Final Report for Period: 09/2008 - 08/2009
Submitted on: 09/01/2009
Principal Investigator: Fujimoto, Richard M.
Award ID: 0326431
Organization: GA Tech Res Corp - GIT
Submitted By:
Fujimoto, Richard - Principal Investigator
Title:
Collaborative Research: ITR: Global Multi-Scale Kinetic Simulations of the Earth's Magnetosphere Using Parallel Discrete Event Simulation

Project Participants

Senior Personnel

Name: Fujimoto, Richard
Worked for more than 160 Hours: Yes
Contribution to Project:

Name: Pande, Santosh
Worked for more than 160 Hours: Yes
Contribution to Project:

Name: Perumalla, Kalyan
Worked for more than 160 Hours: Yes
Contribution to Project:

Name: Omelchenko, Yuri
Worked for more than 160 Hours: Yes
Contribution to Project:
Worked on hybrid codes and parallel discrete event simulation of large scale physics and its performance prediction.

Name: Driscoll, Jonathan
Worked for more than 160 Hours: Yes
Contribution to Project:
Worked on PDES2 virtualization environment related to performance debugging and fine tuning.

Post-doc

Graduate Student

Name: Anand, Saswat
Worked for more than 160 Hours: Yes
Contribution to Project:
research in prediction functions to optimize execution of space simulaitons.

Name: Mosi, Shomari
Worked for more than 160 Hours: Yes
Contribution to Project:
Research in run time optimizations of parallel simulation software.

Name: Tang, Yarong
Worked for more than 160 Hours: Yes
Contribution to Project:
Optimistic parallel simulation of space particle code.

Name: Dave, Jagrut

Worked for more than 160 Hours: Yes

Contribution to Project:

Working in interfacing Scibernet codes to Musik parallel simulation system

Name: Li, Donghan

Worked for more than 160 Hours: Yes

Contribution to Project:

Compiler support for efficient parallel simulation.

Name: Yang, Kemin

Worked for more than 160 Hours: Yes

Contribution to Project:

Compiler optimizations for optimistic parallel execution.

Name: Thakare, Prashant

Worked for more than 160 Hours: Yes

Contribution to Project:

work on performance assessment system for MS project

Name: Llamas, Ignacio

Worked for more than 160 Hours: Yes

Contribution to Project:

preliminary research in providing support using Chombo software.

Name: Amara, Ashok

Worked for more than 160 Hours: Yes

Contribution to Project:

preliminary research in grid computing

Name: Thiruchiloganathan, Rajaprabhu

Worked for more than 160 Hours: Yes

Contribution to Project:

Worked on virtualization environment PDES2 adding support for generating statistics for performance debugging. Supported as a graduate research assistant.

Name: Sreeram, Jaswanth

Worked for more than 160 Hours: Yes

Contribution to Project:

Worked on PDES 2 virtualization environment generating implementation for priority queues for handling different event types. Supported as a graduate research assistant.

Name: Park, Alfred

Worked for more than 160 Hours: Yes

Contribution to Project:

Worked on distributed simulations for scalability and fault tolerance.Supported as a graduate research assistant.

Name: Gu, Yan

Worked for more than 160 Hours: Yes

Contribution to Project:

Yan is working on a remote execution capability for distributed simulations and emulations, and has developed applications focusing on network emulations.

Undergraduate Student

Technician, Programmer

Other Participant

Research Experience for Undergraduates

Organizational Partners

SciberQuest Inc.

Nanyang Technological University
We are collaborating with NTU on grid computing infrastructure software for distributed simulations.

University of Rostock
We are collaborating with the University of Rostock in a German-funded project to use the Aurora software for systems biology simulations.

Other Collaborators or Contacts

Adelinde Uhrmacher, University of Rostock is our collaborator concerning systems biology simulations.

Steve Turner, Nanyang Technological University, is our collaborator on grid-based distributed simulations.

