Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.

OPTIMISTIC PARALLEL DISCRETE EVENT SIMULATION OF THE EVENT-BASED TRANSMISSION LINE MATRIX METHOD

David W. Bauer Jr.
Ernest H. Page

The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102, U.S.A.

ABSTRACT

In this paper we describe a technique for efficient parallelization of digital wave guide network (DWN) models based on an interpretation of the finite difference time domain (FDTD) method for discrete event simulation. Modeling methodologies based on FDTD approaches are typically constrained in both the spatial and time domains. This interpretation for discrete event simulation allows us to investigate the performance of DWN models in the context of optimistic parallel discrete event simulation employing reverse computation for rollback support. We present parallel performance results for a large-scale simulation of a 3D battlefield scenario, 100km^2 and at a height of 100m with a resolution of 100m in the X-, Y-planes, and 10m in the Z-plane for 754 simultaneous radio wave transmissions.

1 INTRODUCTION

Parallel discrete event simulation (PDES) technology has been employed successfully over the last 30 years to improve the performance of many modeling methodologies. Recently, researchers in this field have begun to investigate the efficacy of PDES as applied to modeling physical systems. Common approaches to modeling physical systems include, but are not limited to, ray-tracing and the finite difference time domain (FDTD) methods. However these methods traditionally have not benefitted from discrete event simulation because of constraints in the spatial and time domains, related to each method. For example, the FDTD method is limited by the Courant-Lewy-Friedrichs (CFL) condition (Courant, Friedrichs, and Lewy 1928, Courant, Friedrichs, and Lewy 1967).

In the past few years, researchers have begun to adapt these methods to the discrete event paradigm. Notably, a 2005 study applied discrete event simulation to the particle-in-cell (PIC) method (Karimabadi et al. 2005) and achieved a two order of magnitude increase in performance. The PIC method has a long history, reviewed in Birdsall (1991), and is commonly used in the area of plasma physics. The break-through in this approach was the removal of the CFL constraint, which allowed for a two order of magnitude improvement of the runtime. This interpretation was then studied in the context of parallel discrete event simulation by Tang et al. (2006) for a 1D spacecraft model, which improved the performance by an additional order of magnitude.

In 2006, James Nutaro published a study adapting the FDTD method with respect to digital wave guide network wave simulation to the discrete event paradigm. Here, the focus was on the propagation of electromagnetic waves through a complex 3D environment. A formal description of the algorithm was defined for discrete event systems. Nutaro indicated greater than an order of magnitude improvement in the cost of computation over the FDTD method, for high resolution 3D models of Digital Waveguide Networks (DWN).

In this paper we apply the formal model outlined in Nutaro (2006) to the PDES paradigm, specifically optimistic synchronization enabled by reverse computation. We chose this method to study because the algorithm allowed for resolutions independent of the wavelength of the electromagnetic waves modeled. Our challenge is to apply this method to a DWN physical simulation for battlefield scenarios where the scale of the environment (100km long X 100km wide X 100m in height) is large and where the number of radio wave transmissions is large, in this case 754 simultaneous wave transmissions.

The main contribution of our work is the application of the Event-Based Transmission Line Matrix (ETLM) modeling method to the area of PDES known as optimistic simulation. Optimistic synchronization was first proposed by Jefferson (1985) and allows as fast as possible event execution where violations of the causality constraint may occur. Where violations occur, the causality constraint is then preserved by *rolling back* improperly processed events,

1-4244-1306-0/07/$25.00 ©2007 IEEE

restoring model state prior to the violation, and then allowing processing to resume.

Initial implementations of the rollback mechanism focused on storing model entity or logical process (LP) states. During rollback, these states could then be recalled prior to the causal violation, ensuring proper execution of the model. Initially, *copy-state saving* was employed, whereby an LPs entire state would be stored for each event processed by an LP. Improvements were soon introduced, such as *incremental state-saving*, where only the LPs modified state variables were stored for each event processed. *Infrequent state-saving* attempted to reduce the amount of memory consumed to support rollback by decreasing the frequency of LP state-saving.

Advances in PDES and optimistic synchronization have recently focused on a relatively new approach to supporting rollback, *reverse computation*. Here, rollback is supported by performing the inverse operations on an LPs state for each event processed. This approach has been employed successfully in a variety of modeling contexts, and the main benefit is that memory is not typically not consumed storing individual LP states. One trade-off with this approach is that the computational costs associated with reverse computation can be high compared to state-saving. Second, some operations may not be reverse-computable without the loss of data in the model. These types of operations are known as *destructive statements*, and one example would be floating point operations.

The paper is organized as follows: we outline previous work in this area in Section 2. We provide the details of the Event-Based Transmission Line Matrix (ETLM) modeling method and the parallel, reverse computable model in Section 3. We perform a performance study in Section 4. Our summary of the paper is given in Section 5 and Future Work is indicated in Section 6.

