A GIS-enabled Distributed Simulation Framework for High-Performance Ecosystem Modedling

Dali Wang, Nick Buchanan, Michael W. Berry
Department of Computing Science
University of Tennessee
Knoxville, TN 37966
[dwang, nbuchana, berry]@cs.utk.edu

Eric Carr, Jane E. Comiskey, Louis J. Gross
Institute for Environmental Modeling
University of Tennessee
Knoxville, TN, 37996
[carr, ecomiske, gross]@tiem.utk.edu

Shih-Lung Shaw
Department of Geography
University of Tennessee
Knoxville, TN, 37996
[sshaw@utk.edu

ABSTRACT

A distributed simulation framework is presented to enable natural resource managers to take advantage of both Geographic Information System (GIS) functionality and computationally intensive ecological modeling. Based on one of the newest products from a major vendor of GIS software (ESRI's ArcGIS engine), a user interface was developed to manipulate and visualize digital maps and spatially explicit simulation results. An objective is to allow access to computationally intensive simulations while utilizing an interface system with which natural resource managers have extensive prior experience. Distributed communication services were also developed to support spatially explicit ecological modeling on both local and remote computational platforms. Such a capability is important for natural resource agencies which may have little in-house capacity for extensive computation and could greatly benefit from the use of a computing grid.

To provide examples of the practical application of such a system, we present two ecological modeling cases. These models are components of a major ecological multimodel, the Across Trophic Level System Simulation (ATLSS), developed for application to one of the world's largest ecological restoration projects, in the Everglades of South Florida. The first example is a sequential model to show the working procedure for short-time simulations on a single platform. The second example is a parallel model to demonstrate the procedure for long-time simulations on remote high performance computational platforms. In both cases, we illustrate how the framework provides user access to computationally intensive dynamic, spatially-explicit simulations, enhancing the capabilities available through a stand-alone GIS, while retaining many of the benefits of a GIS.

1. INTRODUCTION

Most natural resource management issues have a significant spatial component, such as the distribution of land use in a watershed, proximity of human populations to a contaminated site or accident, or the degree of habitat fragmentation in an ecosystem. Therefore, Geographic Information System (GIS) technology is used extensively in natural resource management to visualize, analyze, and model natural resource data for management and problem solving.

During the past two decades, researchers have developed a variety of ecological models that are becoming useful tools to assist spatially-explicit natural resource management. Computational ecology has particularly emphasized spatial aspects of natural systems, with a diversity of methods having been developed which are particularly applicable to practical problems faced regularly by resource managers, in areas such as harvest management and invasive species control. As our understanding of ecological processes has increased, ecological models have become more sophisticated, especially in the context of integrated ecosystem simulations which may require high performance computing [4]. This is particularly true of situations in which the management actions have implications at multiple spatial and temporal scales, for which multimodels, utilizing linked sets of models of different mathematical form, are appropriate.

However, the integration of GIS and high-performance ecological models lefts far behind the developments in either fields. There are fundamental reasons: On one hand, a GIS system is traditionally regarded as the high-tech equivalent of the map. An individual map contains a lot of information which is used in different ways by different individuals and organizations. A GIS focuses on providing the facility to extract the different sets of information from a map (roads, settlements, vegetation, etc.) and use these as required. Therefore, a GIS usually provides interactive, friendly interfaces to manually manipulating maps. In the aspect of software design, a GIS typically uses database system to support data storage and information retrievals, and executes the associated computations on-the-fly. Two disadvantages related to the GIS software design are well known [1]: i) inefficiency in dynamic modeling with a strong temporal component, and ii) huge overhead in the GIS software design which leads to poor performance. These two limitations directly account for the fact that most researches, addressing natural resource management problems within monolithic GIS framework, only use simple computational models. On the other hand, high-performance computing (HPC) focuses on non-interactive computations with minimum system overhead, and requires extensive personnel training. Some researches [11] have successfully deployed HPC to solve geographic problems. But these researches are not related to the integrated simulation using both GIS and HPC.