Activities and Findings

Research and Education Activities:

A principal goal of this project is to determine whether discrete event simulation (DES) can be developed into a general technique for multi-scale modeling of physical systems. We have been engaged in several efforts in this area, and have been successful in using DES as a multi-scale technique. Specifically, we have successfully applied DES technology to PIC simulations (electrostatic, and electromagnetic hybrid), reactive computational fluid dynamics, diffusion equation, MHD, and elastodynamic problems. We have found DES to be stable, fast and time-accurate with superior metrics compared to standard techniques. Additionally, DES is extendable to arbitrary meshes (structured and unstructured) and we have implemented parallel version of it. One particularly encouraging development is the interest in the application of our technology to two-phase and three-phase fluid flow in fractured porous media, a subject of significant interest to oil companies and petroleum industries.

A second line of research has focused on creating scalable parallel discrete event simulation infrastructure to address runtime issues for executing large-scale PDES codes. Our efforts in this regard have focused on a grid computing infrastructure for automating many tasks associated with parallel execution, e.g., resource allocation, load balancing, and fault tolerance. An optimized, second generation implementation of the Aurora parallel simulation grid computing infrastructure has been developed for this purpose that provides optimized communications, improved reliability, and increased scalability through the support of a multi-threaded, multi-server architecture. In addition, we have continued development of reverse execution techniques for parallel execution, and developed applications for multiprocessor and network simulations to complement our work in physical system simulations.

We have been engaged in a variety of educational activities in conjunction with this project. Specifically, we have integrated research results from this effort in a new graduate level course in modeling and simulation that has been developed. More broadly, we have been developing new graduate degree programs in Computational Science and Engineering at Georgia Tech. A new Masters degree program has been created, and a new PhD program has been developed and began operation in Fall 2008. Finally, several masters and doctoral students have been actively engaged in this project.

Findings:

Multi-Dimensional General Purpose DES API (SciDES)

Architectural decisions make a significant impact on the robustness and performance of adaptive computational frameworks, of which DES is a good example. In the past period we have started developing a new DES application programming interface (API), whose purpose is to abstract (generalize) the development of uni-dimensional (dimension-independent), multi-physics, event-driven PIC and fluid simulation codes. In order to provide flexible and CPU-efficient numerical solutions on modern massively distributed computer systems, this API has been designed to meet a number of important criteria, which are summarized below. Note that DES provides a unique approach to building physics-driven simulation codes, which, remarkably, in many ways reflects the way modern computer operating systems (OS) are interfaced with device drivers. This is made possible by separating the DES code components into the immutable ôkernelö and flexible problem-dependent ôdriversö. The seamless interface between the simulation kernel and physics drivers is achieved via object-oriented programming and use of virtual (overloaded) class member functions (C++). Since our new simulation infrastructure is geared towards multi-dimensional problems, we have paid a great deal of attention to optimizing data calculations both memory and CPU wise.

Separation of the problem-dependent and problem-independent DES components. Examples of problem-dependent components include discretization schemes for particle/fluid equations of motion and MaxwellÆs equations, initial and boundary conditions, etc. Problem-independent details include data structures, memory and inter-processor communication routines, grid generation and partitioning algorithms, scheduling and synchronization algorithms, computational domain descriptors, etc. The problem-independent parts of DES compose a ôkernelö, whose purpose is similar to that of the core of a real-time OS (event prediction, processing, cancellation, preemption, inter-event communication, etc.). The physics (ôdriverö) code is written to provide problem-specific functionalities (event initialization, temporal updates, synchronization conditions, I/O, packing/unpacking event data into/from data streams, etc.) required by the kernel code.

Object-oriented programming (OOP) design. The robustness and flexibility (extendibility) of modern computational software critically depends on the designerÆs ability to effectively abstract and encapsulate the most essential algorithmic concepts of the computational framework. This is achieved by constructing ôclassesö, which define closely related data and operations (class methods). Class extendibility, achieved via the use of virtual functions, enables the designer to create codes highly tuned for specific applications without having to break the ôsystemö code. The advantages of OOP programming are especially important for the robust design of parallel PIC-DES codes, which handle closely interacting asynchronous particle and field objects distributed over multiple processors.