2 RELATED WORK

In the seventeenth century Issak Newton proposed a corpuscular model for the phenomena of light. In the same century, Christian Huygens proposed a model of light based on wave propagation. Quantum physics indicates that the models are compatible; that light in particular, and electromagnetic radiation in general, possesses both granular (photons) and wave properties. For electromagnetic radiation occurring a microwave frequencies, the physical properties exhibited by the wave model dominates to determine the effects of propagation and scattering. In 1971, Johns and Beurle (1971) proposed a numerical modeling technique based on Huygen's model for solving 2-D scattering problems. This work was followed up with extensions to three dimensions and the concept of dielectric loading (Akhtarzed and Johns 1975, Johns 1974c, Johns 1974b, Akhtarzad and Johns 1974). Numerous researchers since have added features and extended the method to include concepts such as variable mesh size, simplified nodes, error correction techniques, and anisotropic media. The TLM algorithm was outlined concisely in Johns (1974a), and a subsequent analysis of the theory and applications of TLM are outlined in Hoefer (1985).

To date, we are aware of two modeling methods for physical systems that have been interpreted for the discrete event paradigm: particle-in-cell (PIC) and Event-Based Transmission Line Matrix (ETLM). The PIC method is derived from the numerical method called *finite difference time domain* or FDTD (Shlager and Schneider 1995). This approach is commonly used when modeling physical systems because it is highly accurate and provides a bounding of the error in the computed result. This error is related to the representation of the environmental space in the model as a multi-dimensional grid. The spacing between points in the grid determines the unknown quantities and generates the error bounds for the given data. While this method is capable of generating very precise results (e.g., using small grid-spacings), an unknown amount error is introduced by the input representation of the environmental space.

The PIC method has a long history going back over 40 years and is outlined well in Birdsall (1991). Interestingly, this idea was conceived for discrete models. In practice, this approach is most commonly encoded as a time-stepped simulation, possibly because of the simplicity of time-stepped simulation systems. While this method has a long history, it was not until 2005 that it was interpreted for the discrete event paradigm in Karimabadi et al. (2005). The major impact of this work is that the Courant-Levy-Freidrichs condition does not apply in the context of the asynchronous PIC-DES method. This allowed for a two order of magnitude improvement in the runtime of a 1-dimensional model of a spacecraft interacting with the solar winds.

In concert, Tang et al. (2006) implemented this model in an optimistic parallel discrete event simulation that employed reverse computation for rollback. Nearly another order of magnitude improvement in the runtime of the model was seen utilizing 8 processors.

The second interpretation of a numerical method for physical systems for DES is called *Event-Based Transmission Line Matrix* or ETLM. This method has been used to study both fire-spreading (Muzy et al. 2005) as well as electromagnetic wave propagation (Kuruganti and Nutaro 2006). This method is based on a discrete approximation of a continuous structure. This relaxes the constraint on the distance that can be used between points in the grid used to model the spatial domain. Consideration of the grid cell spacing must be considered in the computation of the wave velocity when the wavelengths are either much larger, or much smaller than the spacing.

A validation study was performed in Kuruganti and Nutaro (2006) for electromagnetic (EM) wave propagation using ETLM. The experiment was configured with a single transmitter placed on one side of a laboratory, and 90 receivers in a line on the other side. The transmitter signal was then calibrated and measured at each receiver and recorded. Then the room was modeled in 3D, and the experiment was conducted in simulation. The measured results were then compared for accuracy against the ETLM model and a ray-tracing simulation result based on ray-tracing. The results established ETLM as a highly accurate method for EM wave propagation. More importantly, the performance of the ETLM was shown to increase as the environment became increasingly cluttered, and that the ETLM method did not suffer from the number of receivers contained in the model. Conversely, it is well understood that ray-tracing complexity increases linearly with respect to the number of receivers, and quadratically with respect to the number of elements in the environment.

Nearly as fast as these methodologies are being interpreted for the discrete event paradigm, researchers in PDES are investigating the degree to which they can be parallelized. PDES can be generally categorized in terms of two major processor synchronization methods known as *conservative* and *optimistic*.

The Time Warp mechanism outlined in Jefferson (1985) described for the first time the optimistic synchronization protocol and rollback mechanism. Traditionally, state-saving has been employed as the rollback mechanism, outlined by the numerous studies in this area. The major approaches to rollback mechanisms include copy state saving, incremental state-saving (Steinman 1993, Gomes 1996), and a large amount of effort has been expended on infrequent state saving (Lin and Preiss 1991, Lin et al. 1993, Fleischmann and Wilsey 1995).

More recently, the technique of reverse computation has been suggested as an approach to rollback that does not rely on storing state information (Carothers, Perumalla, and Fujimoto 1999). Here, LP states are restored by computing the inverse operations for each improperly executed event processed. Several reverse computation models have been designed for a variety of models, including communications network modeling (Yaun et al. 2003, Yaun, Carothers, and Kalyanaraman 2003, Bauer et al. 2006), and physical systems (Tang et al. 2006). Additional optimizations for reducing memory consumption related to the storing of processed events was proposed in Bauer and Page (2007) for reverse computation and Li and Tropper (2004) for state-saving.