In the research, a component-based distributed simulation framework was present to integrate GIS functionality and high-performance computing to address practical natural resource management issues. Specifically, a major objectives of this paper is to point out the methods we have used to integrate a GIS system with high performance ecological modeling. We illustrate how the framework developed provides user access to computationally intensive dynamic, spatially-explicit simulations, enhancing the capabilities available through a stand-alone GIS, while retaining many of the benefits of a GIS. This offers the capability to deliver high performance ecological modeling to natural resource managers, who typically are familiar with GIS, but have little experience or training with computational science.

2. SIMULATION FRAMEWORK

In this paper, we present a component-based, distributed simulation framework for natural resource managers to take advantage of both GIS functionality and computational intensively ecological modeling. A simplified architecture of this simulation framework contains three major components: a GIS-enabled user interface, distributed communication services, and a back-end computational ecological modeling package. The user interface provides a common gateway for users to launch simulations (either locally or remotely), and visualize the spatial explicit simulation results locally. The distributed communication services provide mechanisms for information exchange between the user interface and the back-end simulation package. The back-end simulation package consists of several stand-alone spatially explicit ecological simulation modules. In this research, we have used the Across Trophic Level System Simulation (ATLSS) package as an example. ATLSS has been designed to assess the effects on key biota of alternative water management plans for the regulation of water flow across the Everglades landscape. The conceptual model for ATLSS is based on the trophic structure of consumption, who eats who, across the landscape. Organisms are related energetically as members of a food chain – a series of organisms that eat one another. (See www.atlss.org for more information).

2.1 GIS-enabled user interface

The GIS-enabled user interface provides three basic functions: i) view GIS data[1]; ii) launch external model simulations; and iii) create new GIS layer(s) from spatially explicit simulation results. Building a customized GIS-enabled user interface from stretch is not trivial work, therefore, we chose the ESRI's ArcGIS Engine 9.1 as our software development kit. ArcGIS Engine is the first ESRI's product that allows users to access the core functionality of ArcGIS (ArcObject) outside the monolithic ESRI's integrated software development environments. In other words, ArcGIS Engine provides a feasible channel for the component-based integrated simulation using both GIS and HPC. Particularly, ArcGIS Engine Java Application Programming Interfaces (APIs) were used to build the user interface.

2.2 Distributed communication services

The distributed communication services provide linkages between the GIS-enabled user interface and the back-end ecological modeling package (ATLSS). Distributed communication services run as daemons, and are designed to support both short-time and long-time computations. For short-time computations, the communication services provide mechanisms to report simulation progress status, to provide feedback the intermediate results to the user's screen, and to force the user to wait for job completion. For the long-term computations, the communication services ensure the job is successfully launched on remote machines, monitor the computation unintrusively, transfer the model output to the user's local machine, and finally notify the user when the computational jobs have been completed. The structure and functions the communication services are explained in Section 3.1.2 and 3.2.2.

2.3 Back-end ecological modeling package

As mentioned above, ATLSS is a family of ecological models in which models are integrated within a spatially explicit landscape. At the bottom of ATLSS's model hierarchy, process models are used for lower trophic-level organisms, such as algae, which interact only locally. 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 changes in water-levels. Finally, individual-based models represent populations of top predators and other large-bodied species, such as wading birds and panthers. These individual based models are rule-based approaches that can track the growth, movement, and reproduction of many thousands of individuals across the landscape [4]. Depending on the trophic position, some ecological models are stand-alone, which means the activities of those species (such as panther and crayfish) are not directly related, while some ecological models (such wading bird and fresh-water fish) are tightly coupled. For coupled models, ATLSS provides mechanisms for both stand-alone and integrated simulations [6].

2.4 Basic software structure

Although the models within ATLSS use various modeling approaches to simulate different species, the software structure of each model is similar: each consists of three components - the Landscape library, Date library and Model engine.

All models within ATLSS use a landscape library [3] to handle the spatial patterns (maps) of ecosystem components generally derived from a geographic information system (GIS). The core of the Landscape library comprises

[1] including open/save, display and control of GIS maps.

three groups of classes (*Geo-referencing, Regrid, and IODevice classes*) that can transform and translate spatial data to and from other forms. Contrary to a traditional GIS, this library uses vector-based C++ templates to expedite computations over raster-based geographic information data.