Intelligent data structures. Debugging and optimizing sophisticated parallel PIC codes becomes a daunting challenge if communication data containers (e.g., messages, stacks, queues, buffers, etc.) are not equipped with built-in data descriptors and mechanisms for ensuring object consistency (e.g., object type, size, timestamp, internal composition, etc). In our design physics events communicate with each other by generating events, which are packed/unpacked to/from data streams in a unified manner. This allows transporting a stream of heterogeneous information (e.g., particles, fluids, etc.) in a single data message.

Optimized data structures. In our previous work DES cells were represented by data structures containing both ôphysicsö (e.g., particle queues, fields, etc.) and DES components (logical states, flags, pointers to higher level structures, etc.). This prevented the effective separation of specific physics details from the underlying DES logic. Even more importantly, the old data arrangement led to inefficient multi-dimensional array operations. In our new design, the DES specifics are encapsulated separately from the actual physics data. This enables fast access to physical data and makes it possible to use Fortran subroutines for fast local data crunching.

Scalable DES Models

New discrete-event formulations of physics simulation models are emerging that can outperform traditional time-stepped models, especially in simulations containing multiple timescales. Detailed simulation of the EarthÆs magnetosphere, for example, requires execution of sub-models that operate at timescales that differ by orders of magnitude. In contrast to time-stepped simulation which requires tightly coupled updates to almost the entire system state at regular time intervals, the new discrete event simulation (DES) approaches help evolve the states of sub-models on relatively independent timescales. However, in contrast to relative ease of parallelization of time-stepped codes, the parallelization of DES-based models raises challenges with respect to their scalability and performance. One of the key challenges is to improve the computation granularity to offset synchronization and communication overheads within and across processors. Our previous work on parallelization was limited in scalability and runtime performance due to such challenges. We developed optimizations on DES-based plasma simulation models to improve parallel execution performance. The mapping of the model to simulation processes was optimized via aggregation techniques, and the parallel runtime engine was optimized for communication and memory efficiency. The net result is the capability to simulate hybrid particle-in-cell (PIC) models with over 2 billion ion particles using 512 processors on supercomputing platforms.

Physical systems often involve a wide spectrum of time scales, making traditional synchronous time integrators too slow even on the worldÆs fastest supercomputers. Asynchronous variational integrators (AVIs) have recently been introduced to address this challenge for a class of nonlinear elastodynamic problems. We developed a new and efficient algorithm for parallel AVIs. We proposed a new local-minima interpretation of the dependencies in the AVIs. Based on this interpretation, we show that the expected amount of parallelism in the AVIs is proportional to the number of elements of a mesh assuming a uniform distribution of the dependencies. Second, based on the local minima interpretation, we introduce a parallel algorithm for the AVIs guided by the dependency graph of the computation that avoids using priority queues, thereby improving scalability. Third, we developed an efficient multi-threaded implementation of the AVIs and introduce performance-optimization techniques using super-elements. Experimental results verify our theoretical analysis and demonstrate the effectiveness of the proposed algorithms.

Many computational applications involve multiple physical components and require exchanging data across the interface between them, often on parallel computers. The interface is typically represented by surface meshes that are non-matching, with differing connectivities and geometry. To transfer data accurately and conservatively, it is important to construct a common refinement (or common tessellation) of these surface meshes. Previously, Jiao and Heath developed an algorithm for constructing a common refinement by overlaying the surface meshes. The original algorithm was efficient and robust but unfortunately was complex and difficult to implement and parallelize. We developed a new algorithm for overlaying surface meshes. Our algorithm employs a higher-level primitive, namely face-face intersection, to facilitate easy parallelization of mesh overlay while retaining the robustness of the original algorithm. We also introduced a safeguarded projection primitive to improve the robustness against non-matching features and potential topological inconsistencies. We developed numerical examples to demonstrate the robustness and effectiveness of the new method on parallel computers.

