

SCALABLE RF PROPAGATION MODELING ON THE IBM BLUE GENE/L and CRAY XT5 SUPERCOMPUTERS

David W. Bauer Jr.

The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102, USA

Christopher D. Carothers

Department of Computer Science
Rensselaer Polytechnic Institute,
110 Eighth Street,
Troy, NY 12180, U.S.A.

ABSTRACT

We present a performance analysis for a highly accurate, large-scale electromagnetic wave propagation model on two modern supercomputing platforms: the Cray XT5 and the IBM Blue Gene/L. The electromagnetic wave model is used to simulate the physical layer of a large-scale mobile ad-hoc network of radio devices. The model is based on the numerical technique called Transmission Line Matrix, and is implemented in a Time Warp simulation package that employs reverse computation for the rollback mechanism. Using *Rensselaer's Optimistic Simulation System* we demonstrate better than real-time, scalable parallel performance for network scenarios containing up to one million mobile radio devices, highly accurate RF propagation and high resolution, large-scale complex terrain.

1 INTRODUCTION

Since its inception in 1985, the Time Warp protocol (Jefferson 1985) for parallel discrete event simulation (PDES) has been limited in its ability to efficiently scale to many processors for models that communicate a large amount of data between them. It is well known that model dynamics heavily impact the parallel performance of parallel discrete event simulation (PDES) systems. Today, that is beginning to change as modern supercomputing platforms focus on more than FLOP rates, and instead focus on the balance of processing power to memory and network bandwidth (Adiga et al. 2002, Gara et al. 2005). We present an analysis of the parallel performance of a Time Warp simulator executing on modern supercomputing architectures, namely the IBM Blue Gene/L and Cray XT5, executing an RF propagation model that communicates a large amount of data between processors.

The premise behind Time Warp, or *optimistic simulation*, is to process events in parallel under the assumption that they are arriving in timestamp order. A major component of Time Warp is the rollback mechanism that functions to correct causal errors in the timeline by rolling back the state of the model, correcting for events that are processed out-of-order. The rollback mechanism has widely been viewed as the limiting factor in these systems and has been a major area of research over the past two decades. The primary implementation for rollback relies on saving the state of the model such that the entities, or logical processes (LPs) can be restored when errors in the *causality constraint* occur (Lampert 1978). A great deal of research has been performed investigating optimizations for state-saving (West 1988, Lin et al. 1993, Steinman 1993, Fleischmann and Wilsey 1995, Gomes 1996, Quaglia 1999, Chen and Szymanski 2003, Li and Tropper 2004, Zeng et al. 2004).

In 1999, a second technique was proposed called *reverse computation* (Carothers et al. 1999). This technique restores the state of the model by computing the inverse operations for each event being rolled back. While this approach is not without its limitations (e.g., the irreversibility of floating point operations without loss of precision), it has allowed for much larger scale models because LP state is no longer preserved by the simulator, consuming large amounts of memory. Frequently limitations of this approach can be overcome through high level inspection of the model, and code can be generated to compute the inverse operations less directly than at the instruction level (L'Ecuyer and Andres 1997, Bauer and Page 2007a). In addition to memory savings, this approach typically involves minimal overhead in the forward execution of events, minimizing the costs for supporting rollback in the common case – that is, events are processed in timestamp order and no causal error occurs.

The model used in our performance study is based on the numerical technique called Transmission Line Matrix (TLM). TLM accurately models light at microwave frequencies as energy values propagating through a complex environment represented by a mesh. This technique has applications in electromagnetics and acoustics, and in this paper models the physical layer in a mobile ad-hoc network (MANET) containing up to 1 million radio devices. The environment contains