In ATLSS, a Date library provides a set of functions for manipulating dates. Those functions include setting and retrieving dates, incrementing and decrementing dates by a specified time interval, taking the difference of two dates, determining whether one date is later than another, and computing the day of year of a given date.

The Model engine of each ATLSS model executes three phases: initialization, computation and finalization. In the process of initialization, the model uses configuration files to find appropriate input files (ecological parameters and geoinformation data). For better simulation performance, geoinformation data is stored in a format different from the traditional raster GIS format. We create files to separate storage of location-related information items (such as Universal Transverse Mercator (UTM) and the actual cell values (such as water depth associate with each cell). More specifically, location related geo-information are stored in plain text format for easier conversion and transform between various formats (e.g., ESRI and netCDF). Cell values are stored in binary format for convenient and efficient handling using a programming language such as C++. Due to inherited ecological behaviors and characteristics of a given species, the computation requirements of ecological models can vary: simple models may only need sequential implementation, so that the overall execution time is typically only several minutes. Complex models may require parallel implementations on high performance computational platforms, and consume several hours for each run. All outputs are produced in the finalization phase, which includes several geo-information datasets created by the Landscape library.

3. CASE STUDIES

In this section, we present two case studies to demonstrate the working procedure under this simulation framework which links ATLSS and GIS. The first case is a short-time, single platform sequential simulation. A good example is the Spatially Explicit Species Index (SESI) model for deer, referred as DEER [2]. The second case is the long-time, remote parallel simulation produced by the Parallel Across Landscape Fish model (PALFISH) [10].

3.1 Short-time, single platform sequential simulation (DEER)

3.1.1 Model structure of DEER

DEER is one of a family of landscape-based, spatially explicit species index models developed to assess the impact of management scenarios on habitat conditions of different species in South Florida. SESI models differ from Habitat Suitability Index (HSI) models and the associated habitat evaluation procedures (HEP) in that SESI models incorporates temporal and dynamic landscape features. They provide a relatively simple method of comparing species responses to environmental conditions, as an alternative to more complex approaches such as process models, structured population models and individual-based models. Currently, DEER produces an index map of the annual quality of breeding potentials for white-tail deer.

The peak reproductive season for deer is between January 1 and May 31. Ponded water can act as an impediment to fawning, movement and foraging during the breeding season. If the food supply is interrupted during this period, which can happen during high water, the health of mother and offspring may suffer, and fawns are less likely to be recruited into the herd. Elevated water levels can make beds uninhabitable, and high water can drown young fawns. In DEER a water depth of 55 cm is defined to be the depth above which movement and foraging are assumed to be precluded. Water of any depth during this season is assumed to restrict fawning and impede movement, and so subtracts from the index. The degree to which water is an impediment is represented by calculating the ratio of (water-depth days)/(maximum possible water-depth days) during the reproductive season. The hydroperiod during the previous year is an indication of the quality and availability of forage prior to fawning, which will influence the health of females and thus their likelihood to fawn. A shortened a hydroperiod reduces the quality of forage; a prolonged hydroperiod reduces the availability of forage. The breeding potential index is weighted by a multiplicative factor that reflects those hydroperiod effect.

The DEER model uses topographic and hydrological data provided by the South Florida Water Management District (SFWMD) and vegetation data provided by the Florida Gap Analysis project. The spatial resolution of the SFWMD data is 3.2 km by 3.2 km (2 by 2 miles). The outputs of DEER include the breeding potential index (BPI) values assigned to each cell (500 m by 500 m) and a time series of the mean overall index attained each year on each hydrologic scenario[2].

3.1.2 Computational platform and underlying working procedure

The computational platform used in this simulation is a Debian Linux computer configured with duel 2.2GHz 32-bit AMD Opteron processors and 1 GB memory. The machine has also been configured to run ArcGIS Engine 9.1 Java API[3].

