Toward Ecosystem Modeling on Computing Grids

Grid-based ecosystem modeling holds great promise for aiding the investigation of complex environmental systems. A prototype framework and generic software architecture provide increased interoperability and productivity for spatially explicit ecosystem modeling on heterogeneous grids.

Ecosystem modeling presents a variety of challenges. Much of classical ecological theory originates from very simple differential equations in which a single variable represents population densities. Researchers typically analyze the solutions mathematically and then compare them to abundance estimates from field or lab observations. Although highly influential in ecological theory, the models' aggregated form is particularly difficult to relate to observational biology. Applying them to complex natural systems with spatially and temporally varying environmental factors, for example, typically produces analytically intractable models that must be investigated numerically.

Recently, researchers have begun emphasizing integrated, multicomponent ecosystem models. These models involve complex interactions between some or all of an ecosystem's trophic layers (feeding levels), resulting in models that link multiple components that researchers can model using several different mathematical approaches. One implication of these efforts is that monolithic software development within a traditional computing framework is hindering further sustained innovation in complex, highly integrated model simulation. Traditional ecosystem modeling is sequential and not spatially explicit. Grid computing is a natural means to address the spatially explicit requirements of ecosystem models with multiple components. Researchers use different grid service modules[1] for different ecosystem model components, based upon the most efficient partitioning of the system's spatial domain.

Herein, we present a conceptual model for ecosystem modeling, and use Across Trophic Level System Simulation (ATLSS; see www.atlss.org) as an example to explain the key design considerations and efforts for such spatially explicit modeling on a computing grid at the University of Tennessee.

Ecosystem Modeling

To model an ecosystem, we first design a conceptual model that aggregates our knowledge of that system. Such a model requires that we select the system's essential components and processes for study in a given spatial–temporal context.

For example, a simple conceptual model might consider a linear food chain, modeling a series of organisms in an energetic hierarchy based on their trophic relationships. Researchers might use dif-

1521-9615/05/$20.00 © 2005 IEEE
Copublished by the IEEE CS and the AIP

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

ferent modeling approaches for organisms at different trophic positions. At a lower trophic level, they typically place more emphasis on the kinetics of the ecosystem's energy or nutrient flow, or the pollutant transport within the food web. At a higher level—especially for endangered species with small populations—researchers might need to monitor and simulate each individual member's basic behaviors.

Basic Modeling Approaches

There are three basic modeling approaches in ecological modeling: individual-based models, structured models, and compartment models.

Individual-based models are simulations that indicate the global consequences of local interactions among population members.[2] These models, also called agent-based models, typically consist of an environment and some number of interacting individuals defined in terms of their behaviors (procedural rules) and characteristic parameters. In an individual-based model, each individual's characteristics are tracked through time and space.

Structured models average certain population characteristics and attempt to simulate changes in these characteristics for the whole population. Rather than following every individual member, structured models use a set of structured classes to model a population. The population's structure is determined by a set of age or size classes, and often framed as a matrix model—as is typical in human demography. The size of each structured class, along with the model's time step, influences the model's parameter values and affects the transition rates between classes.

Compartment models typically describe the kinetics of several highly aggregated components in an ecosystem, expressed as a system of difference or differential equations. Researchers might use these equations for entire trophic levels (producers, consumers, and so on), unstructured populations, or components such as nutrients or energy—wherein they'd follow the flow of nutrients or energy across the different trophic levels.

An Integrated, Multimodeling Framework

To better address ecosystem modeling complexity, we adopted an ecological multimodeling approach, which was first proposed by Louis Gross and Donald DeAngelis[3] during the development of ATLSS. ATLSS is an ecosystem modeling package designed to assess how alternative water management plans for regulating water flow across the Florida Everglades' landscape will affect key biota. ATLSS's immediate objective is to assist various stakeholders in assessing the biotic impacts of alternative future