High Throughput Parallel Simulations on Computational Grids

The excessive amount of time necessary to complete large-scale discrete-event simulations of complex systems continues to plague researchers and impede progress in many important domains. Parallel discrete-event simulation techniques offer an attractive solution to addressing this problem, however, effective, practical solutions have been elusive. This is largely because of the complexity and difficulty associated with developing parallel discrete event simulation systems and software. In particular, an effective parallel execution mechanism must simultaneously address a variety of issues, such as efficient synchronization, resource allocation, load distribution, and fault tolerance, to mention a few. Further, most systems that have been developed to date assume a set of processors is dedicated to completing the simulation computation, yielding inflexible execution environments. Effective exploitation of parallelization techniques requires that these issues be addressed automatically by the underlying system, largely transparent to the application developer.

We have developed an approach to execute large-scale parallel discrete event simulation programs over multiple processor computing resources that automates many of the tasks associated with parallel execution. Based on concepts utilized in volunteer distributed computing projects, we developed an approach that supports execution over computing platforms shared with other users. This approach automatically adapts as new processors are added or existing processors taken aware during the course of the execution. The runtime environment based on a master-worker approach is used to automatically distribute simulation computations over the available processors to balance workload, and to automatically recover from processor failures that may occur during the computation. Synchronization, a particularly important problem for parallel discrete-event simulation computations, is handled automatically by the underlying runtime system.

Utilizing desktop grid infrastructures is challenging for parallel discrete event simulation (PDES) codes due to characteristics such as inter-process messaging, restricted execution, and overall lower concurrency, problems that do not exist in typical volunteer computing projects. The Aurora2 system was developed that uses an approach that simultaneously provides both replicated execution support and scalable performance of PDES applications through public resource computing. It improves upon our previous implementation by providing a multi-threaded distributed back-end system, low overhead communications middleware, and an efficient client implementation. We developed the Aurora2 architecture and issues pertinent to PDES executions in a desktop grid environment that must be addressed when distributing back-end services across multiple machines. We quantified improvements over the first generation Aurora system through a comparative performance study detailing PDES programs with various scalability characteristics for execution over desktop grids.

We examined issues concerning optimistic time management on public-resource computing infrastructures and desktop grids. The master/worker approach used for these platforms calls for a rethinking of optimistic synchronization and the development of new mechanisms and protocols specific to this paradigm. Approaches to rollback, message cancellation, state management and state saving are described. Challenges specific to adapting optimism to this computing paradigm were examined and optimizations and overhead reduction techniques were developed. The performance of an optimistic master/worker implementation in DGSIM, a framework supporting PDES over desktop grids was analyzed and compared to a conservative synchronization time management system operating in the same environment.

Master/worker PDES applications incur overheads not found in conventional PDES executions executing on tightly coupled machines. We developed four techniques for reducing the overheads of master/worker PDES systems on public resource and desktop grid infrastructures. Work unit caching, pipelined state updates, expedited message delivery, and adaptive work unit scheduling mechanisms were developed that provide significant reduction in overall overhead when used in tandem. We produced performance results showing that an optimized master/worker PDES system using these techniques can exhibit performance comparable to a traditional PDES system for a queueing network and a particle physic simulation.

We developed an approach to achieve efficient execution of optimistic PDES systems in cloud computing architectures. Execution of traditional optimistic PDES systems on such architectures can lead to an excessive number of rollbacks. We developed the TW-SMIP protocol that dynamically adjusts the execution of each logical process based on local parameters and straggler messages. The protocol avoids barrier synchronizations, and instead dynamically limits forward execution of logical processes to reduce the amount of erroneous computation and generation of incorrect messages. An implementation of this approach was developed, and performance measurements showed that it yielded superior performance compared to the original Time Warp synchronization algorithm.

Finally, we expanded development of our approaches to optimistic parallel simulation to other application domains. We examined use of this technology to perform instruction-level simulations of distributed memory multi-processor systems. A static program analysis approach was developed to optimize pre-processed simulated applications in order to remove certain overheads associated with forward event execution and to enable reversible execution. Reverse execution of floating point operations were also considered, and approaches to ensuring truly reversible operations were developed. Performance measurements indicate this approach offers promise in speeding up parallel multi-processor simulations.