3 MODELING METHOD

In this section we outline the Event-Based Transmission Line Matrix modeling method, and the approach taken for parallelization using the reverse computation technique for rollback in a Time Warp simulator. The formal model description provided here is related entirely from Nutaro (2006). Those seeking details on how the DES-TLM method was derived from numerical methods are strongly encouraged to review this previous work. A high level overview of the model is presented for the purpose of understanding how the model is implemented in a parallel discrete event simulator.

3.1 Event-Based Transmission Line Matrix Model

The Event-Based Transmission Line Matrix (ETLM) modeling method requires two main components in order to perform wave propagation, relating to the time and spatial domains of the modeled physical system. First, a discrete representation of the spatial environment modeled must be formed. This is performed by conceptualizing the environmental space as a dimensional grid over the environment. This grid may be homogenous or inhomogeneous in terms of dielectric constants and/or resolution. The second main component is the algorithm for scatter-gather of the electromagnetic fields for a point in the grid and includes all of the computations related to wave propagation and attenuation: velocity, impedance, amplitude and wavelength and where the material bulk modulus and density are specified for each point in the grid.

Within any homogenous region of the grid, the structural and numerical properties of the digital wave guide network are preserved. At the interface between two disparate homogenous regions, waves are both transmitted and reflected. Both the transmitted and reflected waves share the same frequency, though their amplitudes and velocities are defined by their respective homogenous regions. Special care must be taken at material interfaces to ensure proper wave propagation through each region, and so a "junction" model is specified at interfaces. The junction model ensures that waves are sampled consistently for the varying time domains on either side of the interface (as dictated by the wavelength).

The scatter-gather algorithm is propagates a wave through the spatial grid elements. The external transition function defined in Nutaro (2006), Formula 36, describes how energy is gathered at a grid cell. The total energy at a cell at time t is defined as the sum of the attenuated wave amplitudes incident upon a cell in all dimensions at time t. The time advance function for a cell is defined as infinite if the energy value at a cell is zero, d/V otherwise, where d is the cell grid spacing and V is the wave velocity and $d \gg \lambda$ (where λ is the wavelength).

Once the time advance has expired, the cell is then responsible for scattering the energy contained therein to the surrounding cells (Formula 38 in Nutaro 2006). The stored energy is attenuated for the current cell, and scattered equally in all dimensions except for those directions from which input energy was received. In the directions where

Algorithm 1 LP input energy event handling. Each dimension must be updated for the time advance caused by the incident wave. The spatial offset timestamp ts is determined by the grid spacing for the dimensional plane divided by the incident wave velocity d/V.

Steps to Handle Input Energy Displacement at a Cell

```
for each dimension, d
    if no time advance set
        set time advance[d] to spatial ts

stored energy += input energy
```

energy was received, the total output energy is reduced by the incoming energy for that direction. For example, in a 1-D homogenous model, energy traveling from the left to the right, and arriving at a given cell should be scattered in all directions. However, the scattering of the energy back to the left is reduced by exactly the same amount of energy as received from the left neighbor. Therefore, the amount of energy to be scattered to the left is zero, and the wave continues propagation to the right.

3.2 Parallel Simulation

At each dimensional intersection within the grid, an LP is defined and performs the scatter-gather algorithm. While junction LPs can be defined between inhomogeneous grid regions, for simplicity we model a complex 3-D terrain environment for the ground plane only. This simplification allows us to ignore transmitted waves (e.g., into the Earth) and focus solely on the main algorithm.

Each LP state contains an array storing the incoming energy displacement values by direction, and the size of the array is twice the number of dimensions modeled (corresponding to the directions -X, +X, -Y, +Y and -Z, +Z within a 3-dimensional environment). Because the grid may be heterogenous in scale in different dimensions, we also store the time advance values in an array sized by the number of dimensions. A field for storing the total displacement is also used. The intersection of 2 or more dimensions in the grid defines a grid cell LP that implements the scatter-gather algorithm.

There are two types of events handled by the LP, input energy displacement values and time advance requests. Upon receiving a displacement from a neighbor, the displacement array stores the incident wave energy, and updates the time advance value for each dimension. Finally, the total energy displacement for the cell is updated, as shown in Algorithm 1.

Upon receiving a time advance request, the LP event handler must then determine the corresponding energy values to propagate for each dimension, as shown in Algorithm 2. The LP first attenuates the energy stored at the cell according to the grid cell dielectric constants. If an attenuation threshold is defined and met, no energy is propagated. Otherwise, propagation begins on a per dimension basis. Each dimension has a corresponding spatial coefficient that modifies the propagated energy value based on the total number of dimensions. The incident energy for a direction is then subtracted from the energy value. If the resulting energy value is above a defined threshold for a direction, then that value is propagated to the neighboring LP in that direction.