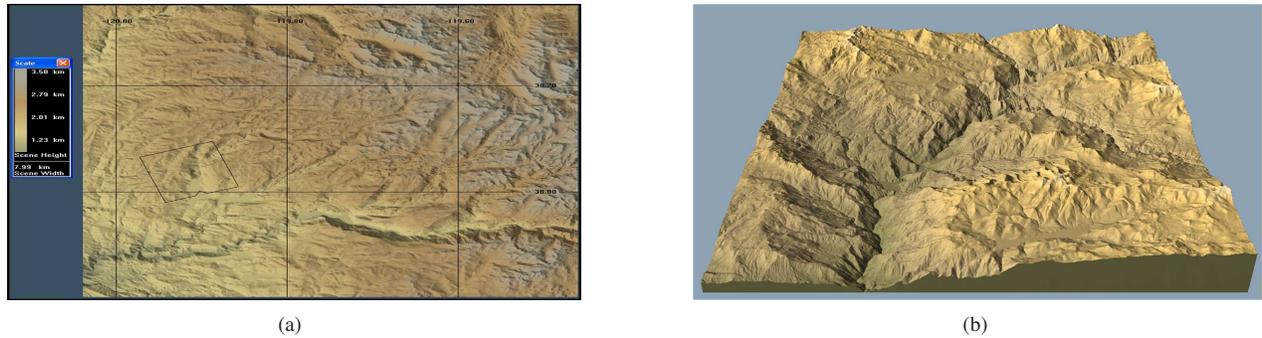


Figure 1: 2D and 3D images of a portion of the mountainous terrain used in this study. This image represents 1/4 of the actual terrain modeled, encompassing the foothills of the Rocky Mountains in California between the 37th and 38th parallel and from -120 to -119 degrees at a $\frac{1}{9}$ arc second resolution.

mountainous terrain in 3D, and the mesh contains 1 billion cells. Radio mobility precludes a static, a priori parallel decomposition in the spatial domain, and the 3D environment dictates that the event population has a cubic growth rate in the time domain. The event population is multiscale; mobility events occur at a fixed interval of 1 second while wave events travel near the speed of light through the mesh.

The primary contribution of this paper is a parallel performance analysis of Time Warp with reverse computation executing a highly dynamic model on the IBM Blue Gene/L and Cray XT5 supercomputers. Our results demonstrate good scaling on either platform, however we do observe a normalized 15-fold improvement on the Blue Gene/L over the XT5. After describing the TLM model in Section 2, and the supercomputing architectures in Section 3, we present a performance study in Section 4. This performance analysis demonstrates our second contribution in the area of network modeling and simulation. For the first time we report far better than real-time performance for a MANET scenario that contains: highly accurate physical layer communications, including the effects of diffraction, reflection and MULTIPATH; a large-scale, complex terrain; and one million mobile radios.

2 TRANSMISSION LINE MATRIX MODEL

Transmission Line Matrix (TLM) models light at microwave frequencies based on Christian Huygen's model of light as a wave (Johns and Beurle 1971, Johns 1974). TLM has applications in acoustics and electromagnetic wave propagation, and can be used to study radio wave propagation in wireless communication networks. TLM is a mesh refinement technique with two main components: a discrete representation of the spatial environment in 1, 2 or 3 dimensions (mesh), and an algorithm for scattering and gathering of the electromagnetic field energies through the mesh. The model is capable of capturing the primary effects of wave propagation, namely, *diffraction, reflection, and scattering*, each of which contributes to the multipath effect. This model provides highly accurate signal strengths for mobile ad-hoc networks and other wireless networking technologies operating in complex terrain as compared to traditional line-of-sight models that neglect multipath effects.

An excellent description of the conversion of the TLM model from ordinary differential equations to discrete event simulation is reported in (Nutaro 2006). A comparison of the discrete model to experimental measurements was further reported in (Kuruganti and Nutaro 2006), demonstrating highly accurate results. In 2007, the model was ported to a parallel discrete event simulation package called *Rensselaer's Optimistic Simulation System (ROSS)* and tested for a scenario consisting of a 100km² environment at a resolution of 100m and 754 mobile radios (Bauer and Page 2007b).

Representing the mesh as a uniform structured grid at a resolution of 100m in the XY plane and 10m in the Z-axis, the number of cells was 100 million. In discrete event simulation, mesh cells are realized as Logical Processes (LPs) and compute the scattering function; energy values are then communicated between cells as events. Radios are a second type of LP and compute mobility, initiate wave transmissions and handle receiving signal values. Scalability was limited to 25 CPUs on a prototypical Linux cluster, and runtime improvement was 50%. The limited scalability was primarily due to the limited network bandwidth. As radio waves expand in a 3-dimensional mesh, the event population has a cubic growth rate. Scalability was further limited by the mobility patterns of the radios, which precluded an optimal static decomposition of the mesh for parallelization.

For this investigation we move from cluster computing to supercomputing, and we are able to improve the scale of the model to 1 million radio LPs. Fidelity of the terrain was also increased to 10m in all dimensions, resulting in a total of 1 billion cell LPs. Figures 1 illustrate portions of the terrain modeled for this study. The overall terrain size remains 100km². The terrain data was gathered from the US Geological Societies *National Map Seamless Server National Elevation Dataset* $\frac{1}{9}$ arc sec (NED) online at <http://seamless.usgs.gov/index.php>.