The underlying working procedure of a short-time simulation is illustrated in Fig.1. The user opens GIS maps (i.e., digital maps showing the boundaries and study regions) via *ModelViewer*, and then clicks *JobLaunchBar* to invoke *JobGenerator*, which in turn generates a script to execute the DEER model. *JobMonitor* captures all the DEER model execution information and creates a dynamic progress status bar. Once the simulation has completed, model results (including those geo-information data (BPI) produced by the Landscape library) are stored in a specified directory. The user can then use *MapGenerator* to create GIS maps from the BPI data, which can be viewed via *ModelViewer*

[2] See www.atlss.org/cerp_runs/mod_info/d_deer.html for more details.

[3] Although Linux/Unix computers were used here, this distributed simulation framework is capable of unifying multiple computational platforms (i.e., Windows, Linux and Unix).

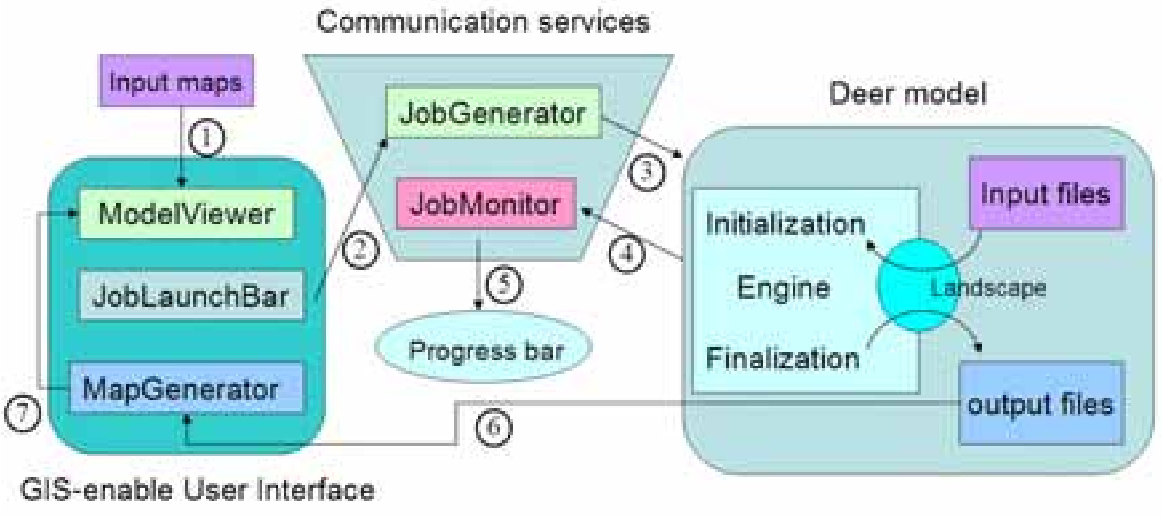


Figure 1: Working procedure for a short-time ecological simulation

or manipulated by standard ArcGIS systems (such as ArcGIS desktop).

3.1.3 Screen-shot of DEER simulation

Fig.2 contains three screen-shots of the DEER model execution under the distributed simulation framework. The top left graphic shows the progress status bar in front of *ModelViewer*. The top right graphic illustrates the DEER results (BPI) in ArcGIS map format as viewed by the *ModelViewer*. The bottom graphic shows information monitored by the *JobMonitor* during DEER execution.

3.2 Long-time, cross-platform parallel simulation (PALFISH)

3.2.1 Model Description of PALFISH

PALFISH is a parallel, age-structured population model for freshwater fish functional groups in South Florida. PALFISH includes two main subgroups (small planktivorous fish and large piscivorous fish), structured by size. In the complex integrated system of ATLSS, PALFISH is an important link to the higher level landscape models, since it provides a food base for several wading bird models. An objective of the PALFISH model is to compare, in a spatially explicit manner, the relative effects of alternative hydrological planning scenarios on fresh-water fish densities across South Florida. Another objective is to provide a measure of dynamic, spatially-explicit food resources available to wading birds.

