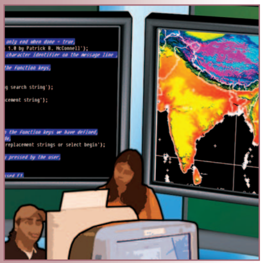

A Grid Service Module for Natural-Resource Managers

To facilitate transparent use of the high-performance Across Trophic-Level System Simulation (ATLSS) ecosystem-modeling package for natural-resource management, the authors developed a grid service module. The module exploits grid middleware functionality to process complex computation without requiring users to handle underlying issues. It represents the first application of grid computing to this discipline and provides a potential template for researchers in other disciplines to exploit scientific computation without extensive training in high-performance computing.

Dali Wang, Eric A. Carr, Mark R. Palmer, Michael W. Berry, and Louis J. Gross
University of Tennessee, Knoxville

During the past two decades, researchers have developed a variety of ecological models that serve as useful tools for natural-resource management. Those models can summarize information on a resource, determine where gaps exist, extrapolate across them, and simulate scenarios to evaluate the outcomes of various management decisions.[1] However, poor integration with existing models and the lack of new ones keep ecological models from being used effectively in natural-resource management.[2] Moreover, natural-resource managers typically have little experience or training in scientific computing. For new modeling applications to be effective, developers must therefore create new computational methodologies to let users perform complex simulations without extensive additional training.

As ecological models become more complex, their computational demands make it prudent to have tools that can carry out the simulations and associated visualizations. Fortunately, developments in high-performance networks, computers, and information services now make it feasible to incorporate remote computing and information resources into local computational environments. Specifically, grid computing has emerged as one of the most important new developments in building the computational science infrastructure. (For more on the fundamental concepts behind and current applications of grid computing, see www.gridcomputing.com.)

With these issues in mind, we developed a grid service module to deliver high-performance ecosystem-modeling capabilities to remote natural-resource managers through the Across Trophic-Level System Simulation modeling package (www.atlss.org) developed at the

 Published by the IEEE Computer Society 1089-7801/05/$20.00

University of Tennessee. ATLSS is designed to assess the effects of alternative management plans for regulating water flow across the Everglades landscape. Our immediate objective was to provide quantitative, projective modeling software for guiding the South and Central Florida restoration effort. We developed the module described here specifically to support the use of ATLSS models by resource-management professionals in the 12 government and private agencies involved in restoration assessment for the Everglades, as well as by academic researchers assisting in the restoration planning. Indeed, we designed the interface and output formats in collaboration with users from these resource agencies. Our long-term goals are to aid in understanding how South Florida's biotic communities are linked to the hydrologic regime and other abiotic factors, and to provide a projective tool for scientific research and ecosystem management.

Computational Platform and Grid Middleware

We built the grid service module on top of the Scalable Intracampus Research Grid (SInRG; http://icl.cs.utk.edu/sinrg/), a computational platform supported by the US National Science Foundation. SInRG is a research infrastructure at the University of Tennessee, Knoxville; it mirrors the underlying technologies and interdisciplinary research collaborations that are characteristic of the emerging national technology grid. SInRG's primary purpose is to provide a technological and organizational microcosm in which researchers can address key challenges underlying grid-based computing with better communication and control than wide-area environments usually permit.

Foster and colleagues define a grid service as "a (potentially transient) stateful service instance supporting reliable and secure invocation (when required), lifetime management, notification, policy management, credential management, and virtualization."[3] All of these properties are consistent with our implementation, which we built around two main grid middleware products: Netsolve (http://icl.cs.utk.edu/netsolve/) and the Internet Backplane Protocol (IBP; http://loci.cs.utk.edu/modules.php?name=Projects&pa=showpage&pid=3_1). We chose these to enhance the grid service module's seamless integration with the underlying C++ code for the ATLSS models.