Algorithm 2 LP time advance event handling. First the stored energy amplitude attenuates with respect to the material dielectric constants and the cell spacing. If the result is above the threshold, then the scattering model is performed. Energy is displaced in all directions, minus any input displacements, if and only if the final displacement in any one direction continues to be above the threshold. Finally, the LP state variables are reset to their resting positions, which may be zero, or contain a distribution representing background noise.

Steps to Handle Time Advance Event

```
stored energy *= wave loss coefficient

if stored energy < attentuation threshold
    done

for each direction, d
{
    displacement = (spatial coeff *
                    stored energy) -
                    stored displacement[d]

    if displacement >= attenuation threshold
        send displacement in direction d
}

reset stored energy
reset directional displacements
```

At the completion of a time advance request, the cell's total displacement is reduced by the amount of the displacement that generated the request. In our particular scenario, all transmissions have the same properties (i.e., frequency and simultaneous transmission). Therefore, the time advance requests in the grid are synchronized in time (simultaneous transmissions), and the total cell displacement and directional replacements may be simply restored to the steady state (i.e., no displacement from a transmission, but possibly background noise).

3.3 Reverse Computation

As noted in Carothers, Perumalla, and Fujimoto (1999), the key property that reverse computation exploits is that a majority of the operations that modify the state variables are "constructive" in nature. That is, the undo operation for such operations requires no history. Only the most current values of the variables are required to undo the operation. For example, operators such as ++, –, +=, -=, *= and /= belong to this category. Note, that the *= and /= operators require special treatment in the case of multiply or divide by zero, and overflow/underflow conditions. More complex operations such as circular shift (swap being a special case), and certain classes of random number generation also belong here.

Operations of the form $a = b$, floating point operations, modulo and bit-wise computations that result in the loss of data, are termed to be *destructive*. Typically these operations can only be restored using conventional state-saving techniques. However, we observe that many of these destructive operations are a consequence of the arrival of data contained within the event being processed. Frequently we can make use of the swap operation to make this operation reversible by swapping LP state variables with variables stored in the events, avoiding resorting to the state-saving technique.

The reverse computation for the pseudo-code presented in Algorithms 1 and 2 is rather straightforward. Many of the operations applied to the LP state are floating point precision, and therefore destructive. The approach we take here is to swap the LP state variable being destroyed with the incoming message data area in the event.

Reverse computing the portion of the LP state needed to compute the time advance function is even simpler. A high level examination of the model indicates that all waves in an homogenous region have the same frequency. This simplifying assumption indicates that all waves also have the same wavelength. So no computation need be performed, as the time advance variables are simply reset to infinity.

We do not need to be concerned with reverse computing LP state variables to handle the case where multiple waves occur at a cell, since they would occur simultaneously. The rollback mechanism is required to rollback each wave incident at a cell for an instant in time, the effect of which is to return the cell to equilibrium (i.e., no energy).

4 PERFORMANCE STUDY

In this section we outline the computing testbed used to perform experiments. We also outline the model scenario used, and the performance achieved across multiple processors.

4.1 Computing Testbed

All experiments were conducted on the MITRE Corporation “Hive” Cluster that contains 35 compute nodes, or “drones”. Each Hive compute node is a dual-processor, dual-core AMD Opteron server configured with 2.2 GHz processors and 8.192 GBs of RAM. The AMD Opteron 2000-series chip enables 64-bit computing, and provides up to 24GB/s peak bandwidth per processor using HyperTransport technology. The DDR2 memory controller is 128-bits wide and provides up to 6.4GB/s of bandwidth per processor. Our RAM configuration consisted of a 4 x 2.048GB configuration of 667MHz DDR2 ECC RAM modules. The network interconnect used was a Cisco 48-port GbE switch.

4.2 Simulator Setup

The simulation executive was configured using the performance tuning parameters listed in Table 1. Because of the large amount of memory consumed by the 10 million grid LPs, optimistic event memory was constrained by the following equation:

$$\lceil nLP/nCPU \rceil * C \tag{1}$$

where C was set to 1.2, or 20% more than the sequential case requires.

Table 1: Simulator input parameters.

Parameter	**Value**
Batch Loop Size	1024 events
Fujimoto GVT Interval	32 batch loops
7 O’Clock GVT Interval	0.001 secs
Kernel Processes	8

GVT batch and interval parameters were set at 1024 and 32 respectively. Thus, up to 32,768 events will be processed per processor between GVT epochs. These settings where experimentally determined to yield the highest level of performance for the model for a particular computing testbed.

The simulator used was *ROSS* which employs Fujimoto’s GVT algorithm for shared memory multiprocessors when executed on a single compute node, and the Seven O’clock GVT algorithm when executed across multiple compute nodes (Carothers, Bauer, and Pearce 2002, Fujimoto and Hybinette 1997, Bauer et al. 2005). The ROSS simulator is a general-purpose simulator exposing a simple API for the creation of PDES models.