The study area for PALFISH modeling contains approximately 111,000 landscape cells, with each cell 500m on a side. Each landscape cell has two basic types of area: marsh and pond. The fish population simulated by PALFISH is size-structured and is divided into two functional groups: small and large fishes. Both of these groups appear in each of the marsh and pond areas. Each functional group in each area is further divided into several fish categories according to age, and each age class has 6 size classes. The fish population in each cell is summarized using the total fish density (or biomass density) within that cell. Each cell, as an element of the landscape matrices, contains an array of floating-point numbers representing individual fish density of various age classes. The length of the array corresponds to the number of age classes for that functional group. Normally, when a overall fish density is referenced, the value reflects the total fish densities of all the fish age classes combined. In PALFISH, spatial and temporal fluctuations in fish populations are driven by a number of factors, especially the water level. Fluctuations in water depth, which affect all aspects of the trophic structure in the Everglades area, are provided through an input hydrology data file for each timestep throughout the execution of the model.

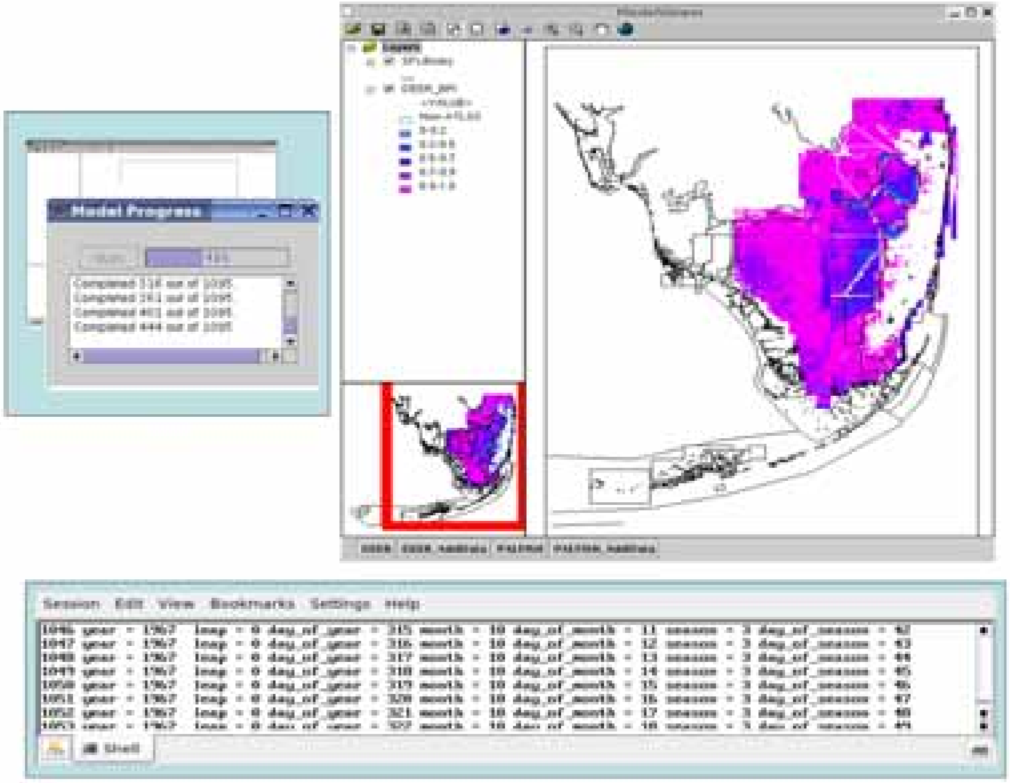
Major concerns associated with PALFISH include its long runtime. The average runtime of PALFISH is around 2-3 hours (using all processors of a 14-CPU 400 MHz Sun Enterprise 4500) for a typical 31-year simulation[4].

3.2.2 Computational platforms and underlying working procedures

Two computational platforms are used in this study. The front-end is the Linux box mentioned above, which is configured to run ArcGIS Engine applications. The back-end high performance platform used is a Symmetric Multi-Processor (SMP) Sun Enterprise 4500 configured with 14 400MHz Sun Ultra Sparc II processors, 10 GB memory and 3Gb/s interconnections. The SMP is a part of the Scalable Intracampus Research Grid (or SInRG) (icl.cs.utk.edu/sinrg) at the University of Tennessee, Knoxville. An implementation of the MPI standard library, LAM-MPI (www.lam-mpi.org), was selected to support the message-passing communication between processors.