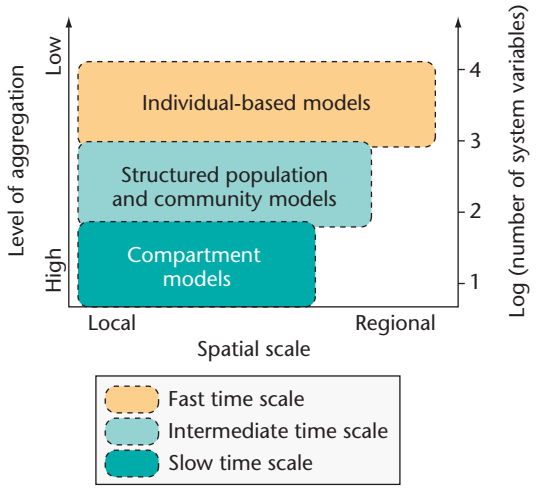

Figure 1. The Across Trophic Level System Simulations for South Florida. The ATLSS multimodeling framework helps stakeholders assess how future restoration scenarios for South Florida's natural systems will impact key biota.

scenarios for restoring South Florida's natural systems. The long-term goals are to aid in understanding how the biotic communities of South Florida are linked to various physical driving influences—particularly hydrology—and to provide a predictive tool for both scientific research and ecosystem management.

As Figure 1 shows, the hierarchical spatial pattern of ATLSS's dynamic components arises from associating different model types with different ecosystem species or groups of species. The compartment models deal with variables representing spatially localized biota—mainly the biomasses of lower trophic-level organisms, such as algae, which only interact locally. Given this, the researchers represent these variables across a landscape using many local, uncoupled spatial-unit-cell models. Specifically, they chose a cell size that was small enough to represent a tract with relatively homogeneous substrate and elevation, which might be several hundreds of meters in a relatively flat landscape such as the Everglades.

The age- and size-structured population and community models represent intermediate trophic levels, such as fish, macroinvertebrates, and small nonflying vertebrates. This population might undergo short-distance movements in response to water-level changes. Their spatial interaction domains are larger (up to a square kilometer) and thus potentially encompass many smaller unit cells, which are coupled to allow population movements.

Finally, individual-based models represent pop-

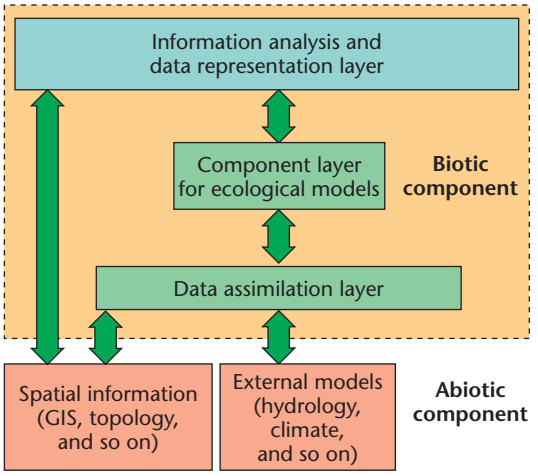

Figure 2. A software architecture for ecosystem modeling. The data assimilation layer provides necessary abotic inputs for spatially explicit ecological modeling. Simulation result are analyzed and visualized by the information analysis and data representation layers.

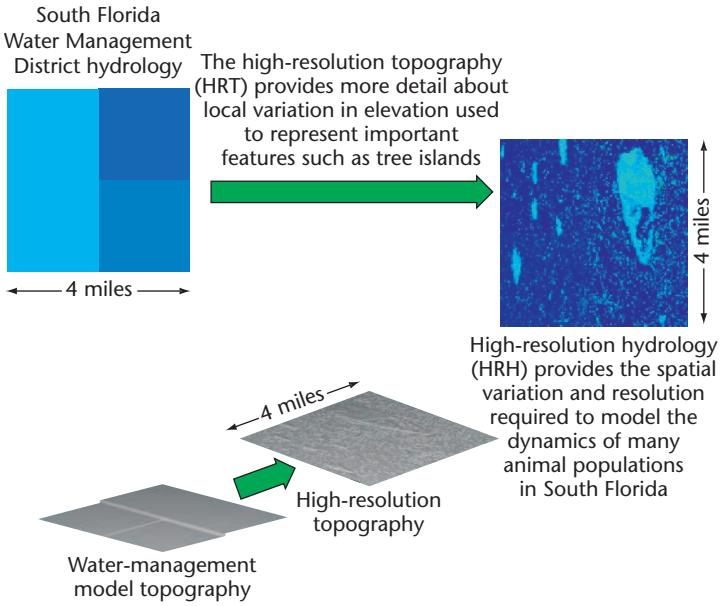

Figure 3. Spatial information processing in ATLSS. High-resolution topography was used in the water-volume conservative reproduction of high resolution hydrological data from the South Florida Water Management District Model.

ulations of top predators and other large-bodied species, such as wading birds and panthers. Individuals of these species might roam over large areas, with movements over short time periods spanning thousands of spatial unit cells. These individual-based models are rule-based approaches that can track the growth, movement, and reproduction of many thousands of individuals across the landscape. Adequate description of a single individual can easily require many variables, detailing its current age, size, location, physiological status, and other information pertinent to future actions. The spatial and temporal scales for different model components are interrelated in the ATLSS multimodel.