NetSolve is an RPC-based middleware system that lets users access hardware and software resources remotely. It includes three main components: agent, servers, and remote users. NetSolve tracks which machines have computational servers running and what computational services (software) they're provisioned with. It also tracks each NetSolve server's workload to locate the best choice for a given job request. In other words, NetSolve takes care of the details in finding machines on which to execute computational tasks. NetSolve also provides extensible service creation via an Interface Definition Language (IDL) facility — the NetSolve Problem-Description File (NPDF) — to generate wrappers for the user's software (ecological models, in our case). After compilation, the users codes thus become NetSolve services that server instances can enable. Netsolve also implements the GridRPC programming model,[4] consistent with Global Grid Forum standards (www.gridforum.org). This allows it to serve as a seamless bind mechanism to major Grid back ends — particularly Globus (www.globus.org) and Condor (www.cs.wisc.edu/condor) — and ensures easy access to key grid services, such as the Globus Security Infrastructure (www.globus.org/security/overview.html) and the Network Weather Service (http://nws.cs.ucsb.edu).

IBP is middleware for managing and using remote storage. It supports global scheduling and optimizes data movement, storage, and computation using a model that takes into account all of the network's underlying physical resources. This contrasts with most traditional networking, which doesn't explicitly model storage or computational resources in the network. An IBP-based system can separate large data files into multiple parts for storage on different IBP servers by creating a small ASCII-based file, called an *exnode*, for each data-storage location. IBP-based systems can also store multiple copies of individual files. Based on this kind of configuration, IBP provides a mechanism for using distributed storage for logistical purposes.

Design and Implementation

Figure 1 (next page) presents a simplified view of the grid service module's four major components:

- A dedicated *Web interface* provides a common gateway through which multiple users can specify simulation inputs and launch multiple tasks at the same time; it also checks user identifications and performs process authorization.

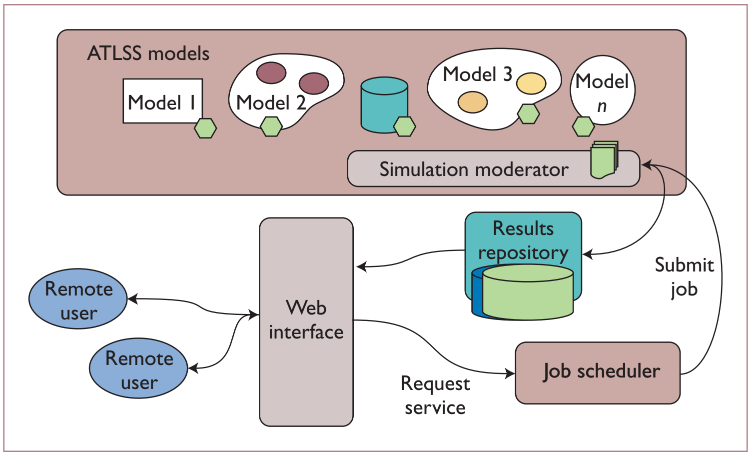

Figure 1. Grid service module architecture. The job scheduler receives a user's computation request through the Web interface and allocates appropriate Across Trophic-Level System Simulation (ATLSS) models in the simulation moderator for executions. After a simulation is complete, the Web interface notifies the user by email that simulation results are available from the result repository.

- The *job scheduler* contains a NetSolve agent, which allocates appropriate ATLSS models for (remote) job executions in response to users' computation requests received via the Web interface; the scheduler also contains an IBP client for efficient data transfer, because ATLSS model simulations usually require large data movement.
- The *simulation moderator* is built on NetSolve servers, on which the ATLSS models are configured as services, and IBP servers, which manage the models' input and output.
- The result repository is a database that stores simulation results for authorized users to review and reuse in future analysis.

Three features distinguish our grid service module from other grid services: a customized Web interface for the ATLSS modeling package, unique computational resource allocation and remote execution using NetSolve, and efficient network-based storage and data transmission through IBP. Depending on actual model requirements, users can use the module to simultaneously launch several parallel ATLSS simulations and a bundle of sequential ecological simulations on the computing grid (SInRG).