Kernel Processes were introduced in Carothers, Bauer, and Pearce (2002) and aggregate the LP processed event lists. The intent of the aggregation is to reduce the cost of traversing all LPs processed event lists during fossil collection. However, aggregated processed event lists have the side-effect of potentially longer rollbacks. Because we are allocating a small amount of optimistic memory, processing elements, or PEs, are unable to run too far into the future, and so a small number of KPs does not negatively impact simulator efficiency (i.e., ratio of rolled back events to total events processed during sequential simulation). It does give the benefit of extremely high performance fossil collection.

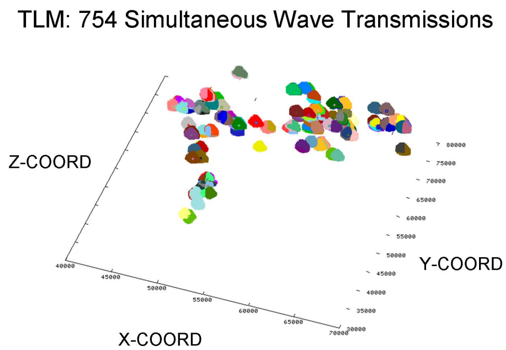

Figure 1: Scenario includes 754 radios simultaneous transmitting over complex 3D terrain. The XY labels indicate DTED1 coordinates.

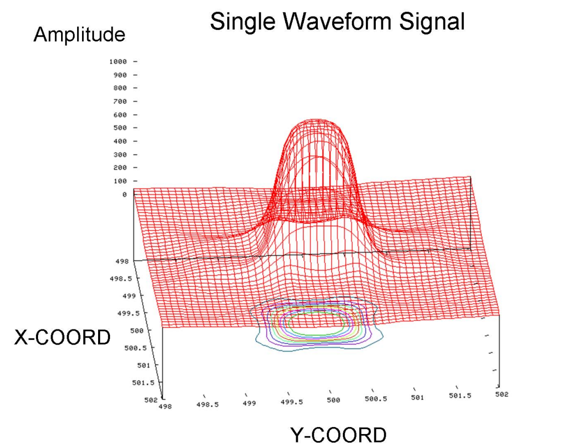

Figure 2: An example wave propagation over time. The wave amplitude is shown for a 1KHz signal over a flat ground plane.

4.3 Experiment Setup

Our experimentation focused on a hypothetical radio communication network scenario based in Southwest Asia. The terrain modeled was a 100km^2 area and to a height of 100m, uniformly sampled at a 100m resolution in the XY plane, and 10m in the Z plane. Level 1 digital terrain elevation (DTED1) data was obtained from GeoCommunity (www.geocomm.com). The model scenario consisted of 754 radio LPs simultaneously transmitting a 10MHz signal with a threshold wave amplitude of 1KHz (Figure 1). At this resolution, the battlefield model consumed 7.7 GB of memory representing 10 million grid LPs.

Interestingly, the worst case for this method turns out to be a sparsely populated environment. The reason for this behavior is that there is comparatively little wave attenuation through the atmosphere. Because of the degree of effort required to describe buildings and foliage on the grid, we were satisfied with a 3D complex ground terrain environment only, understanding that additional environmental elements would likely only improve the performance via rapid attenuation and localized propagation.

Other parameter settings are given in Table 2 for propagating isotropic waves. The spatial coefficient defines how energy is propagated in 3 dimensions outward from a cell. The spatial ground coefficient determines how waves reflecting from the ground plane are attenuated. The wave loss coefficient determines wave amplitude attenuation between cells for the atmosphere. The wave velocity is used to determine the time advance steps during propagation. Finally, the amplitude threshold determines a cutoff threshold for the signal, where the signal is indistinguishable from background noise. An example isotropic wave propagation over a flat ground plane is shown in Figure 2.

Table 2: Experiment input parameters.

Parameter	Value
Spatial Coefficient	0.33
Spatial Ground Coefficient	0.9
Wave Loss Coefficient	0.9
Wave Velocity	1.0
Amplitude Threshold	100

4.4 Parallel Performance

4.4.1 Priority Queues

Because this is only the second model of a physical system for PDES, we investigate the impact on performance using a variety of priority queues for event scheduling. Understanding how to tune the simulator requires understanding the behavior of the model. In a vacuum with no obstacles, an isotropic wave will propagate as a sphere. This indicates that an increasing number of events are consumed over time at a factor related to the surface area of a sphere, $4\pi r^2$. Also, each event on the wavefront surface has the same timestamp, therefore the choice of priority queue used in the simulator has a major impact on performance, as shown in Figure 3.