Figure 1's vertical axis shows the number of state variables or aggregation level required to describe the system.

A Multilayered Software Architecture

Our general, multilayered architecture for ecosystem modeling includes a data assimilation layer, a components layer for ecological models, and an information analysis and data representation layer (see Figure 2).

The *data assimilation layer's* main functionalities are to

- provide a uniform structure to integrate spatial information and information about physical data (such as hydrology) from external models;
- generate computational domains (meshes) for different ecological models; and
- determine parameter estimates for ecological models.

Figure 3 shows the procedure to generate a high-resolution, geo-referenced hydrological data.[4] To simulate animal population dynamics in South Florida, we need higher resolution hydrological data (such as at 100 meters), while the original hydrological data, created by the South Florida Water Management District Model, is at a resolution of 2 square miles. We therefore created a high-resolution topography map (at 30 meters) using satellite images of vegetation and information about the hydrologic ranges at which each type of vegetation can exist. From this, we developed a high-resolution hydrology map by redistributing the conservative water volume within each 2 square mile area.

The *components layer* contains different kinds of ecological models. Figure 4 shows some of the possible models in ATLSS. Based on the ecological relationships, users might link several models together—the Lower Trophic-Level Model, Fish Functional Group Model, and Wading Birds Model, for example, comprise a tightly linked model group.

The *information analysis and data representation layer's* main functions are to

- use various tools to analyze data, such as per-

forming different spatio–temporal averages;
- provide a specific data format for data visualization; and
- provide a standard data format for exporting data to standard Geographic Information System (GIS) software.

Figure 5 shows an example of one of ATLSS's structured population models. The ATLSS Landscape Fish Model (Alfish)[5] contains a set of habitat rules and appropriate hydrologic conditions for fish growth based on field biologists' observations and experience. As with all ATLSS models, Alfish takes various information—such as hydrological data, Florida vegetation information (from the Gap Analysis Program map)—as input and produces estimates of fish abundance that natural resource managers can use to help assess the impact of different hydrologic plans.

Integrated, spatially explicit ecosystem modeling generally requires intensive, parallel computations. For example, a single ecological model in ATLSS for a single hydrological scenario easily consumes hours of elapsed CPU time on a standard high-performance symmetric multiprocessor, or even a high-end computer.[6,7] To expedite integrated ecosystem modeling, we deployed grid computing to deliver advanced simulation functionalities to naïve computer users (biologists, natural resource managers, and so on).

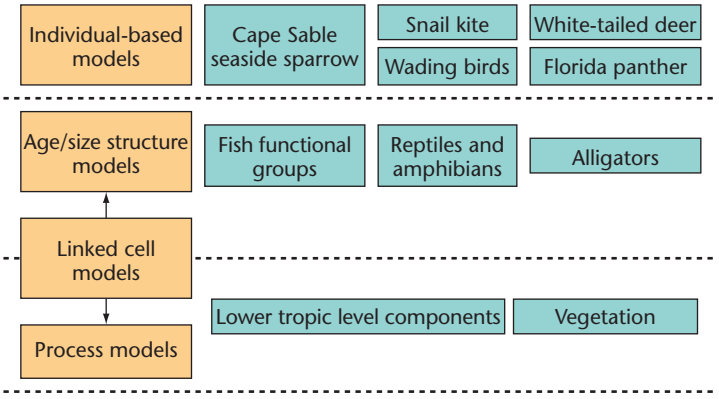

Figure 4. Ecological models in ATLSS. Users can link models together based on ecological relationships.