In addition, we examined the application of the Aurora framework for large-scale cell-biological simulations, a field that has emerged as data and knowledge concerning these systems increases and biologists call for tools to guide wet-lab experimentation. We examined an approach to exploit parallelism in this domain in the integration of spatial aspects, which are often crucial to a model's validity. We developed a parallel and distributed variant of the Next-Subvolume Method (NSM), a method that augments the well-known Gillespie Stochastic Simulation Algorithm (SSA) with spatial features. We examined requirements imposed by this application on a parallel discrete event simulation engine to achieve efficient execution, and completed a thorough performance analysis in the context of a grid-inspired Aurora system.

Training and Development:

We incorporated work on the project in a new course on modeling and simulation that covers both discrete and continuous simulation methods. This course was offered in Spring semester 2007, 2008, and 2009. A course in parallel and distributed simulation was offered in Fall semester 2007. Both courses include class projects related to research conducted in this program.

The project websites (http://sciberquest.com and http://www.cc.gatech.edu/computing/pads/physics) include introductory materials as well as highlights of some of our research results.

Motivated in part by work on this project, we created new masters and doctoral degree programs in computational science and engineering. These interdisciplinary programs are jointly offered by eight departments at Georgia Tech. The programs begain operation in Fall 2008.

This interdisciplinary research has been very beneficial to the students working on the project in exposing the computer science students to techniques and codes used in physical system simulation.

Outreach Activities:

We have also presented several seminar/invited talks highlighting our modeling efforts and on the performance benefits of our simulation engines at universities and research labs. In a separate effort, we have been exploring use of the Aurora software for missile defense simulations for the Missile Defense Agency, and have integrated a tool called the Ballistic Missile Defense Benchmark program into the Aurora infrastructure.

We have been collaborating with a group lead by Professor Adelinde Uhrmacher at the University of Rostock to use Aurora for systems biology simulations. Faculty and students from Rostock visitied Georgia Tech in December 2006, and are expected to visit again in December 2007 to continue work on this effort.

We have similarly been collaborating with a group a Nanyang Technological University in Singapore on grid computing infrastructure for distributed simulation. Multiple visits have taken place both to Singapore and Atlanta on this collaborative project.

We developed a minority and women outreach summer intern program to encourage students in underrepresented groups to consider doctoral studies. The CSE Research Undergraduate Intern Summer Experience (CRUISE) operated in summer 2008 and hosted 15 minority and women students on the Georgia Tech campus. These students engaged in research in high performnce computing and distributed simulation.

Invited talks:

Omelchenko YA, and Karimabadi H.,Self-adaptive event-driven simulation of multi-scale plasma systems, Calspace-IGPP/UCR Conference, March 27, Palm Springs, 2006.

R. Fujimoto, ôAurora: An Approach to High throughput Parallel Simulation,ö Universitaet der Bundeswehr Munich, Germany, April 2006.

Omelchenko YA, and Karimabadi H., A New Multiscale Technique for Time-Accurate Geophysics Simulations, Fall AGU, 2006.

R. Fujimoto, ôTowards Flexible, Reliable, High Throughput Parallel Discrete Event Simulations,ö Cooperative Research in Science and Technology Symposium, March 2007.

R. Fujimoto, ôModeling, Simulation, and Parallel Computation: The Future is Now,ö SpringSim Æ07 Society for Modeling and Simulation Intl., and Simulation Interoperability Standards Organization Joint Keynote Address, Norfolk, VA, March 2007.

Omelchenko, YA, Discrete-event simulation (DES): A new approach to asynchronous time integration of heterogeneous systems, The Universities Forum on Reservoir Description and Simulation (UFORDS), September 2-6, 2007.

Contributed presentations:

Omelchenko YA, and Karimabadi H., Discrete-event simulations of multi-scale plasmas and fluids, APS conference on Plasma Physics, 2006.

Karimabadi, H., and Y. Omelchenko, Extension of adaptive mesh refinement to electromagnetic PIC simulations, ISSS-8 conference, Kauai, 2007.

Omelchenko YA, and Karimabadi H., Adaptive control of multiple scales in space plasma simulations, ISSS-8 conference, Kauai, 2007.

A. Park, R. M. Fujimoto, ôA Scalable Framework for Parallel Discrete Event Simulations on Desktop Grids,ö 8th IEEE/ACM International Conference on Grid Computing (Grid 2007), September 2007.