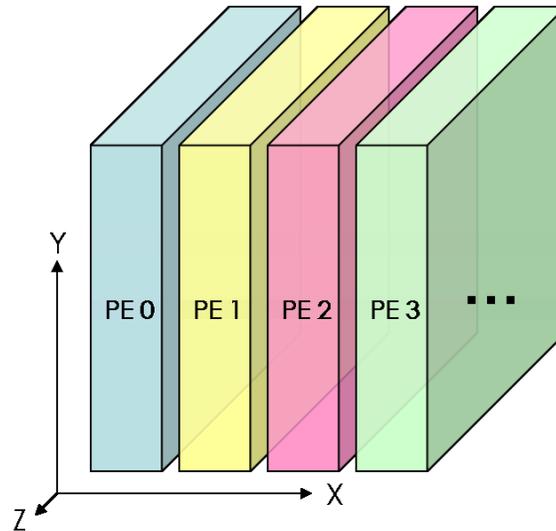


Figure 2: Parallel decomposition of the TLM model

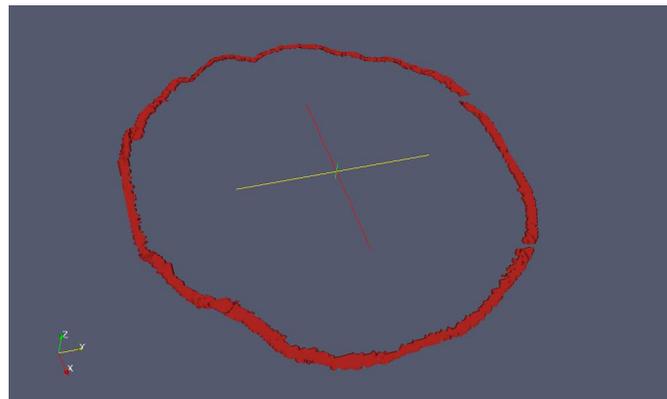


Figure 3: Image of a single wave transmission in the TLM model. As the Z-axis is truncated, the wave becomes torus-shaped as it continues to propagate in the XY directions (terrain omitted from image).

Scaling TLM on the Cray and IBM supercomputers was challenging because decomposing the spatial domain of the cell LPs to physical processors (PEs) can be performed in a variety of ways, none of which lead to optimal performance. Figure 2 illustrates the spatial decomposition of the terrain along the X-axis of the mesh grid. Previously, for an environment of size 100km^2 at a resolution of 100m, this yielded 1,000 X-axis rows, limiting the number of CPUs used to 1,000. Increasing the resolution of the environment allows us to scale up the CPU limit and the benefit is more accuracy in the results generated by the model. At a resolution of 10m, the environment now yields 10,000 X-axis rows, allowing us to utilize up to 10,000 CPUs. Scaling the model by increasing the resolution generates 100 million LPs for each Z-layer modeled, and for our experiments we used a Z-axis height of 100m for a total of 1 billion mesh LPs.

The number of radios modeled is a configurable parameter of the simulation, and radios are placed at uniformly random locations over the terrain. Radio mobility is defined by the random walk algorithm. For simplicity, each radio travels at a rate of 10 m/s (one cell) and changes direction according to a uniformly distributed number of timesteps at a resolution of 1 second. The range of the distribution is defined by configurable parameters. Without realistic mobility trace data, the random walk algorithm ensures that unrealistic scenarios do not occur where the radios are too close together to communicate due to interference. The density of the radio population dictates whether radios will be in range for communications. No optimizations are made for out-of-range radios and so the amount of work in the scenario remains constant with respect to the radio density and relatively balanced over the processors. The final configurable parameter is the frequency of radio wave transmission. The parameter can be set in the range of 0..1 and indicates the probability that a radio will transmit a wave at a given timestep according to a uniform distribution.

Figure 3 illustrates a radio wave propagating over the complex terrain in 3 dimensions. Note that the TLM model only computes the “front” of the electromagnetic wave, and not the full wave. As such, the wave takes on a torus-like shape when the height of the mesh in the Z-axis is far smaller than the XY-plane. In this investigation we are primarily concerned with

the effects of complex terrain on radio wave transmissions and are not concerned with propagation into the higher levels of the atmosphere. In addition, the model currently models isotropic antennas (no null area) rather than dipole antennas. Assuming an orientation parallel to the ground, a dipole antenna waveform would exhibit similar behavior.

Scaling TLM in Time Warp was challenging because of the fixed width timestamps on the events and the difference in the time scales between mobile radios (10 m/s) and the speed of RF propagation (speed of light, 3×10^8 m/s).

3 SUPERCOMPUTING ARCHITECTURES

In this study we use two modern supercomputing architectures: the Cray XT5 and the IBM Blue Gene/L. We describe the relevant features of each architecture.