Fig.3 illustrates the underlying working procedure of a long-time simulation. On the front-end computer (Linux machine), the user first opens GIS maps (using *ModelViewer*),

[4] See [5, 9, 10] for more information on the parallel implementation and performance of PALFISH.



```
Completed 316 out of 1095
Completed 361 out of 1095
Completed 401 out of 1095
Completed 444 out of 1095
```

```
1046 year = 1967  loop = 0 day_of_year = 315 month = 10 day_of_month = 11 season = 3 day_of_season = 42
1047 year = 1967  loop = 0 day_of_year = 316 month = 10 day_of_month = 12 season = 3 day_of_season = 43
1048 year = 1967  loop = 0 day_of_year = 317 month = 10 day_of_month = 13 season = 3 day_of_season = 44
1049 year = 1967  loop = 0 day_of_year = 318 month = 10 day_of_month = 14 season = 3 day_of_season = 45
1050 year = 1967  loop = 0 day_of_year = 319 month = 10 day_of_month = 15 season = 3 day_of_season = 46
1051 year = 1967  loop = 0 day_of_year = 320 month = 10 day_of_month = 16 season = 3 day_of_season = 47
1052 year = 1967  loop = 0 day_of_year = 321 month = 10 day_of_month = 17 season = 3 day_of_season = 48
1053 year = 1967  loop = 0 day_of_year = 322 month = 10 day_of_month = 18 season = 3 day_of_season = 49
```

Figure 2: Screen shots in a short-time ecological simulation

and then clicks *JobLaunchBar* to invoke *JobGenerator*, which in turn creates a new thread to communicate with *JobGeneratorProxy* on the back-end computer (SMP). The new thread will keep running until the remote *JobGenerater-Proxy* successfully creates a new process to generate a script to execute the PALFISH model. Beyond this point, while PALFISH is running quietly on the back-end computer, the user on the front-end computer is free to perform other tasks. On the SMP, the *JobMonitor* captures all the PALFISH model execution information. Once the simulation has completed, model results will be transferred to a specific directory on the front-end computer via the file transfer services (i.e., *FileTransferClient* on the SMP and *FileTransferServer* on the Linux box). The user will receive an email notification of job completion (sent out by the *EmailClient* on the SMP), the user can then use *MapGenerator* (which will automatically convert data from Big-endian format to Little-endian format) to create new GIS maps from PALFISH's output.

3.2.3 Screen-shot of PALFISH simulation

Fig.4 contains two screen-shots of the PALFISH model on the distributed simulation framework. The top graphic illustrates the PALFISH results (averaged fish density on four given dates) in ArcGIS map format. The bottom graphic shows a sample user notification message.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a GIS-enabled distributed simulation framework to deliver GIS functionality and high performance ecological modeling capacities to natural resource managers. We believe it is one of very few efforts based the integration of GIS and high-performance ecological modeling under a single simulation framework to address practical natural resource management problems. This framework includes a user-friendly interface (similar in look to ArcGIS desktop which is very familiar to resource managers) and transparent ecological modeling (including those models that require high performance computing), so that users can focus on the actual ecological findings and analysis, instead of on complex computational techniques, geoprocessing and visualization.

Future work includes further enhancements of GIS-enabled user interfaces, to allow users to interactively manipulate thre resource maps and model requests (allowing for example site-location selection, model parameter selection, automated sensitivity analysis, etc). The distributed communication services presented herein are developed for dedicated use in secured network domains. We do not address security issues as those would be typically associated with the underlying security configurations of networks. Extensions to the communication services will focus mainly on incorporating common grid middleware[5](such as GLOBUS (www.globus.org) and NetSolve (icl.cs.utk.edu/netsolve)) and batch systems (such as Portable Batch System (www.openpbs.org)) to deliver non-dedicated, grid-enabled simulation capacities to stakeholders. We are also in the process of em-