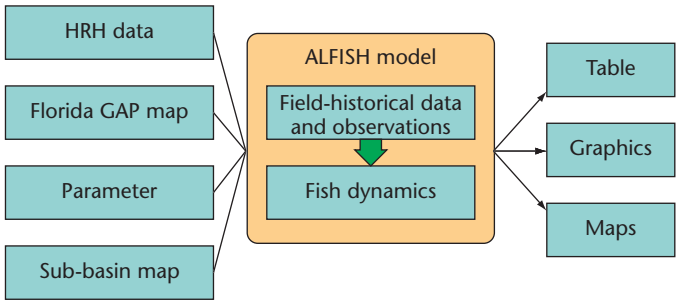

Figure 5. Information analysis in ATLSS. The Landscape Fish Model contains a set of habitat rules and appropriate hydrologic conditions for fish growth based on field biologists' observations and experience.

Grid-Based Ecosystem Modeling

The Grid has emerged as a key development in building the computational science infrastructure. By integrating networking, communication, computation, and information, the Grid provides a virtual platform for computation and data management—just as the Internet provides a virtual platform for information access. Using the Grid, users can access remote computers and employ networked resources for the computational challenges of large-scale ecosystem modeling.

With the support of the National Science Foundation, the University of Tennessee established an Intracampus grid to provide researchers the capability to develop new computational science methods in ecology, medicine, engineering, and materials science. The Scalable Intracampus Research Grid project (SInRG; http://icl.cs.utk.edu/sinrg) deploys a research infrastructure that mirrors the underlying technologies and interdisciplinary research collaborations characteristic of the emerging national technology grid. SInRG's primary purpose is to provide a technological and organizational microcosm in which the key research challenges underlying grid-based computing can be attacked with better communication and control than wide-area environments usually allow. Knowledge acquired in this smaller environment can then help improve the national grid infrastructure.

Figure 6 shows the SInRG skeleton. The architecture's primary building block is the grid service cluster (GSC). A GSC is an ensemble of hardware and software that a single workgroup builds, optimizes, and administers so that its resources are easily available for use by its own grid-enabled applications and those of others in the network. As Figure 6 shows, each GSC is first built around an advanced data switch (at least 1Gbits per link) to provide high-level quality of service within the cluster itself. Each GSC contains a large data storage unit attached directly to its switch to facilitate—for both local and remote SInRG users—remote data localization for efficient GSC processing.

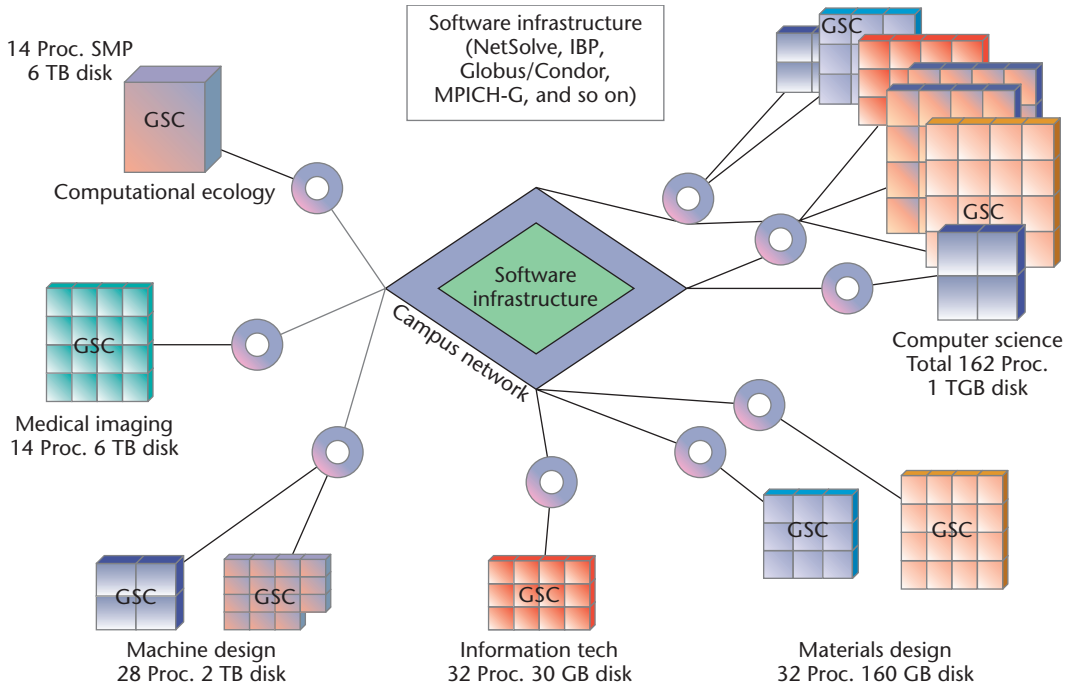

Figure 6. The Scalable Intracampus Research Grid. SInRG is built around grid service clusters, which individual workgroups build, optimize, and administer to service grid-enabled applications throughout the grid.