3.1 The IBM Blue Gene/L Architecture

The Blue Gene philosophy holds that more powerful processors are not the answer when it comes to winning the massively parallel scaling war (Adiga et al. 2002). Instead, the Blue Gene/L architecture balances the computing power of the processor against the data delivery speed of the network. This philosophy led the designers to create smaller, lower power compute nodes (only 27.5 KW per 1024 nodes) consisting of two IBM 32-bit PowerPCs running at only 700 MHz with a peak memory per node of 1.0 GB. Each Blue Gene rack is composed into 1024 nodes consisting of 32 drawers with 32 nodes in each drawer. Additionally, there are specialized I/O nodes that perform all file I/O. Nominally there is one I/O node for every 32 compute nodes.

Interconnecting both drawers of nodes and racks are five specialized primary networks. The first is the point-to-point network which allows data to be sent between nodes. This network is a 3-D torus consisting of 12 bi-directional links with a bandwidth of 175 MB/s each in the X, Y and Z directions. The latency of a point-to-point message is a function of the distance traveled between nodes. The 32,768 processor Blue Gene/L used in this study consists of 16 racks with each rack being a 32x32x1 torus yielding a network of 32x32x16. The max distance is the sum of half the distance for each direction which is 40 (e.g., 16 + 16 + 8) leading to a max delay of 4 μ s (i.e., each hop has a max delay of 100 ns).

In addition to the point-to-point network, there is a global collective network that enables data collection, reduction and redistribution to all nodes (or a subset) with a latency of 5 μ s. We observe here that the collective network is able to compute a global reduction operation across *all processors* almost as fast as the single longest 1-way delay of the point-to-point network. This suggests that any GVT algorithm using the point-to-point network will not scale as well as using the collective network. The interface for sending data over the Blue Gene network is MPI.

For this experimental study a 32,768 processor Blue Gene/L located at the Rensselaer Computational Center for Nanotechnology Innovations (CCNI) is used. The IBM XLC C compiler was used for all the results in this paper. We were able to take full advantage of the compiler's peak optimization level as well as architecture specific settings. Our specific compiler options were: `-O5 -qarch=440d -qtune=440` for the Blue Gene/L.

3.2 The Cray XT5 Architecture

The Cray XT5 is also a scalable platform for supercomputing, and the best known installation, "Jaguar", at Oak Ridge National Labs is the world's fastest system for open science. Focused primarily on the number of floating point operations computed per second (FLOPS), the XT5 employs 2000 series AMD Opteron processors interconnected via the improved SeaStar2+ torus network. Our particular system contains 1,592 compute nodes, each with dual processor, quad-core 2.8GHz chips and 8GB of RAM. The batch queue is configured to allow a maximum node request of 256, for a total of 2,048 cores. The SeaStar2+ network provides 6GB/s of sustained bandwidth through 6 ports per compute node configured as a 3D torus. However, there are no other specialized networks for I/O, network barriers, or global collective operations as there are on the Blue Gene.

4 PERFORMANCE STUDY

In this section we perform a study of the parallel performance of the TLM model on the Cray XT5 versus the IBM Blue Gene/L. We define the metrics used for the study, then we present the results for TLM in *Rensselaer's Optimistic Simulation System* (ROSS) on the Cray and IBM machines. The ROSS code for either system is the same, and asynchronous MPI routines are used for sending/reading events.

4.1 Metrics

For the performance results we use three primary metrics to indicate the performance of the parallel discrete event simulation: *event rate*, *remote event rate* and *rollback efficiency*. The event rate measures the number of events processed per second

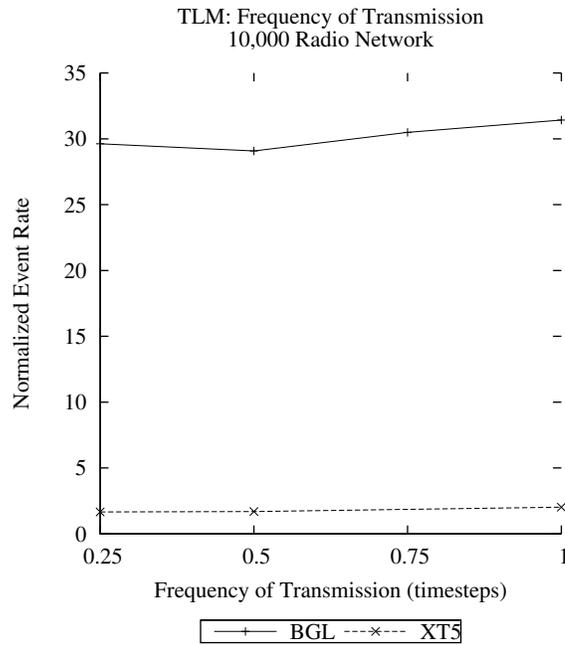


Figure 4: Normalized performance of TLM with a 10,000 radio network. At each timestep the probability a radio will transmit is computed according to a uniform distribution and threshold. At 1.0, all radios transmit at each timestep. 1,000 CPUs on the Cray XT5 and 2,000 CPUs on the Blue Gene/L were used and per MHz Event Rates are shown.