Three US federal government agencies in South Florida and several research groups at the University of Tennessee are currently using the grid-enabled ATLSS modeling package via our grid service module. As more ATLSS models become grid-enabled, we expect to further increase modeling activities in the near future.

Web Interface

Various stakeholders and agencies have expressed strong interest in being able to access, run, and retrieve data from multiple ATLSS ecological models through a single Web interface. Yet, the detailed information and model parameterizations associated with complex ecological models necessitate that the Web interface provide natural-resource managers access only to limited functionality.

Figure 2 illustrates the password-protected Web interface for accessing ATLSS models. To run ATLSS models via our grid module, users need only Internet access and a Web browser. Natural-resource managers can choose from several models, which they can parameterize as needed (within certain limits). The module provides users an intuitive process through which they can apply ecological models for particular species, conditions, or spatial domains.

Each model lets users vary simple simulation-control parameters (such as simulation time) and input conditions (such as hydrological scenarios). Through this interface, users can launch both sequential and parallel jobs on SInRG resources. Once a simulation is complete (a process that might take hours of CPU time on a high-performance computer[5,6]), the Web interface notifies the user via email. In addition, the interface acts as the gateway for users to access result repositories for data from previous simulations.

Computational Resource Allocation and Remote Execution

As Figure 3 illustrates, the grid service module's resource-allocation and remote-execution functionality is based on NetSolve: the job scheduler operates via a NetSolve agent; the simulation moderator incorporates functions from NetSolve servers; and the ecological models are preconfigured as NetSolve services at compile time through NPDFs.

When NetSolve servers register themselves and their services with the NetSolve agent, the job scheduler collects all the necessary information about the ATLSS simulation's capabilities. To better utilize grid-computational resources, we can install an ecological model on multiple NetSolve servers, taking different architectural features into account — for example, we install a Pthread version of the ATLSS Landscape Fish Model (ALFISH) on a symmetric multiprocessor (SMP) and a message-passing-interface (MPI) version of ALFISH on clusters, as shown in Figure 3. Depending on the code implementation, we can also compile the ecological model on Windows, Macintosh, and Unix-based systems and report those models, through the NetSolve server's registration procedure, to the NetSolve agent within the job scheduler. Thus, the grid service module supports heterogeneous computing by providing the ability to harness diverse machine architectures to work together on a single computational task.

Once the job scheduler accepts a user's job request through the Web interface, its NetSolve agent looks at service availability and machine workload to find the best NetSolve server within the simulation moderator. It then ships the job request to the NetSolve server for processing. The simulation moderator contains a set of *problem-definition scripts* (PDS) that initialize the computational environment, prepare the model input, and launch the model on the computer on which the NetSolve service resides. If a NetSolve service had to handle an MPI-based model,[7] for example, a PDS would initialize the MPI environment and determine the number of processes for parallel execution.

This framework provides several advantages. First, it balances computational workload across the entire NetSolve organization because all jobs are scheduled through the centralized, intelligent job scheduler. It also enhances system security by insulating users from actual software and data and from direct access to the high-performance computational facilities. Finally, it lets users take advantage of high-performance computing without extensive knowledge of computational science.

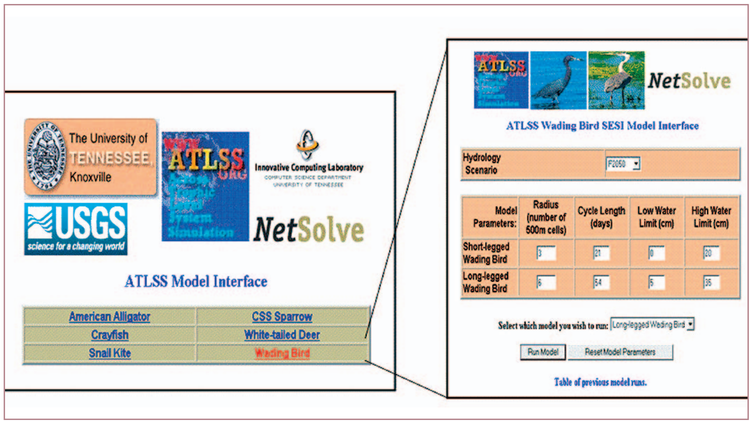

Figure 2. Web interface for ATLSS. Users can access ecosystem models and control parameters in the ATLSS modeling package through a single, uniform Web interface.