As the figure shows, GSCs are typically commodity clusters. Different SInRG research projects, however, require customized GSCs, such as a symmetric multi-processor or a set of adaptive computing devices. In all cases, a GSC's typical and specialized resources will, under appropriate conditions, have the same availability for users across the network as for the GSC's owners.

In addition to its advanced hardware features, SInRG is distinguished from a general-purpose networked cluster in two ways. First, it has advanced hardware features, such as high-latency switches. Second, it makes available to all GSCs a variety of supported middleware, such as NetSolve (http://icl.cs.utk.edu/netsolve) and Internet Backplane Protocol (http://loci.cs.utk.edu), which unify the hardware and network infrastructure into a computational grid.

Technical Issues

Our work addresses several technical challenges involved in ecosystem modeling and ecosystem modeling on grids. Although our methods address these problems in the ecosystem simulation context, they have broader implications for many research applications involving GIS-based dynamic modeling, multiscale system simulation, and distributed heterogeneous computation, as well as for data intensive grid computing.

Multispatial Resolution

A central issue in ecosystem modeling is the need to link dynamic models that operate across different spatial regions and at different rates. Given ecological modeling's characteristics, the spatial patterns (maps) of ecosystem components are generally derived from a GIS, which uses either raster or vector data to represent geographic information. ATLSS addresses spatial resolution problems through the landscape library (see Figure 7).[8] The library's core comprises three groups of classes—Geo-referencing, Regrid, and IODevice classes—that can transform and translate the spatial data to and from other forms. Geo-referencing classes extract regional information from geo-spatial data. These classes accept any number of vertices, specified in the universal transect mercator (UTM), which forms a polygon. In a GIS, regions are usually based on the polygon's shape and location, defined in terms of UTM coordinates. It's therefore possible to extract the same regional information from data sets with different resolutions, registrations, or spatial extents.

Regrid classes facilitate data transformation between different resolutions and registrations. Users can configure and initialize these Regrid classes by

providing the size of a single cell or assigning the exact number of rows and columns to which the data sets should be resized. Spatial data sets can then be passed to the object, which creates and returns a new, rescaled map. IODevices let users automatically incorporate new data formats into the landscape library without changing the code that depends on those formats. Data can be exchanged between the models via file streams or other applications, such as databases or visualization systems.

In addition to spatial resolution problems, different ecological models in a tightly linked simulation group might use different time steps, creating a *multistepping* challenge. In ATLSS, models can pass data, but it's the model developers' responsibility to ensure proper data processing and time-synchronization.

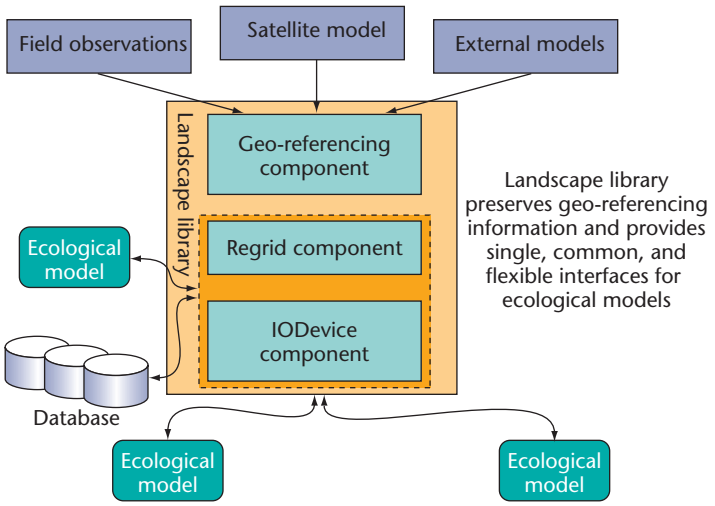

Figure 7. The landscape library's structure. The landscape library provides essential georeferencing functionalities for spatially explicit ecosystem modeling.

Model Interactions

We adopted a component-based modeling methodology in our software design—that is, we built a superstructure (class) for each ecological model. A superstructure contains a model component and a unified communication interface. Once we construct or adapt an ecological model to fit within a superstructure, it can seamlessly exchange data with other superstructures.