Calendar queue (Brown 1988) performance is known to have a complexity of O(n) when the frequency of ties among events is high, accounting for it's poor performance in all cases. The performance of the Heap (Deo and Prasad 1992) is a large improvement, but the Splay Tree (Rönngren and Ayani 1997) implementation clearly attained the best performance at almost 1.3 million events per second on a single processor core. The main reason for comparing the model performance using different priority queues is that the Calendar queue reports greater than 40-fold speedup over sequential using 4 CPUs. This is misleading when placed in

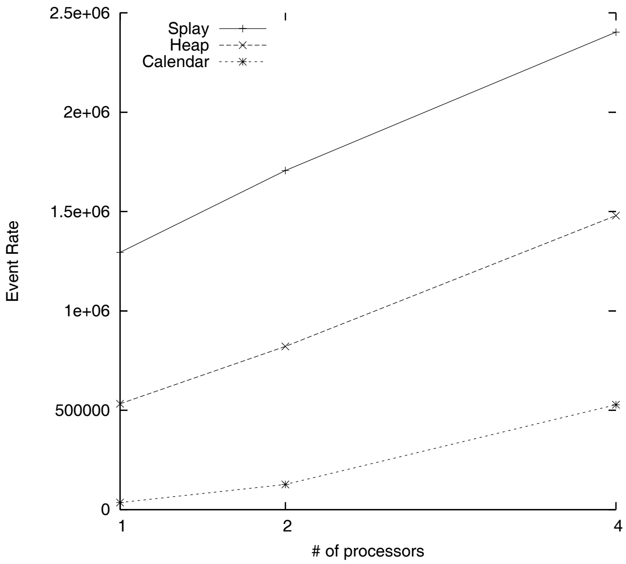

Figure 3: Simulator performance implementing a variety of priority queues. Results collected on a single cluster node.

the context of other queue algorithms. Conversely, speedup based on the Splay Tree results reports only 53% improvement across 4 CPUs. We hypothesize that the constrained amount of optimistic memory (20% over sequential) throttles how far into the future Time Warp can execute events. In all following experiments, the Splay Tree is used.

4.4.2 LP Mappings

Thus far in our experimentation, LPs have been allocated to PEs on a round-robin basis through the grid. Starting at the origin of the cubic grid (i.e., a corner), we begin round-robin allocation in the X-axis, then the Y-axis, and finally the Z-axis. Therefore, each YZ plane in the X-axis is mapped to a processor in turn, as shown in Figure 4.

The LP-PE mapping dictates performance in terms of the number of events that must be passed between processors. Crucial to using multiple compute nodes, the LP-PE mapping *must* minimize the number of events passed between nodes. An iterative improvement on the round-robin mapping would be to combine the YZ-planes allocated to a processor so that they are concurrent. Doing so reduces the percent of remote events from 23% to 0.17%, and allows us to utilize multiple compute nodes.

Figure 5 illustrates the performance of the model on multiple compute nodes. Using multiple nodes, less memory is consumed by the model and performance improves dramatically. Therefore, we increase the model event population by reducing the amplitude threshold to 10, and increase the amount of optimistic memory (C = 2.0).

In the 20 and 25 CPU cases, the work was relatively even across all CPUs. However, in the 50 CPU case, 25 of the processors never execute an event, because no wave

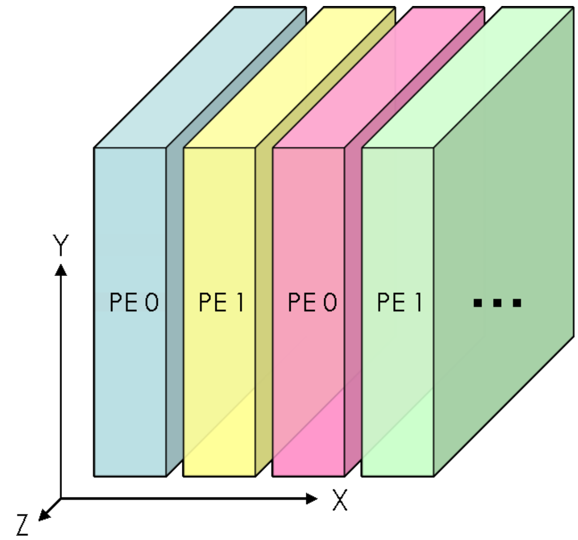

Figure 4: Round-robin mapping of LPs to PEs.

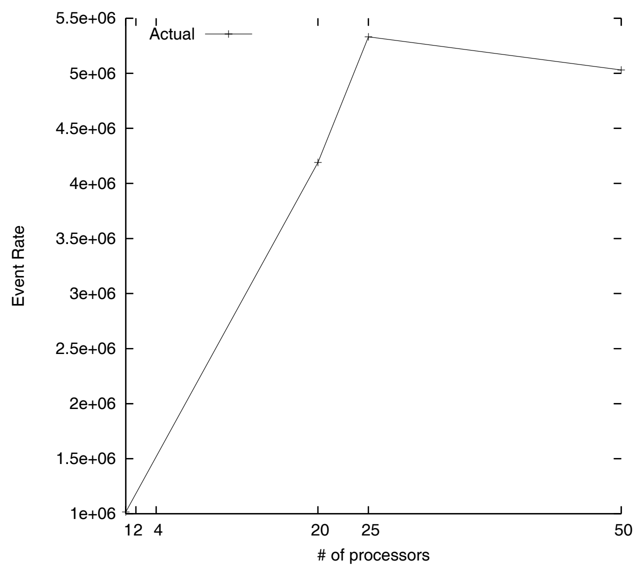

Figure 5: Hive Cluster performance. In the 50 CPU case the model no longer maps work to all processors, and performance drops. Performance is directly impacted by the LP-PE mapping.

propagation occurs in those areas of the grid (namely, the edges).