[5] See [7, 8] for more information on grid computing for integrated regional ecosystem modeling

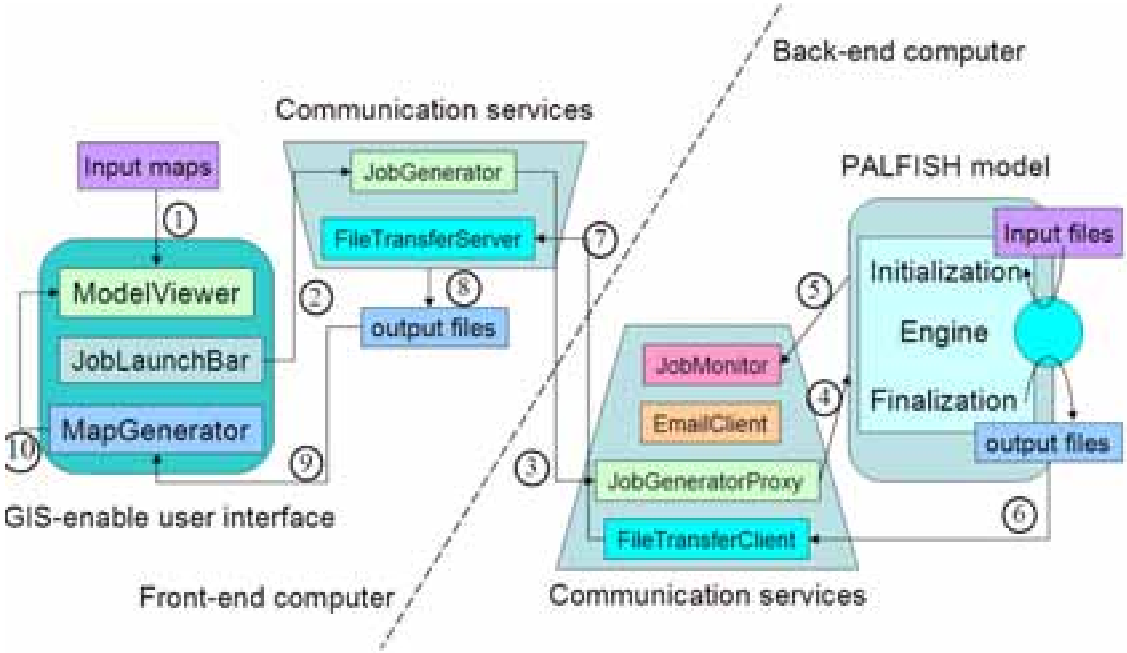
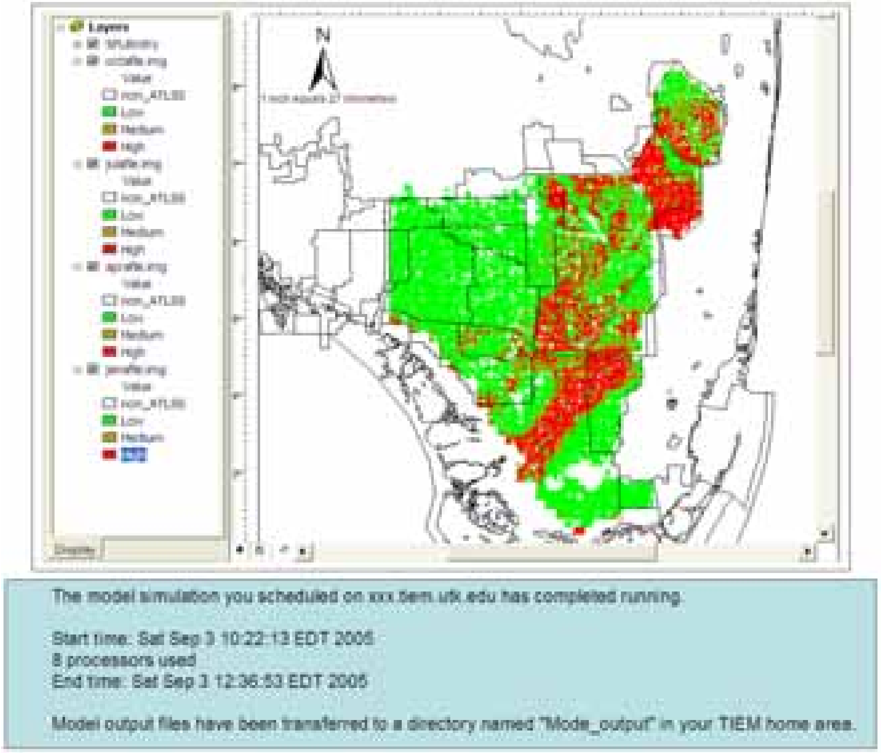


