[10], have been designed to alleviate some of the associated problems. However, additional research is needed to take this to the next level of complexity that is envisioned for the XMSF.

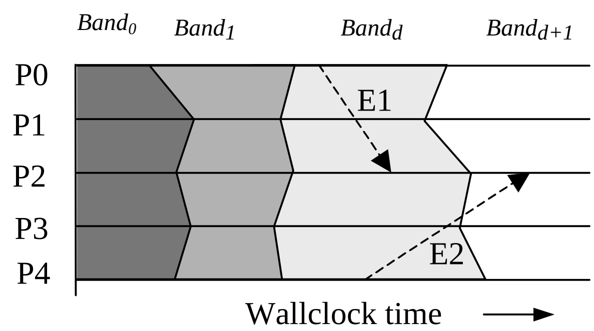

Figure 3: Illustration of messages that are transient across distributed snapshots (bands). Transient messages pose a special problem in time management, since all transient messages need to be accounted for to prevent messages arriving in the past. Wide-area communication, such as over the web, exacerbate the transient message problem.

Additional Observations

A few additional observations on the implications of XMSF approach are as below.

1. For high-performance computing implementation, it seems difficult to use XML-based interaction among processors. Use of native format seems imperative for achieving desired levels of performance (in our experience, even PVM/MPI message-passing interfaces proved to be too heavy weight for analytic distributed simulations). Perhaps a compilation approach can be used to retain XML's interoperability features without sacrificing runtime performance – textual XML specifications may be preprocessed and translated to native format for high frequency interactions.

2. XMSF-based front-ends appear to be useful as highly available and natural Web-based interfaces to results from online analytic simulations. Such capability, for example, enables non-simulation experts to easily access and use simulation facilities using interfaces that such users are already accustomed to (namely, web browsers).

3. Due the very nature of the web-based approach, the XMSF possesses its own strengths and weaknesses. Hence, it needs to be carefully applied only where it is beneficially applicable (e.g., integrating coarse units of models and simulations across relatively high latency links), and avoided at other uses (tightly coupled interactions over shared-memory or local area network).

We, as developers of middleware, such as FDK, which provides a high-performance implementation of the HLA RTI, are yet unclear as to the relation or distinction between HLA services and XMSF services. On the outset, one of our viewpoints is illustrated in Figure 4, which depicts XMSF as a grand facilitator of multiple HLA RTI-based simulations. While RTI is used as the integration mechanism among federates on a local area network (LAN) or metropolitan area network (MAN), the XMSF is used to provide higher-level integration of several such RTI-based clusters, using web technologies as the integrating machinery over local, metropolitan and wide area networks (WAN). It is XMSF, and not RTI, that the high-level user interacts with for high-level operations

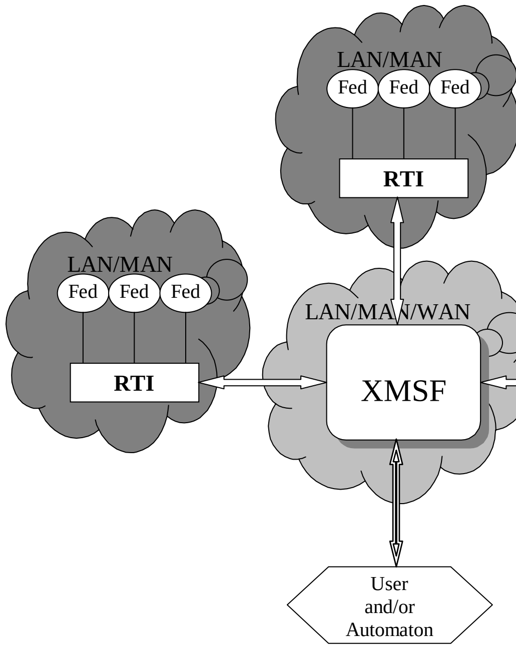

Figure 4: XMSF usage scenario to provide efficient and portable integrated execution capability over the web to multiple optimized RTI federations. The RTI-based executions are contained within LAN/MAN, while XMSF control can span LAN/MAN as well as WAN.

References

[1] D. Anagnostopoulos, "Experiment scheduling in faster-than-real-time simulation," presented at 16th Workshop on Parallel and Distributed Simulation, Washington, D.C., 2002.

[2] M. Hybinette and R. M. Fujimoto, "Cloning parallel simulations," *ACM Transactions on Modeling and Computer Simulation*, vol. 11, pp. 378-407, 2001.

[3] R. M. Fujimoto, *Parallel and Distributed Simulation Systems*: Wiley Inter-science, 2000.

[4] F. Wieland, "Practical parallel simulation applied to aviation modeling," presented at 15th Workshop on Parallel and Distributed Simulation, Lake Arrowhead, CA, 2001.

[5] K. S. Perumalla and R. M. Fujimoto, "Interactive Parallel Simulations with the Jane Framework," *Future Generation Computer Systems*, vol. 17, pp. 525-537, 2001.

[6] R. M. Fujimoto, T. McLean, K. S. Perumalla, and I. Tacic, "Design of High-performance RTI Software," presented at Workshop on Distributed Simulations and Real-time Applications, 2000.

[7] B. Fitzgibbons, T. McLean, and R. M. Fujimoto, "RTI Benchmark Studies," presented at Spring Interoperability Workshop, Orlando, FL, 2002.

[8] R. M. Fujimoto, S. Ferenci, M. Loper, T. McLean, K. S. Perumalla, G. F. Riley, and I. Tacic, "FDK Users Guide," College of Computing, Georgia Institute of Technology 2001/02/14 2001.

[9] K. S. Perumalla, R. M. Fujimoto, T. McLean, and G. F. Riley, "Experiences Applying Parallel and Interoperable Network Simulation Techniques in On-Line Simulations of Military Networks," presented at 16th Workshop on Parallel and Distributed Simulation, Washington, DC, 2002.

[10] K. S. Perumalla and R. M. Fujimoto, "Virtual Time Synchronization over Unreliable Network Transport," presented at Workshop on Parallel and Distributed Simulation, 2001.

URLs

1. The Federated Simulations Development Kit (FDK) http://www.cc.gatech.edu/computing/pads/fdk.html.
2. The interoperable network simulations, on-line simulation and emulation project at Georgia Tech http://www.cc.gatech.edu/computing/pads/nms.
3. The Scalable Simulation Framework http://www.ssfnet.org.

About the Author

Kalyan Perumalla is a Research Scientist at the College of Computing, Georgia Institute of Technology since 1997. He received his Ph.D. in Computer Science from the Georgia Institute of Technology in 1999. He has over 7 years of research and development experience in the area of parallel and distributed simulation systems, including high-performance RTI implementation, and large-scale simulation. His research interests include reverse computation, parallel and distributed simulation, network modeling and parallel combinatorial optimization.