Our conclusion can only be that a static mapping can only achieve limited success when using a large number of processors, and the scenario is not static (i.e., the radios are mobile). A dynamic LP-PE mapping could focus all available processors into the areas of the grid active in wave propagation. Also, it should be noted that for this model scenario, with this mapping, we had 25 idle CPUs available that could have focused effort on dynamic load-balancing. We leave an investigation of dynamic load-balancing to future work.

5 CONCLUSIONS

We have investigated the Event-Based Transmission Line Matrix modeling method within the context of optimistic parallel discrete event simulation with reverse computation. We have illustrated the performance of a complex terrain, encompassing 100km^2 with realistic signal parameters. Our model captured the physical effects of reflection, diffraction with respect to electromagnetic wave propagation. Model performance in an optimistically synchronized, parallel discrete event simulation employing reverse computation proved to be an efficient mechanism for parallelization. However, simulations must have dynamic load balancing of LP to PE allocations in order for all CPUs to become utilized in dynamic scenarios.

6 FUTURE WORK

In the future we would like to study additional features of electromagnetic wave propagation, such as scattering and transmission through inhomogeneous materials (i.e., adaptive mesh refinement, buildings and foliage). From a communications perspective, we would also like to study the effects of multipath, MIMO channel modeling, background interference effects, modulation as well as a variety of other waveforms.

Other concerns raised in Tang et al. (2006) related to a static mapping of grid elements to processors. Further investigation is required to understand the performance impact of the LP-PE mappings, and to determine if improvements can be achieved over the runtime of a scenario with dynamic load balancing.

Finally, we would like to study representing this model in the context of general purpose processors, such as graphics processors and the Cell Broadband Engine. We believe this modeling method will fit these paradigms well, and that additional improvements in performance can be gained.

REFERENCES

Akhtarzad, S., and P. Johns. 1974. Solution of 6-component electromagnetic fields in three space dimensions and time by the tlm method. *Electronics Letters* 10:535.

Akhtarzed, S., and P. Johns. 1975. Solution of Maxwell's equations in three space dimensions and time by the tlm method of numerical analysis. *Institution of Electrical Engineers, Proceedings* 122:1344–1348.

Bauer, D. W., and E. H. Page. 2007. An approach for incorporating rollback through perfectly reversible computation in a stream simulator. In *PADS '07: Proceedings of the 21st International Workshop on Principles of Advanced and Distributed Simulation*, 171–178. Washington, DC, USA: IEEE Computer Society.

Bauer, D. W., G. Yaun, C. D. Carothers, M. Yuksel, and S. Kalyanaraman. 2005. Seven-o'clock: A new distributed gvt algorithm using network atomic operations. In *PADS '05: Proceedings of the 19th Workshop on Principles of Advanced and Distributed Simulation*, 39–48. Washington, DC, USA: IEEE Computer Society.

Bauer, D. W., M. Yuksel, C. Carothers, and S. Kalyanaraman. 2006. A case study in understanding ospf and bgp interactions using efficient experiment design. In *PADS '06: Proceedings of the 20th Workshop on Principles of Advanced and Distributed Simulation*, 158–165. Washington, DC, USA: IEEE Computer Society.

Birdsall, C. 1991, April. Particle-in-cell charged-particle simulations, plus monte carlo collisions with neutral atoms, pic-mcc. *IEEE Transactions on Plasma Science* 19 (2): 65–85.

Brown, R. 1988. Calendar queues: A fast o(1) priority queue implementation for the simulation event set problem. *Communications of the ACM (CACM)* 31:1220–1227.

Carothers, C. D., D. W. Bauer, and S. O. Pearce. 2002. Ross: A high-performance, low memory, modular time warp system. *Journal of Parallel and Distributed Computing*.

Carothers, C. D., K. S. Perumalla, and R. M. Fujimoto. 1999. Efficient optimistic parallel simulations using reverse computation. *ACM Trans. Model. Comput. Simul.* 9 (3): 224–253.

Courant, R., K. Friedrichs, and H. Lewy. 1928. ber die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen* 100 (1): 32–74.

Courant, R., K. Friedrichs, and H. Lewy. 1967. On the partial difference equations of mathematical physics. *IBM Journal*:215–234. English translation of the 1928 German original.

Deo, N., and S. Prasad. 1992. Parallel heap: an optimal parallel priority queue. *J. Supercomput.* 6 (1): 87–98.

Fleischmann, J., and P. A. Wilsey. 1995. Comparative analysis of periodic state saving techniques in time warp simulators. *SIGSIM Simul. Dig.* 25 (1): 50–58.