in the simulator, not including rollback events. The remote event rate measures the number of events transmitted over the network over the entire runtime of the model compared to the total number of events processed, not including rollback events.

Comparing performance between supercomputing platforms is challenging because each element of the architectures is unique. Lacking a simple metric for accurate comparison, our approach is to normalize event rates between platforms to clock speeds and processor core counts.

We normalize the event rates (ER) between platforms using the equation:

$$\frac{ER}{MHz} = \frac{ER}{(N_{cores} * MHz_{core})} \tag{1}$$

. This equations yields the event rate for each MHz of processing power applied to the model. As an example, the normalized equivalent to using 10,000 CPUs on the IBM Blue Gene/L is 3044 CPUs on the Cray XT5.

We also report runtime improvements as a ratio of simulated time to real-time in order to demonstrate much faster than real-time performance. In fact, most of the results reported in this paper are a fraction of real-time.

4.2 Scaling

Scaling the TLM model for a performance study is difficult. Weak scaling normally consists of increasing the model size as the number of processors is increased. The model contains 1 billion LPs representing the complex environment as a mesh. Increasing the number of radios, even to 1 million, represents an increase of 0.1% of the total LP count.

The primary dynamic we are interested in studying is the ability of the Time Warp protocol to manage a multiscale event population that is growing at a rate of $O(MN^3)$. The size of the event population grows linearly with respect to the number of radios modeled (M) and at a resolution of 1 second for mobility. The scattering of the electromagnetic waves in a 3D environment contributes the N^3 variable and have a resolution close to the speed of light. Therefore, we measure the scalability of TLM by increasing the size of the event population.

ROSS Variable	Value
GVT Interval	1024
BATCH SIZE	4
#KPs	100
Send Buffer	150,000
Read Buffer	150,000

Figure 5: Experimentally determined configuration settings for ROSS Time Warp simulator

4.3 Rensselaer's Optimistic Simulation System

The simulation package used in this study is *Rensselaer's Optimistic Simulation System (ROSS)* (Carothers et al. 2002). Table 5 outlines the most common configuration settings used for tuning ROSS based on earlier testing of the model using low-resolution terrain (Bauer and Page 2007b).

4.4 Frequency of Radio Transmission

To gain an initial understanding of the impact of the event population, our first analysis varies the frequency of radio transmissions for a network containing 10,000 radios. The probability that a radio transmits an electromagnetic (EM) wave at a given timestep is determined by a uniform distribution between 0..1 and a threshold of 0.25, 0.50, 0.75 and 1.0 is applied. If the draw exceeds the threshold, then no wave is transmitted.

We fix the number of CPUs used on the XT5 to 625 and on the Blue Gene to 2,000, yielding 1.4THz of computational power per platform. Radios change direction based on a uniform distribution over the range of 1..5 timesteps. With a threshold percentage of 1.0, radios transmit at each timestep in the simulation at a radio frequency of 3GHz. The end time is fixed at 600 seconds, or 10 minutes.

Figure 4 illustrates that as the frequency of radio transmission increases, the normalized event rates remain relatively flat on either platform. The remote event rate on either system was approximately 22%. Performance on the Blue Gene is more than 15 times faster than the XT5. Performance improves by over 10% on the Blue Gene and 30% on the XT5 as the event population increases. This behavior indicates that the Time Warp protocol is having little problem handling the cubic growth rate of the event population. While we were unable to directly measure the difference in performance, we believe the major factor impacting the XT5 is related to OS jitter. Limitations of the batch queue system on the XT5 limited us to using all cores per node, which suffer from competition with OS processes in the form of context switching.