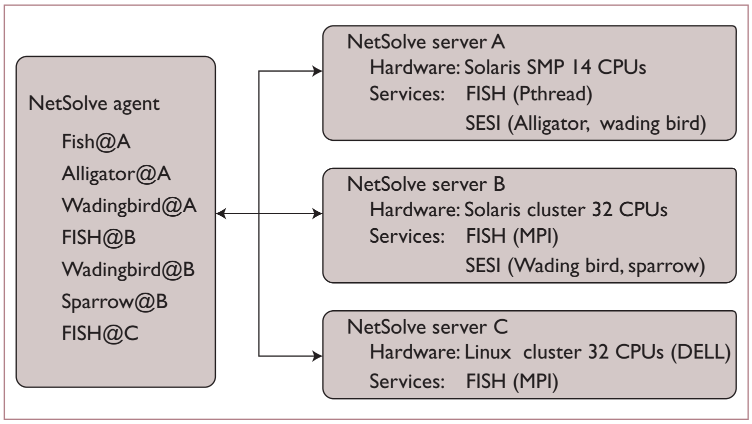

Figure 3. ATLSS installation in NetSolve organization. A NetSolve agent within the grid service module in the left-hand box distributes a set of ATLSS models among SInRG servers based on available resources: it schedules the Pthread Fish model on the symmetric multiprocessor (NetSolve server A) due to the model's requirement for shared memory computing, while scheduling other models on clusters (NetSolve servers B and C) to utilize distributed-memory computing.

Network-Based Storage and Data Transmission

NetSolve's limited support for transporting large amounts of data over the network presents some difficulty for ecological modeling. For this reason, we adopted IBP to provide efficient data-transport capabilities. Our grid service module uses IBP exnodes as *transfer keys* through the NetSolve system to establish a novel way to pass large files to and from remote computational facilities without direct user access. Thus, IBP allows the simulation moderator to allocate and schedule storage resources as part of its resource brokering, thus leading to improved performance and enabling fault-tolerance when resources fail or are revoked.

Figure 4 (next page) shows the typical data flow in the grid service module. To illustrate, we use a spatially explicit species index (SESI) model[8] drawn from a family of index models that have been developed as part of the Everglades restoration to provide a basis for quantifying the effects of different water-regulation plans on various species. The SESI for long-legged wading birds (SESIWB) includes as input a landscape map of South Florida at a 500-meter scale resolution, two hydrological scenarios over several decades, and a set of control parameters that specify model assumptions regarding the spatial pattern of wading birds' foraging rules over the landscape. The