Fujimoto, R. M., and M. Hybinette. 1997. Computing global virtual time in shared-memory multiprocessors. *ACM Trans. Model. Comput. Simul.* 7 (4): 425–446.

Gomes, F. 1996. *Optimizing incremental state-saving and restoration*. Ph. D. thesis, University of Calgary.

Hoefer, W. 1985. The Transmission-Line Matrix Method–Theory and Applications. *Microwave Theory and Techniques, IEEE Transactions on* 33 (10): 882–893.

Jefferson, D. R. 1985. Virtual time. *ACM Trans. Program. Lang. Syst.* 7 (3): 404–425.

Johns, P. 1974a. A new mathematical model to describe the physics of propagation. *Radio Electron. Eng* 44:657–66.

Johns, P. 1974b. Application of the transmission-line matrix method to homogeneous waveguides of arbitrary cross-section. *Proc. Inst. Electr. Eng* 119:209–215.

Johns, P. 1974c. The Solution of Inhomogeneous Waveguide Problems Using a Transmission-Line Matrix. *Microwave Theory and Techniques, IEEE Transactions on* 22 (3): 209–215.

Johns, P., and R. Beurle. 1971. Numerical solution of 2-dimensional scattering problems using a transmission-line matrix. *Proc. IEE* 118 (9): 1203–1208.

Karimabadi, H., J. Driscoll, Y. A. Omelchenko, and N. Omidi. 2005. A new asynchronous methodology for modeling of physical systems: breaking the curse of courant condition. *J. Comput. Phys.* 205 (2): 755–775.

Kuruganti, T., and J. Nutaro. 2006. Validation radio wave propagation model. Oak Ridge National Laboratory Technical Report.

Li, L., and C. Tropper. 2004. Event reconstruction in time warp. In *PADS '04: Proceedings of the eighteenth workshop on Parallel and distributed simulation*, 37–44. New York, NY, USA: ACM Press.

Lin, Y.-B., and B. R. Preiss. 1991. Optimal memory management for time warp parallel simulation. *ACM Trans. Model. Comput. Simul.* 1 (4): 283–307.

Lin, Y.-B., B. R. Preiss, W. M. Loucks, and E. D. Lazowska. 1993. Selecting the checkpoint interval in time warp simulation. In *PADS '93: Proceedings of the seventh workshop on Parallel and distributed simulation*, 3–10. New York, NY, USA: ACM Press.

Muzy, A., E. Innocenti, A. Aiello, J.-F. Santucci, and G. Wainer. 2005. Specification of discrete event models for fire spreading. *Simulation* 81 (2): 103–117.

Nutaro, J. 2006. A discrete event method for wave simulation. *ACM Trans. Model. Comput. Simul.* 16 (2): 174–195.

Rönngren, R., and R. Ayani. 1997. A comparative study of parallel and sequential priority queue algorithms. *ACM Trans. Model. Comput. Simul.* 7 (2): 157–209.

Shlager, K., and J. Schneider. 1995, August. A selective survey of the finite-difference time-domain literature. Antennas and Propagation Magazine, IEEE Magazine.

Steinman, J. S. 1993. Incremental state saving in speedes using c++. In *WSC '93: Proceedings of the 25th conference on Winter simulation*, 687–696. New York, NY, USA: ACM Press.

Tang, Y., K. S. Perumalla, R. M. Fujimoto, H. Karimabadi, J. Driscoll, and Y. Omelchenko. 2006. Optimistic simulations of physical systems using reverse computation. *Simulation* 82 (1): 61–73.

Yaun, G., C. D. Carothers, and S. Kalyanaraman. 2003. Large-scale tcp models using optimistic parallel simulation. In *PADS '03: Proceedings of the seventeenth workshop on Parallel and distributed simulation*, 153. Washington, DC, USA: IEEE Computer Society.

Yaun, G. R., D. W. Bauer, H. L. Bhutada, C. D. Carothers, M. Yuksel, and S. Kalyanaraman. 2003. Large-scale network simulation techniques: examples of tcp and ospf models. *SIGCOMM Comput. Commun. Rev.* 33 (3): 27–41.

AUTHOR BIOGRAPHIES

DAVID W. BAUER JR. is a Senior Simulation Systems Engineer at the MITRE Corporation. He earned a Ph.D.,M.S., and B.S. from Rensselaer Polytechnic Institute in 2005, 2004, and 2000, respectively. Prior to joining MITRE, he was a research scientist at AT&T and GE. His research interests include parallel and distributed systems, simulation, wired and wireless networking, and computer architecture. His e-mail address is <dwbauer@mitre.org>.

ERNEST H. PAGE is a member of the technical staff for The MITRE Corporation. He received the Ph.D. in Computer Science from Virginia Tech in 1994. He serves on the editorial boards of ACM Transactions on Modeling and Computer Simulation, SCS Simulation, SCS Journal of Defense Modeling and Simulation, and the Journal of Simulation. He has served as the ACM SIGSIM representative to the WSC Board of Directors since 2001, and currently holds the position of Chair. His e-mail address is <epage@mitre.org>.