4.5 IBM Blue Gene/L Scalability

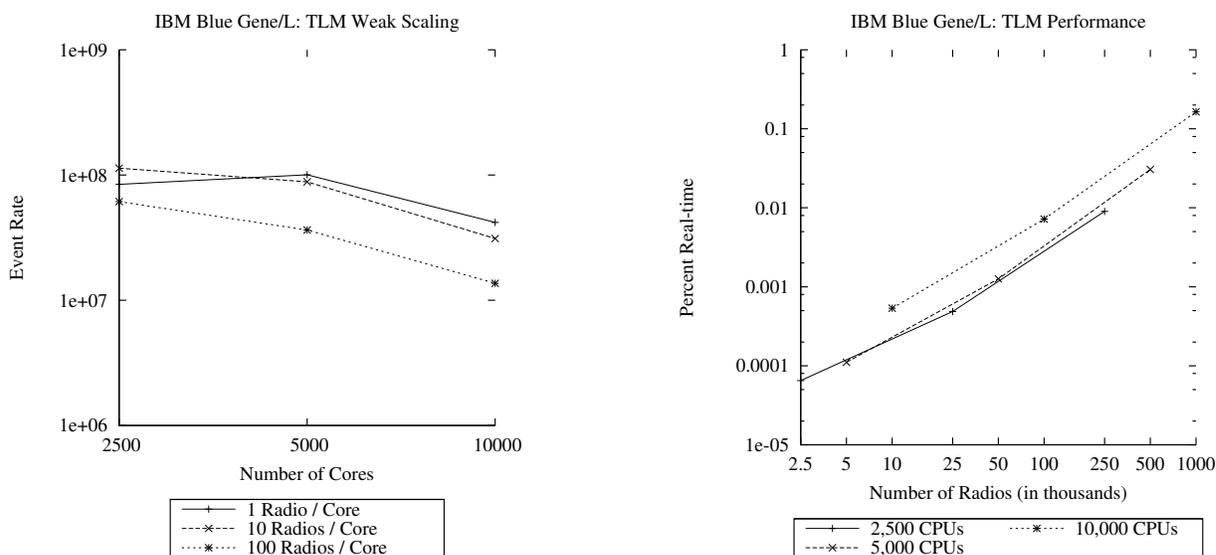
For simplicity of scalability analysis, we now fix the frequency of radio transmission at 100% so that every radio transmits at each timestep. Based on the parallel decomposition of the model we were able to utilize 2,500, 5,000 and 10,000 CPUs for our experiments, and we selected the number of radio transmitters as 1, 10 or 100 per processor used in the simulation, for a maximum of 1 million radios. Please note that a single wave starts as a single event in the spatial grid of LPs, rapidly expanding out in all three dimensions simultaneously. Therefore the event population grows according to the cubic volume of a sphere, deformed by the complex environment. A single transmission has the potential to generate millions of events, many with tied timestamps for when traveling through a homogeneous medium (such as free space). The TLM model generates a worst case scenario for PDES given the growth of the event population and the high number of tied event timestamps. The remote event rate for the 1 radio per CPU cases was generally 17%, and 25-30% for the 10 and 100 radio per CPU cases.

Figure 6(a) illustrates weak scaling on the IBM Blue Gene/L. As the event rate decreases for each configuration of radios to cores, the runtime is increasing, indicating that the amount of exploitable parallelism has been exhausted in the model as we approach 10,000 cores. At 10,000 cores, we have the maximum possible decomposition of the mesh grid in the X dimension, or 1 YZ plane per processor.

Runtime performance of the TLM model is shown in Figure 6(b) for the three processor configurations: 2,500, 5,000 and 10,000 CPUs. On the Blue Gene/L we were able to simulate 10,000 constantly transmitting radios 600 times faster than real time, and 1 million radios 6 times faster than real-time.

4.6 Cray XT5 Scalability

Based on the batch queue configuration of the XT5 we were only able to utilize 500 and 1,000 CPUs for these experiments. The 15-fold decrease in performance limited the number of radio transmitters to 1 or 10 per processor. In Figure 7(a), weak scaling on the XT5 dropped precipitously from 500 to 1,000 cores for the 1 radio per core case. Compared to the Blue Gene processors at 700 MHz, the power of 2.3 GHz processor on the XT5 was wasted. There simply was not enough work in



(a) Weak scaling the TLM model for a range of network sizes

(b) Runtime performance of the TLM model for a range of network sizes, up to 1 million mobile radios. Real-time is represented at 1.0 and both axes are plotted on a log scale.

Figure 6: Scalability and runtime performance of the IBM Blue Gene/L

the system to effectively utilize the system. At 10 and 100 radios per core, the XT5 fared better in terms of weak scaling, but as the runtime performance in Figure 7(b) shows, the results are almost 10 times greater than real-time.