J.-C. Huang, X. Jiao, R. M. Fujimoto, H. Zha, ôDAG-Guided Parallel Asynchronous Variational Integrators with Super-Elements,ö Summer Computer Simulation Conference, July 2007.

A. Naborskyy, R. M. Fujimoto, ôUsing Reversible Computation Techniques in a Parallel Optimistic Simulation of a Multi-processor Computing System,ö Principles of Advanced and Distributed Simulation, June 2007.

Journal Publications

H. Karimabadi, J. Driscoll, Y.A. Omelchenko, and N. Omidi, "A New Asynchronous Methodology for Modeling of Physical Systems: Breaking the Curse of
Courant Condition", Journal of Computational Physics, p. 755, vol. 205, (2005). Published,

H. Karimabadi, Y. Omelchenko, J. Driscoll, N. Omidi, R. Fujimoto, S. Pande, and K. S.
Perumalla, "A New Approach to Modeling Physical Systems: Discrete Event Simulations of
Grid-based Models", PARA'04 Workshop on State-of-the-art in Scientific Computing, p. 1, vol. , (2004). Published,

M. Loper and R. M. Fujimoto, "A Case Study in Exploiting Temporal Uncertainty in Parallel Simulations", International Conference on Parallel Processing, p. 1, vol. , (2004). Published,

M. Loper and R. M. Fujimoto, "Exploiting Temporal Uncertainty in Process-Oriented Distributed Simulations", Proceedings of the Winter Simulation Conference, p. 395, vol. , (2004). Published,

K. Perumalla, "Musik, A Micro-Kernel for Parallel/Distributed Simulation", ACM/IEEE Workshop on Principles of Advanced and Distributed Simulation, p. 59, vol. 19, (2005). Published,

H. Karimabadi, Y. Omelchenko, J. Driscoll, R. Fujimoto and K. Perumalla, "A New Simulation Technique for Study of Collision-less Shocks: Self Adaptive Simulations", Annual International Astrophysics Conference (IGPP), p. 1, vol. 1, (2005). Published,

Y. Tang, K. Perumalla, R. Fujimoto, H. Karimabadi, J. Driscoll, and Y. Omelchenko, "Parallel Discrete Event Simulations of Physical Systems using Reverse Computation", ACM/IEEE Workshop on Principles of Advanced and Distributed Simulation, p. 26, vol. 19, (2005). Published,

C. Lobb, Z. Chao, R. Fujimoto, and S. Potter, "Parallel Event-Driven Neural Network Simulations Using the Hodgkin-Huxley Neuron Model", ACM/IEEE Workshop on Principles of Advanced and Distributed Simulation, p. 16, vol. 19, (2005). Published,

H. Karimabadi, J. Driscoll, J. Dave, Y. Omelchenko, K. Perumalla and R. Fujimoto, "Parallel Discrete Event Simulations of Grid-based Models: Asynchronous Electromagnetic Hybrid Code", Springer-Verlag Lecture Notes in Computer Science, p. 1, vol. 1, (2004). Published,

Omelchenko YA, Karimabadi H., " Event-driven, hybrid particle-in-cell simulation: A new paradigm for multi-scale plasma modeling,", JOURNAL OF COMPUTATIONAL PHYSICS 216 (1): 153-178 JUL 20 2006, p. 153, vol. 216, (2006). Published,

Omelchenko YA, Karimabadi H., "Self-adaptive time integration of flux-conservative equations with sources", JOURNAL OF COMPUTATIONAL PHYSICS 216 (1): 179-194 JUL 20 2006, p. 179, vol. 216 (1), (2006). Published,

Karimabadi H, Driscoll J, Dave J, et al., "Parallel discrete event simulations of grid-based models: Asynchronous electromagnetic hybrid code", LECTURE NOTES IN COMPUTER SCIENCE 3732: 573-582 2006., p. 573, vol. , (2006). Published,

Karimabadi, H., et al., "Global hybrid simulations of the earth?s magnetosphere,", ASP Conference Series,, p. , vol. , ( ). Accepted,