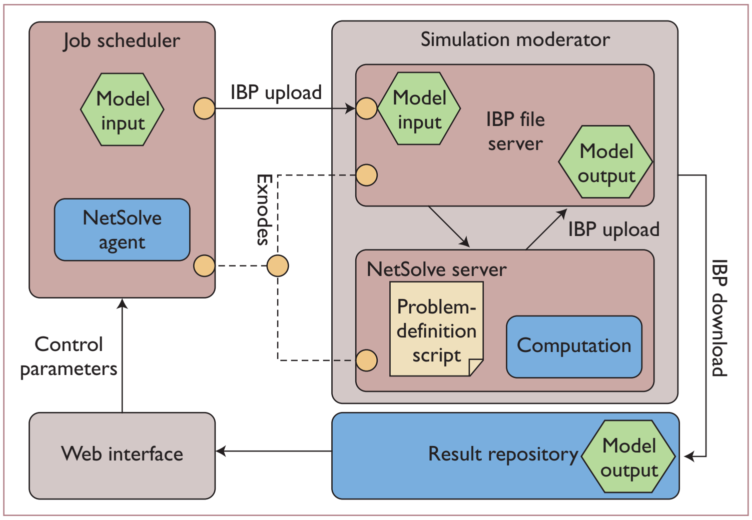

Figure 4. Data flow for an ecological model in the grid service module. Model-input data flows from the job scheduler to the simulation moderator, and model-output data flows from the simulation moderator to the result repository. IBP exnodes (as transfer keys) flow from the job scheduler to both IBP servers and NetSolve servers managed by the simulation moderator.

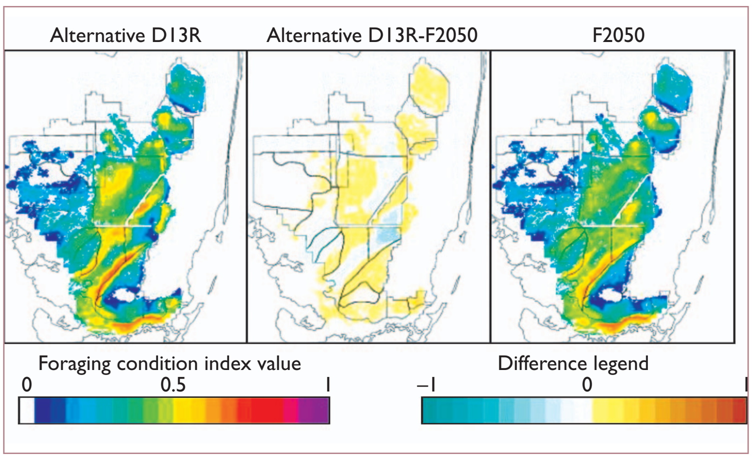

Figure 5. Example model output of ATLSS model for long-legged wading birds. Left and right graphics represent the mean index map of foraging condition for wading birds over the period of two hydrological scenarios, AltD13 and F2050, respectively. The middle graph represents the difference of the two index maps.

total size of input files in this scenario is about 0.5 Gbytes.

Once a natural-resource manager inputs control parameters (such as scenario name and simulation time) and submits a job request, the Web interface sends these parameters to the job scheduler, which then:

- assembles all model input data (landscape map, water-depth distribution for 35 years, and so on);
- determines data-storage locations (represented by one exnode for model input and one for model output) and computational facilities (represented by a NetSolve server);
- launches an `IBP_upload` operation to move the model input into an IBP file server monitored by the simulation moderator; and
- passes the exnode information to the simulation moderator via the connection between its NetSolve agent and the remote NetSolve server.

After receiving the job request from the scheduler, the simulation moderator uses a PDS to download the model input from the IBP file server, initialize the computational environment, launch the computation, and upload the model result back to the IBP file sever. Once the computations are complete, the simulation moderator notifies the job scheduler, issues an `IBP_download` operation to deliver the model output to the result repository, and notifies the user via email.

Figure 5 shows an example of the SESIWB model's output, including a visual representation of the landscape with color-coded values assigned to each cell, which present the mean index over the whole simulation under each hydrologic scenario. The index value (ranging from 0 to 1) reflects the relative potential for appropriate foraging conditions.

To our knowledge, our work with ATLSS represents the first time anyone has applied a computational grid to a natural-resource management problem. Projects such as ours offer resource managers the ability to apply the most scientifically defensible models, even when they involve intensive computation. Spatially explicit and temporally varying models present numerous computational challenges as they realistically account for what we know of environmental variation and its impacts on natural systems. We expect that grid service modules will provide feasible methods for addressing these problems and providing input to the decision-support tools used in natural resource management.