Two kinds of data exchange mechanisms are possible: message-passing and database transactions. Figure 8 shows the schematic of ecological model coupling. In one scenario, Models 1, 2, and 3 must be tightly coupled, so we implement the information exchange (blue arrows) using a message-passing library to ensure high performance (that is, low latency) throughout. In another scenario, only Models 1 and 2 are linked, and other models can later use the intermediate results. Database transactions support the two models' connections (red arrows), and the database system stores intermediate data separately. We can thus take advantage of the data cache and multithreaded processing capabilities that most database systems support. As a result, we can maintain high performance, because we separate time-consuming IO operations from data-intensive computations.

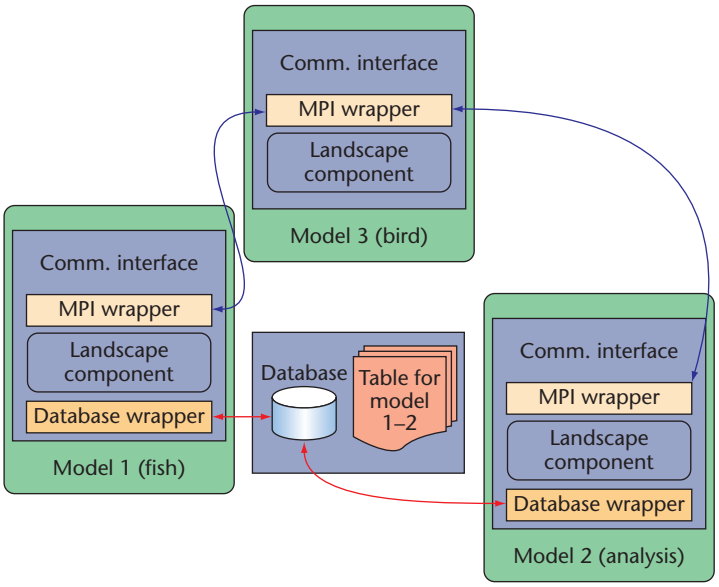

Figure 8. Mechanisms for information change between models. Tightly coupled models can exchange data via high-performance parallel communication libraries, while loosely coupled model will exchange information through database connections.

Heterogeneous High-Performance Computing on Grids

Ecosystem simulation on computing grids will inevitably deploy heterogeneous computation across different computational platforms. We're developing models on an experimental metacomputing framework, Heterogeneous Adaptable Reconfigurable Networked Systems (Harness; see http://icl.cs.utk.edu/harness), which exploits the services of a highly customizable and reconfigurable distributed virtual machine. A DVM is a tightly coupled computation and resource grid that provides a flexible environment to manage and coordinate parallel application execution. The system is designed to support a wide range of DVM sizes, from personal DVMs to enterprise and widely dis-

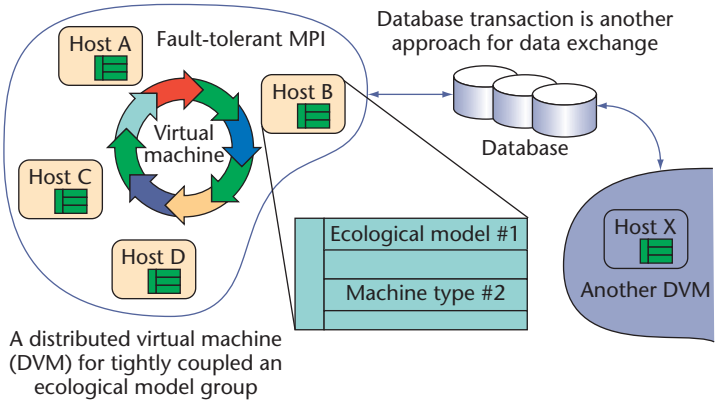

Figure 9. Heterogeneous distributed computation for ecosystem modeling. A distributed virtual machine was created for each tightly coupled simulation. Database connections support data exchange between models in the DVM context.