Omelchenko, Y., et al, "Self-adaptive event-driven simulation of multi-scale plasma systems", ASP Conference Series, p. , vol. , (2006). Published,

Y. Tang, R. M. Fujimoto, K. Perumalla, H. Karimabadi, J. Driscoll, Y. Omelchenko, "Optimistic Parallel Simulations of Physical Systems using Reverse Computation", Transactions of the Society for Modeling and Simulation Intl., p. 61, vol. 82, (2006). Published,

H. Karimabadi1, J. Driscoll1, J. Dave, Y. Omelchenko, K. Perumalla, R. Fujimoto, N. Omidi, "Parallel Discrete Event Simulation of Grid-Based Models: Asynchronous Electromagnetic Hybrid Code", Springer-Verlag Lecture Notes in Computer, p. 573, vol. 3732, (2006). Published,

K. Perumalla, R. M. Fujimoto, H. Karimabadi, "Scalable Simulation of Electromagnetic Hybrid Codes", Third International Workshop on Simulation of Multiphysics Multiscale Systems, p. , vol. , (2006). Published,

A. Park, R. M. Fujimoto, "Aurora: An Approach to High Throughput Parallel Simulation", Principles of Advanced and Distributed, p. , vol. , (2006). Published,

K. Perumalla, R. M. Fujimoto, P. J. Thakare, S. Pande, H. Karimabadi, Y. Omelchenko, J. Driscoll, "Performance Prediction of Large-Scale Parallel Discrete Event Models of Physical Systems", Winter Simulation Conference, December 2005., p. , vol. , (2005). Published,

K. Perumalla, R. Fujimoto, S. Pande, H. Karimabadi, J. Driscoll and Y. Omelchenko,, "Virtual Simulator: An Infrastructure for Design and Performance-Prediction of Massively Parallel Codes", American Geophysical Union (AGU) 86(52), Fall Meeting, p. , vol. , (2005). Published,

K. Perumalla, R. M. Fujimoto, H. Karimabadi, "Efficient Parallel Execution of Event-Driven Electromagnetic Hybrid Models", International Journal for Multiscale Computational Engineering, p. 27, vol. 5, (2007). Published,

J.-C. Huang, X. Jiao, R. M. Fujimoto, and H. Zha, "DAG-Guided Parallel Asynchronous Variational Integrators with Super-Elements", 2007 Summer Computer Simulation Conference, p. 0, vol. 1, (2007). Published,

A. Jain and X. Jiao, "Overlaying Surface Meshes: Extension and Parallelization", 16th International Meshing Roundtable, p. 0, vol. 1, (2007). Published,

A. Park and R. M. Fujimoto, "A Scalable Framework for Parallel Discrete Event Simulations on Desktop Grids", 8th IEEE/ACM International Conference on Grid Computing, p. 0, vol. 1, (2007). Published,

R. M. Fujimoto, A. Park, J.-C. Huang, "Towards Flexible, Reliable, High Throughput Parallel Discrete Event Simulations", Cooperative Research in Science and Technology Symposium, p. 0, vol. 1, (2007). Published,

A. Naborskyy, R. M. Fujimoto, "Using Reversible Computation Techniques in a Parallel Optimistic Simulation of a Multi-processor Computing System", Principles of Advanced and Distributed Simulation, p. 0, vol. 1, (2007). Published,

Omelchenko YA, and Karimabadi H.,, "A time-accurate explicit multi-scale technique for gas dynamics", J. Comp. Physics, p. 282, vol. 226, (2007). Published,

Omelchenko YA, Karimabadi H., "Event-driven, hybrid particle-in-cell simulation: A new paradigm for multi-scale plasma modeling", J. Comp. Physics, p. 179, vol. 216, (2006). Published,

Omelchenko YA, Karimabadi H., "Self-adaptive time integration of flux-conservative equations", J. Comp. Physics, p. 179, vol. 216, (2006). Published,

A. Park and R. M. Fujimoto, "Parallel Discrete Event Simulation on Desktop Grid Computing Infrastructures", International journal of Simulation and Process Modeling, p. , vol. , (2009). Accepted,