From a broader perspective, grid service modules could positively impact applied and scientific computation problems for two major audiences:

- For model developers, they provide a practical, explicit approach for utilizing remote high-per-

Related Work in Natural-Resource Management

Researchers have successfully applied advanced computational technologies to several complicated natural-resource management problems. Currently, however, all such approaches are based on distributed and parallel computing technologies. The LifeMapper project (www.lifemapper.org), for example, uses the Genetic Algorithm for Rule-set Production (GARP)[1] to analyze biodiversity. LifeMapper uses an inherently static, geographic information system (GIS) methodology, utilizing disparate data sets to make maps of species' ranges.

Several projects have employed parallel computation.[2] For example, the Institute for Environmental Modeling (www.tiem.utk.edu) developed parallelizations of a spatially and size-structured population model of freshwater fish across the South Florida landscape, as well as linked individual-based models for predators and prey.[3]

The expanded use of distributed and parallel computing for natural-resource modeling and management presents a variety of difficult issues. Although distributed computing is largely transparent and intuitive, most traditional distributed applications require highly inherent computational independence — that is, little or no communication is allowed between separate data sets. On the other hand, parallel computation is effective for solving complex natural-resource problems, but it can be very difficult for naive users (the public, natural resource managers, and so on) to take advantage of such highly developed computer models and software.

We developed our grid service module for the Across Trophic-Level System Simulation (ATLSS) ecosystem-modeling package to stimulate the adoption of novel computational technologies in ecological problem solving and analysis. The grid-based approach enables more extensive experimentation by accelerating compute-bound activities. We also wanted to create new ways to handle and process the increasingly large amount of information concerning natural-resource (ecological) management. Our objective is to establish a new avenue for natural-resource managers to facilitate high-performance collaborative modeling. We are working to build an infrastructure to promote dynamic modeling research on a computing grid to allow ecologists and natural-resource managers to tackle important problems, which are currently limited by existing computational environments.

References

1. D. Stockwell et al., "The GARP Modeling System: Problems and Solutions to Automated Spatial Prediction," *Int'l J' Geographical Information Science*, vol. 13, no. 2, 1999, pp. 143–158.
2. A. Immanuel et al., "A Parallel Implementation of ALFISH: Compartmentalization Effects on Fish Dynamics in the Florida Everglades," to appear in *Simulation Practice and Theory*, 2004.
3. L.E. Mellott et al., "The Design and Implementation of an Individual-Based Predator-Prey Model for a Distributed Computing Environment," *Simulation Theory and Practice*, vol 7, no. 1, 1999, pp. 47-40.

formance infrastructure and existing simulation packages (without code modification).

- For decision-makers and stakeholders, they create an intuitive method for launching and analyzing model results without concern for the underlying implementations.

The procedures outlined through our grid service module could provide a template for a wide variety of natural resource management problems, including biodiversity assessment, harvest management, and hydrologic controls. Developers could link software repositories for models such as ATLSS with distributed ecological and geographic databases, such as those available through the US National Biological Information Infrastructure (www.nbii.gov).

Currently, the Web interface and underlying grid service modules host nine different ecological models of varying complexity and interconnection. Work is under way to develop a greater complexity of linkages between models and to apply optimal spatial control, allowing resource managers to investigate alternative spatially explicit methods for managing natural systems. Both efforts will increase our utilization of ATLSS. Grid-based postprocessing and data-visualization creation also provide areas for further expanding the application. The computational grid could analyze comparisons between different individual and multimodels and present the results in various output formats as requested by the user. Such interactions demand further developments of the grid service module to facilitate continuous decision-making processes for resource managers.

Acknowledgments

This research is supported by the US National Science Foundation under grant number DEB-0219269. This research used the resources of the Scalable Intracampus Research Grid (SInRG) project at the University of Tennessee, which is supported by NSF's CISE Research Infrastructure Award EIA-9972889. The authors appreciate support for ATLSS model development from the US Geological Survey, through cooperative agreement number 1445-CA09-95-0094 with the University of Tennessee, and the US Department of the Interior's Critical Ecosystem Studies Initiative.