The Cray appears to be suffering from an imbalance in the overall architecture. While further measurements would be required to identify the precise imbalance, overall the XT5 appears to be suffering from OS jitter as well as a mismatch between processor and memory access speeds. On the Blue Gene we are forced to use many processors, as each node contains only 256MB of RAM. On the Cray we have 2GB of RAM per core, allowing us to schedule 10 times as many LPs per PE and use smaller CPU configurations. We hypothesize that the performance of the Cray would improve if more processors could be employed, but due to limitations of the batch queue configuration, we were limited to 2,048 processors, and could realistically only schedule 1,024. Note that the improvement in performance would not be attributed to processor speed, but rather the minimizing of the amount of memory addressed per processor.

A second benefit of the Blue Gene over the Cray is the network configuration. Having a collective network for GVT separate from the toroidal network for events allows Time Warp to nearly completely hide the cost of the GVT computation. Also, the toroidal network can be optimized for small messages, which in the case of TLM are only 160 bytes on the 32-bit Blue Gene (208 bytes on the 64-bit Cray).

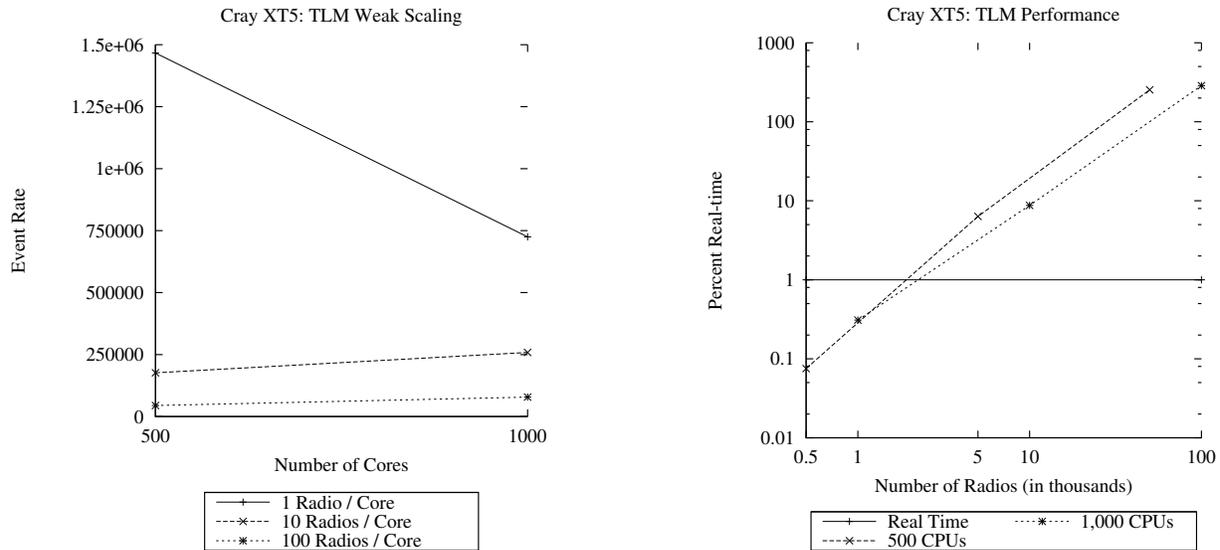
5 CONCLUSION

This paper demonstrates the capabilities of a Time Warp system that leverages the high-performance asynchronous message passing capabilities of the Cray XT5 and IBM Blue Gene/L family of supercomputers. We demonstrated that a bursty event application, the Transmission Line Matrix model for electromagnetic wave propagation, was able to efficiently scale to 10,000 processors on the IBM Blue Gene/L for networks of up to 1 million mobile radios in a complex terrain.

Overall, these performance results indicate that Time Warp is scalable for a worst case event scheduling behaviors, including multiscale timestamps and high frequency of tied timestamps. We have shown that the Blue Gene architecture appears to be the more scalable of the two platforms for parallel discrete event simulations. In particular, we show that the Cray XT5 architecture may be unbalanced in favor of FLOPS, and potentially suffer from OS jitter, rather than overall platform performance.

REFERENCES

- Adiga, N. R., G. S. Almasi, Y. Aridor, R. Barik, D. Beece, R. Bellofatto, G. Bhanot, R. Bickford, and M. Blumrich. 2002. An overview of the BlueGene/L supercomputer. In *Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, 1–22. Los Alamitos, CA: IEEE Computer Society Press.



(a) Weak scaling the TLM model for a range of network sizes

(b) Runtime performance of the TLM model for a range of network sizes, up to 10,000 mobile radios. Both axes are log scale.