A. Park;R. M. Fujimoto;, "Efficient Master/Worker Parallel Discrete Event Simulation", Principles of Advanced and Distributed Simulations, p. , vol. , (2009). Published,

A. Park;R. M. Fujimoto;, "Optimistic Parallel Simulation over Public Resource-Computing Infrastructures and Desktop Grids", Workshop on Distributed Simulations and Real-Time Applications, p. , vol. , (2008). Published,

M. Jeschke;R. Ewald;A. Park;R. Fujimoto;A. Uhrmacher;, "Parallel and Distributed Spatial Simulation of Chemical Reactions", Principles of Advanced and Distributed Simulations, p. , vol. , (2008). Published,

A. Park;R. M. Fujimoto;, "A Scalable Framework for Parallel Discrete Event Simulations on Desktop Grids", 8th IEEE/ACM International Conference on Grid Computing, p. , vol. , (2007). Published,

A. Malik, A. Park, R. M. Fujimoto, "Optimistic Synchronization of Parallel Simulations in Cloud Computing Environments", Proceedings of the IEEE International Conference on Cloud Computing, p. , vol. 1, (2009). Published,

Books or Other One-time Publications

R. M. Fujimoto, K. S. Perumalla, G. F. Riley,, "Network Simulation", (2007). Book, Published
Bibliography: Morgan & Claypool Publishers

Web/Internet Site

URL(s):
http://itr.sciberquest.com, http://www.cc.gatech.edu/computing/pads/physics
Description:
Provides information on the project as well as results.

Other Specific Products

Product Type:
Software (or netware)
Product Description:
Musik Parallel/Distributed Simulation Engine software. This software defines the common

API used for development of Physics codes, and implements synchronization mechanisms for parallel execution of the simulations.

Sharing Information:

The software is made available on the web and can be freely downloaded.

Product Type:

Software (or netware)

Product Description:

PDES 2 is a tool for simulating the simulations - it allows performance prediction and hotspot identification for large scale simulations without actually running them. It uses a techniques called virtualization that emulates the actual run.

Sharing Information:

Currently the software is under development jointly with SciberQuest. We propose to put it on the web exposing APIs using which users will be able to model their simulation to find out their performance demands.

Product Type:

Software (or netware)

Product Description:

Aurora distributed simulation infrastructure software.

Sharing Information:

Available to other researchers on request. Information about Aurora is disseminated through publications describing the software.

Contributions

Contributions within Discipline:

Major contribution of the work are in the Computer Science and Science disciplines (e.g., Physics). This project is contributing the computer science field (or more accurately the Computational Science and Engineering discipline) through advances in parallel discrete event simulation techniques and grid computing infrastructures. It is advancing the state of knowledge in science and engineering disciplines through the development of new approaches to modeling physical systems.

Contributions to Other Disciplines:

Parallel discrete event simulation has broad application to a number of fields in science and engineering, e.g., communication network design, transportation systems, and supply chains, to mention a few. We have demonstrated the application of our methods to the simulation of chemical reaction processes in collaboration with a group from Germany.

Contributions to Human Resource Development:

This interdisciplinary project allows one to develop an understanding of Physics as well as the Computer Science. That allows much broader skill set of our graduate students when they go out in job market (MS as well as PhD).

We are training several graduate students who will contribute to the longer term development of science and technology in parallel simulation techniques. We have used results from this project, e.g., the Aurora framework, to define course projects in graduate simulation courses at Georgia Tech.

Contributions to Resources for Research and Education:

The usik parallel discrete event simulation engine has been made available to the research community for development of parallel simulation applications.

The Aurora grid computing infrastructure is also available for use, and is being utilized by a group at the University of Rostock in Germany for biological system simulations.

Contributions Beyond Science and Engineering:

Our partner in this project, SciberQuest, is commercializing the technology for broader use. Similarly, GTech Systems Inc., co-founded by one of the PIs on this project (Fujimoto) is commercializing the Aurora software for broader use, especially in missile defense applications.

Conference Proceedings

Categories for which nothing is reported:

Any Conference