Figure 3: Working procedure for a long-time ecological simulation



The model simulation you scheduled on xxx.tiem.utk.edu has completed running.

Start time: Sat Sep 3 10:22:13 EDT 2005
8 processors used
End time: Sat Sep 3 12:36:53 EDT 2005

Model output files have been transferred to a directory named "Mode_output" in your TIEM home area.

Figure 4: Screen shots in a long-time ecological simulation

bedding ArcGIS geo-processing functionality inside the backend simulation modules, so that the portion of the simulation package can handle and process vector-based GIS information.

Acknowledgments

This research has been supported by the National Science Foundation under grants No.ĨIS-0427471 to the University of Tennessee. This research used the resources of the Scalable Intracampus Research Grid (SInRG) Project at the University of Tennessee, supported by the National Science Foundation CISE Research Infrastructure Award EIA-9972889. For ATLSS model development, we appreciate the support of the U.S. Geological Survey, through Cooperative Agreement Number 1445-CA09-95-0094 with The University of Tennessee.

5. REFERENCES

[1] A. Brimicombe. *GIS, Environmental Modeling and Engineering*. Talyor Francis, New York, New York, 2003.

[2] J. Curnutt, E. Comiskey, M. Nott, and L. J. Gross. Landscape-based spatially explicit species index models for everglade restoration. *Ecological Applications*, 10:1849–1860, 2000.

[3] 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). In H. R. Gimblett, editor, *Integrating Geographic Information Systems and Agent-Based Modeling Techniques for Stimulating Social and Ecological Processes*. Oxford University Press, 2002.

[4] L. J. Gross and D. DeAngelis. Multimodeling: New approaches for linking ecological models. In J. M. Scott, P. Heglund, and M. L. Morrison, editors, *Predicting Species Occurrences: Issues of Accuracy and Scale*. Island Press, 2002.

[5] A. Immanuel, M. W. Berry, L. J. Gross, M. Palmer, and D. Wang. A parallel implementation of alfish: Compartmentalization effects on fish dynamics in the florida everglades. *Simulation Practice and Theory*, 13(1):55–76, 2005.

[6] D. Wang, E. A. Carr, M. W. Berry, E. J. Comiskey, and L. J. Gross. A parallel simulation framework for integrated ecosystem modeling. *IEEE Transactions on Distributed and Parallel Systems*, in review.

[7] D. Wang, E. A. Carr, M. W. Berry, and L. J. Gross. A grid service module for natural resource managers. *Internet Computing*, pages 35–41, Jan/Feb 2005.

[8] D. Wang, E. A. Carr, L. J. Gross, and M. W. Berry. Toward ecosystem modeling on computing grids. *Computing in Science and Engineering*, pages 44–51, Sep/Oct 2005.

[9] D. Wang, L. J. Gross, and M. W. Berry. A parallel structured ecological model for high end shared memory computers. *Proceedings of 1st International Conference on OpenMP*.

[10] D. Wang, L. J. Gross, E. A. Carr, and M. W. Berry. The design and implementation of parallel fish model for south florida. *Proceedings of 37th Hawaii International Conference on System Sciences*, 9(9):90282c, 2004.

[11] S. Wang and M. P. Armstrong. A quadtree approach to domain decomposition for spatial interpolation in grid computing environments. *Parallel Computing*, 29(10):1481–1504, 2003.