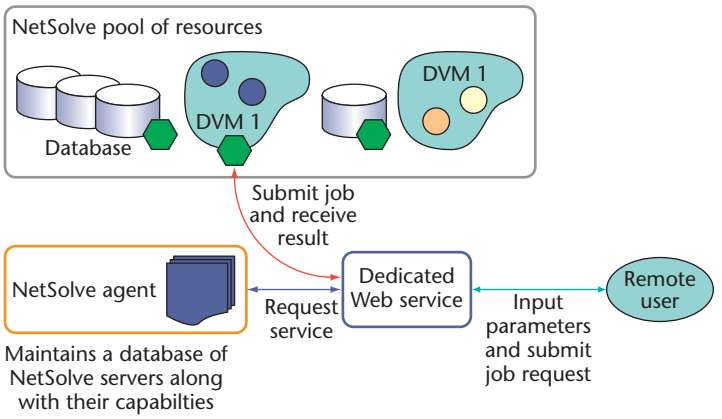

Figure 10. An ecosystem simulation using NetSolve. The NetSolve agent receives a user's computation request through the Web interface and allocates appropriate NetSolve resource executions. After a simulation is complete, the Web interface notifies the user.

tributed DVMs. Different entities collaborate and share resources by temporarily merging and splitting different DVMs. The term "virtual machine" indicates that a system's computing resources can be viewed as a single large distributed-memory computing resource.

Figure 9 shows ecosystem simulation using heterogeneous distributed computation. As the figure shows, we establish a DVM for each tightly coupled ecological model group. Within a DVM context, a heterogeneous message-passing library, fault-tolerant MPI (FT-MPI; see http://icl.cs.utk.edu/ftmpi), provides a series of message-passing primitives, similar to those in standard MPI, to allow high-performance information exchange between models on different computer architectures. For ecological modelers, there's no major difference between developing models using FT-MPI on a virtual machine or using standard MPI on a homogeneous networked cluster. We can therefore conveniently implement information exchange between models in the DVM context. In addition, we can implement any necessary data exchange between separated ecological model groups via database transactions.

Resource Discovery on Grids

Another important grid computing functionality relates to resource discovery. We use the popular NetSolve toolkit (developed at the University of Tennessee) to bring together disparate computational resources connected by computer networks. NetSolve—a client–agent–server system based on Remote Procedure Call—permits remote access to hardware and software components. Alternative grid middlewares include Globus (www.globus.org), Condor (www.cs.wisc.edu/condor/), and Network Weather Service (nws.cs.ucsb.edu). Figure 10 shows an ecosystem simulation using NetSolve.

For each ecological resource (database, DVM, and so on), we establish a NetSolve server and register all the servers into a database maintained by NetSolve agents. When a remote user issues a job request through a dedicated Web site, a NetSolve agent finds the location of an appropriate server, and routes the job to the appropriate ecological resource. Upon completion, the system either stores the result locally (notifying the remote user by email) or ships the result to the remote user via the high-performance Internet Backplane Protocol for file transfer. As we discuss elsewhere,[1] we recently developed a grid service module to further expedite grid computing services over SInRG. The module has four major components: a dedicated Web interface, a job scheduler, a simulation moderator, and a result repository.

Grid computing offers great promise for large-scale ecosystem modeling that will potentially lead to better understanding, control, and management of natural resources. Here, we've focused on a specific application of regional ecosystem simulation. As our understanding of an ecosystem's fundamental properties increases, however, we'll need more complex simulations, which will continue to challenge grid hardware and software architects. From our experience, the scalability of integrated ecosystem simulation depends mostly on the inherited relation-

Related Work in Ecosystem Modeling

Traditionally, ecosystem modeling has focused on the dynamics of nutrient and energy flows as an average across a region. Recent emphasis on global ecology has fostered the development of global biogeochemistry models, linked to climate, with spatial components. Two example models are the Terrestrial Ecosystem Model developed by The Ecosystems Center at the Marine Biology Laboratory (www.mbl.edu/eco42) and the Century model developed by Colorado State University's Natural Resource and Ecology Laboratory (www.nrel.colostate.edu/projects/century).

Regional ecosystem modeling generally aims to understand the relationship between biotic communities and external abiotic factors, including the impact of human activities. Example projects in this area include our Across Trophic Level System Simulation (www.atlss.org) and the Regional Ecosystem Modeling Testbed Project (www.ccpo.odu.edu/RTBproject). Many regional projects focus on particular watersheds, including fisheries and other aquatic components; an example is the Columbia River project (www.columbiariver.org).

Several national projects emphasize the information infrastructures necessary for ecosystem modeling. Among those projects are the US Geological Survey's National Biological Information Infrastructure (www.nbii.usgs.gov), the Science Environment for Ecological Knowledge (seek.ecoinformatics.org), and the Semantic Prototype in Research Informatics (spire.umbc.edu).

ship between individual ecological models and appropriate implementations of parallel communication libraries dedicated for ecosystem simulation. Grid computing can provide virtually unlimited resources for ecosystem simulation, but how to efficiently use those grid resources remains a key challenge.