References

1. V.H. Dale, "Opportunities for Using Ecological Models for Resource Management," *Ecological Modeling for Resource Management*, V.H. Dale, ed., Springer-Verlag, 2003, pp. 3–22.
2. L. Ginzburg and H.R. Akcakaya, "Science and Management Investments Needed to Enhance the Use of Ecological Modeling in Decision Making," *Ecological Modeling for Resource Management*, V.H. Dale, ed., Springer-Verlag, 2003, pp. 249–262.
3. I. Foster et al., "The Physiology of the Grid," *Grid Computing: Making the Global Infrastructure a Reality*, F. Berman, G. Fox, and A.J.G. Hey, eds., John Wiley & Sons, 2003, pp. 217–250.
4. K. Seymour et al., "Overview of GridRPC: A Remote Procedure Call API for Grid Computing," *Proc. Grid 2002*, LNCS 2536, Springer-Verlag, 2002, pp. 274–278.
5. D. Wang et al., "Parallel Landscape Fish Model for South Florida Ecosystem Simulation," *Proc. ACM/IEEE Supercomputing Conference* (SC 03), IEEE CS Press, 2003; www.sc-conference.org/sc2003/inter_cal/inter_cal_detail.php?eventid=10790#11.
6. D. Wang et al., "Design and Implementation of a Parallel Fish Model for South Florida," *Proc. 37th Hawaii Int'l Conf. System Sciences* (HICSS), IEEE CS Press, 2004; http://csdl.computer.org/comp/proceedings/hicss/2004/2056/09/205690282c.pdf.
7. *MPI: A Message-Passing Interface Standard*, Message Passing Interface Forum standard, June 1995; www-unix.mcs.anl.gov/mpi/.
8. J.L. Curnutt et al., "Landscape-Based Spatially Explicit Species Index Models for Everglades Restoration," *Ecological Applications*, vol. 10, no.6, 2000, pp.1849–1860.

Dali Wang is a research assistant professor in the Department of Computer Science and a research associate in the Institute for Environmental Modeling at the University of Tennessee, Knoxville. His research interests include parallel and scientific computing, high-performance system applications and evaluation, ecological and environmental modeling, and computational science. Wang received a PhD in environmental engineering from Rensselaer Polytechnic Institute. He has published more than 20 refereed journal and conference publications. Contact him at wang@tiem.utk.edu.

Eric A. Carr is a senior systems programmer in the Institute for Environmental Modeling at the University of Tennessee, Knoxville. His research interests include ecological and environmental modeling. Carr received an MS in mathematics from the University of Tennessee. He is a member of the Society for Mathematical Biology. Contact him at carr@tiem.utk.edu.

Mark R. Palmer is a senior systems/programmer analyst in the Institute for Environmental Modeling at the University of Tennessee, Knoxville. His research interests include artificial intelligence, operations, database design, environmental restoration, and environmental modeling. Palmer received an MS in computer science from the University of Tennessee, Knoxvile. Contact him at palmer@tiem.utk.edu.

Michael W. Berry is a full professor in and interim department head of the Department of Computer Science at the University of Tennessee, Knoxville. His research interests include parallel and scientific computing, information retrieval, bioinformatics, and computational science. Berry received a PhD in computer science from the University of Illinois, Urbana-Champaign. He is the coauthor and editor of three recently published books on data mining and text retrieval. He is a member of SIAM, the ACM, and the IEEE Computer Society, and is on the editorial board of *Computing in Science and Engineering* magazine and the *SIAM Journal of Scientific Computing*. Contact him at berry@cs.utk.edu.

Louis J. Gross is a professor of ecology and evolutionary biology and mathematics and director of the Institute for Environmental Modeling at the University of Tennessee, Knoxville. His research in computational ecology involves developing multimodeling methods for explicitly linking abiotic and biotic components of natural systems and developing associated spatial-control methods to assist in natural-resource management. Gross received a PhD in applied mathematics from Cornell University. He is currently president of the Society for Mathematical Biology. Contact him at gross@tiem.utk.edu.