Figure 7: Scalability and runtime performance of the Cray XT5

- Bauer, D. W., and E. H. Page. 2007a. An approach for incorporating rollback through perfectly reversible computation in a stream simulator. In *Proceedings of the Twenty-first Workshop on Principles of Advanced and Distributed Simulation*, 171–178. IEEE Computer Society Washington, DC, USA.
- Bauer, D. W., and E. H. Page. 2007b. Optimistic parallel discrete event simulation of the event-based transmission line matrix method. In *Proceedings of the 2007 Winter Simulation Conference*, ed. S. G. Henderson, B. Biller, M. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 676–684. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- 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* 62 (11): 1648–1669.
- Carothers, C. D., K. S. Perumalla, and R. M. Fujimoto. 1999. Efficient optimistic parallel simulations using reverse computation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 9 (3): 224–253.
- Chen, G., and B. K. Szymanski. 2003. Four types of lookback. In *Proceedings of the Seventeenth Workshop on Parallel and Distributed Simulation*, 3. Washington, DC, USA: IEEE Computer Society Washington, DC, USA: IEEE Computer Society.
- 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.
- Gara, A., M. Blumrich, D. Chen, G. Chiu, P. Coteus, M. Giampapa, R. Haring, P. Heidelberger, D. Hoenicke, G. Kopcsay et al. 2005. Overview of the Blue Gene/L system architecture. *IBM Journal of Research and Development* 49 (2-3): 195–212.
- Gomes, F. 1996. *Optimizing incremental state-saving and restoration*. Ph. D. thesis, University of Calgary.
- Jefferson, D. R. 1985. Virtual time. *ACM Trans. Program. Lang. Syst.* 7 (3): 404–425.
- Johns, P. 1974. 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.
- Kuruganti, T., and J. Nutaro. 2006. Validation radio wave propagation model. Technical report, Oak Ridge National Laboratory.
- Lamport, L. 1978. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM* 21 (7): 558–565.
- L’Ecuyer, P., and T. H. Andres. 1997. A random number generator based on the combination of four lcg’s. *Math. Comput. Simul.* 44 (1): 99–107.
- Li, L., and C. Tropper. 2004. Event reconstruction in time warp. In *Proceedings of the Eighteenth Workshop on Parallel and Distributed Simulation*, 37–44. New York, NY, USA: ACM New York, NY, USA: ACM Press.
- Lin, Y.-B., B. R. Preiss, W. M. Loucks, and E. D. Lazowska. 1993. Selecting the checkpoint interval in time warp simulation. In *Proceedings of the Seventh Workshop on Parallel and Distributed Simulation*, 3–10. New York, NY, USA: Association for Computing Machinery, Inc, One Astor Plaza, 1515 Broadway, New York, NY, 10036-5701, USA.: ACM Press.
- Nutaro, J. 2006. A discrete event method for wave simulation. *ACM Trans. Model. Comput. Simul.* 16 (2): 174–195.

- Quaglia, F. 1999. Fast-software-checkpointing in optimistic simulation: embedding state saving into the event routine instructions. In *Proceedings of the Thirteenth Workshop on Parallel and Distributed Simulation*, 118–125. Washington, DC, USA: IEEE Computer Society Washington, DC, USA: IEEE Computer Society.
- Steinman, J. S. 1993. Incremental State Saving in SPEEDES using C++. In *Proceedings of the 1993 Winter Simulation Conference*, ed. G. W. Evans, M. Mollaghasemi, E. C. Russell, and W. E. Biles, 687–696. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- West, D. 1988. Optimizing time warp: lazy rollback and lazy re-evaluation. Master's thesis, University of Calgary.
- Zeng, Y., W. Cai, and S. J. Turner. 2004. Batch based cancellation: a rollback optimal cancellation scheme in time warp simulations. In *Proceedings of the Eighteenth Workshop on Parallel and Distributed Simulation*, 78–86. New York, NY, USA: ACM Press.

AUTHOR BIOGRAPHIES

DAVID W. BAUER JR. is a Principle Research Scientist at The MITRE Corporation, working on a mobile ad-hoc network modeling and simulation for the Department of Defense. His Ph.D. is from The Rensselaer Polytechnic Institute in 2005. His research interests include large-scale modeling and simulation on supercomputing platforms. His email address for these proceedings is <bauerd@cs.rpi.edu>.

CHRISTOPHER D. CAROTHERS is an Associate Professor within the Department of Computer Science, Rensselaer Polytechnic Institute. His Ph.D. is from The Georgia Institute of Technology in 1997. His research interests include applied massively parallel simulation and large-scale network modeling. He is an Associate Editor for *ACM Transactions on Modeling and Computer Simulation*. His email address for these proceedings is <chrisc@cs.rpi.edu>.