Collaborations between ecologists and computational scientists are critical to enabling advances in ecosystem modeling. From the computational ecology perspective, there are several challenges involved in ecosystem modeling on computing grids. First, we must provide sufficient support services to sustain the computational environment. Second, we must design a flexible open software architecture to support high-performance ecological multimodeling. Finally, we must ensure that the ecosystem simulations performed on the Grid constitute the next generation of advances, not just proof-of-concept computations. CiSE

Acknowledgments

The US National Science Foundation supported this research under grant number DEB-0219269. The NSF also supports the University of Tennessee's Scalable Intracampus Research Grid (SInRG) project through CISE Research Infrastructure Award EIA-9972889.

References

1. D. Wang et al., "A Grid Service Module for Natural Resource Managers," *IEEE Internet Computing*, vol. 9, no. 1, 2005, pp. 20–26.
2. D. DeAngelis et al., "Individual-Based Models on the Landscape: Applications to the Everglades," *Landscape Ecology: A Top-Down Approach*, J. Sanderson and L.D. Harris, eds., Lewis Publishers, 2000, pp. 199–211.
3. L. Gross and D. DeAngelis, "Multimodeling: New Approaches for Linking Ecological Models," *Predicting Species Occurrences: Issues of Accuarcy and Scale*, J.M. Scott, P.J. Heglund, and M.L. Morrison, eds., Island Press, 2002, pp. 471–476.
4. D. DeAngelis et al., "Landscape Modeling for Everglades Ecosystem Restoration," *Ecosystems*, vol. 1, no. 1, 1998, pp. 64–75.
5. H. Gaff et al., "A Dynamic Landscape Model for Fish in the Everglades and its Application to Restoration," *Ecological Modeling*, vol. 127, no. 1, 2000, pp. 33–53.
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-37), IEEE CS Press, 2004, p. 90282c.
7. D. Wang et al., "A Parallel Structured Ecological Model for High-End Shared Memory Computers," *Proc. First Int'l Workshop on OpenMP*, forthcoming from Springer Verlag, 2005.
8. S. Duke-Sylvester and L.J. Gross, "Integrating Spatial Data into an Agent-Based Modeling System: Ideas and Lessons from the Development of the Across Trophic Level System Simulation (ATLSS)," *Integrating Geographic Information Systems and Agent-Based Modeling Techniques for Stimulating Social and Ecological Processes*, H.R. Gimblett, ed., Oxford Univ. Press, 2002, pp. 125–136.

Dali Wang is a research assistant professor in the Department of Computer Science at the University of Tennessee, Knoxville, and a research associate at the university's Institute for Environmental Modeling. His research interests include parallel and distributed computing, high-performance scientific computation and optimization, ecological/environmental modeling, geocomputing, and large-scale system integration and simulation. Wang has a PhD in environmental engineering from Rensselaer Polytechnic Institute, Troy, New York. Contact him at dwang@cs.utk.edu; www.cs.utk.edu/~dwang

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 Society for Mathematical Biology. Contact him at carr@tiem.utk.edu.

Louis J. Gross is a professor of ecology and evolutionary biology and mathematics at The University of Tennessee, Knoxville, where he directs The Institute for Environmental Modeling. His research interests include multimodeling methods for linking abiotic and biotic components of natural systems in a spatially explicit manner, and developing associated methods for spatial control to assist in natural resource management. Gross received a PhD in applied mathematics at Cornell University. He is currently president of the Society for Mathematical Biology and was formerly chair of the National Research Council Committee on Education in Biocomplexity Research. Contact him at gross@tiem.utk.edu; www.tiem.utk.edu/~gross.

Michael W. Berry is a professor and interim department head in the Department of Computer Science at the University of Tennessee and a faculty member in the Graduate School in Genome Science and Technology Program at the University of Tennessee and Oak Ridge National Laboratory. His research interests include information retrieval, data mining, scientific computing, computational science, and numerical linear algebra. Berry is a member of the Society for Industrial and Applied Mathematics (SIAM), ACM, and the IEEE Computer Society. Contact him at berry@cs.utk.edu; www.cs.utk.edu/~berry.