International Congress on Environmental Modelling and Software

3rd International Congress on Environmental Modelling and Software - Burlington, Vermont, USA - July 2006

Jul 1st, 12:00 AM

Experiences in Sharing Environmental Models in Distributed Environments

David Swayne

Vimal Sharma

Markiyan Sloboda

Experiences in Sharing Environmental Models in Distributed Environments

David Swayne[a], **Vimal Sharma**[a], **Markiyan Sloboda**[a]

[a]Computing Research Laboratory for Environment, University of Guelph, Guelph, ON, Canada.
[a]*dswayne@uoguelph.ca*

Abstract: Our paper describes experiential work in developing distributed versions of several hydrological and non-point source pollution models. Work in parallelization of the calibration of environmental models and in the population of belief network representations of these models has required us to developing a client-server framework. Newer models often have large computational needs in calibration. Similarly, replacing computational models with belief network representations suggests the utility of parallel computation for the Monte Carlo techniques to generate conditional distributions for responses of outputs to the model parameters. We are required to maintain the integrity of the original model and to decouple interfaces from their computational engines. This has resulted in the development of sophisticated "wrappers", and middleware components for communications and task management.

Keywords: High Performance Computing, Distributed Computing, Hydrological modeling.

1. INTRODUCTION

In this study, a framework to distribute the computational load from client side to server side is proposed. The major advantage of this framework is that the user needs to only concentrate on the data and the problem at hand and rest is taken care of by the server side (developer/maintainer of the model(s)).

This paper is based on work in developing distributed versions of several hydrological and non-point source pollution models. Parallelization of the calibration of environmental models and in the population of belief network representations of these models has led us to develop a number of multi-computer applications. For example, new and complex environmental models often pose a significant overhead in calibration, with serial computation times measured in days(Eckhardt and Arnold [2001]). Similarly, even the simplest of "store and forward" non-point source pollution models require, for watersheds with a few hundred components and a relatively simple yearly meteorological profile, upwards of a thousand Monte Carlo simulations to develop the necessary conditional probability distributions to populate the belief network representation with reasonable results (Bobba et al. [1996]; Sloboda [2005]). These two problems have led our group to investigate the utility of a SHARCNET supercomputer cluster to facilitate the "training" of either the model (for calibration) or the belief network. One key objective is to maintain the integrity of the original model and to decouple the model interfaces from their computational engine, necessitating the development of sophisticated "wrappers", and middleware components for task management and data integrity maintenance. This activity led us to propose and develop an environment consisting of a stand-alone (possibly remote) interface for input of the characteristics of a particular watershed, a communications "manager" to transfer the site-specific information to a task manager, which is capable of managing the training operation on the computer cluster, and the subsequent data re-assembly.

The proposed framework is presented in section 2 followed by brief description of hydrological model used for testing in section 3. Test case of belief networks 4. The results are presented and discussed in section 5.

2. THE PROPOSED FRAMEWORK

In this section, the proposed model is presented.

The proposed framework consists of following four major components (Figure 1):

- Client
- Name server
- Model Server
- High Performance Computing component (if available).

The client side is responsible for gathering necessary input data from the user, required by the hydrological model. The client consists of two main components, a user interface and a Communication layer (Figure 2). The user interface will allow the user to select the model that needs to be run and collect the input files necessary. In addition, the user interface can perform limited error checking for models that are supported by default at client level itself. This will save in communication time over the internet for erroneous input to the model server.

The Name Server consists of three layers as shown in (Figure 3). Client makes a request for address of the available model server, which is stored in a queue and then, in first come first served order, is sent to task manager. The task manager returns to the client either the address of the available model server or non-availability message.

Task manager has two main functions. First, it takes requests from the queue and sends back a response to the Client with the needed information to connect to a model Server. Secondly, it maintains a log of available servers running different hydrological models. If any server starts/stops/busy, it is logged with the task manager of the name server. This framework encourages collaboration among the modeling community while maintaining their independent identity. As a modeling team just needs to register its model server that is available for processing with the name server. And the name server will start directing the clients to the model server for data processing.

Communication layer is present in every component. The purpose of this layer is to act as a single source liaison for the component with other components in the framework. Any communication between the components is basically done through their respective communication layers.

Lastly, the model server is component of the framework that actually executes the model with the input data provided by the user. This component may further contain high performance computing capability.

3. HYDROLOGICAL MODEL

In this section, brief description of the model that has been used for testing is presented.

3.1 The GAMES Model

The Guelph model for evaluating effects of Agricultural Management systems on Erosion and Sedimentation (GAMES), based on Universal Soil Loss Equation (Wischmeier and Smith [1978]), was developed for watershed management (Rudra et al. [1986]). It predicts soil loss by erosion and the delivery of suspended solids from the fields to the streams. GAMES demonstrates areas within a watershed that are critical sediment sources and also provides a method to evaluate various soil conservation practices (Dickinson et al. [1987]; Dickinson et al. [1990]). The watershed used for analysis with GAMES must be discretized into field-sized elements with homogeneous characteristics of land use, soil type, and slope class. The model can be used for seasonal or annual assessments, depending on the selection of input parameter values. The sediment delivered from each cell to the watershed's stream channels is calculated from a delivery ratio for each cell based on the field cell's characteristics. The delivery ratio calculations require parameter 'α', which needs to be calibrated.

3.2 The Belief Network

Joint probability distribution can answer any question about the domain, but can become intractably large as the number of variables grows. Furthermore, specifying probabilities for atomic events is rather unnatural and may be very difficult unless a large amount of data is available from which to gather statistical estimates. We use a data structure called a belief network (also known as a Bayesian network or probabilistic model) to represent the dependence between variables and to give a concise specification of the joint probability distribution (Haas [1991]). A belief network captures believed relations (which may be uncertain, stochastic, or imprecise) between a set of variables, which are relevant to some problem (Sloboda [2005]; Dorner [2000]). They might be relevant because they will be observable, because their value

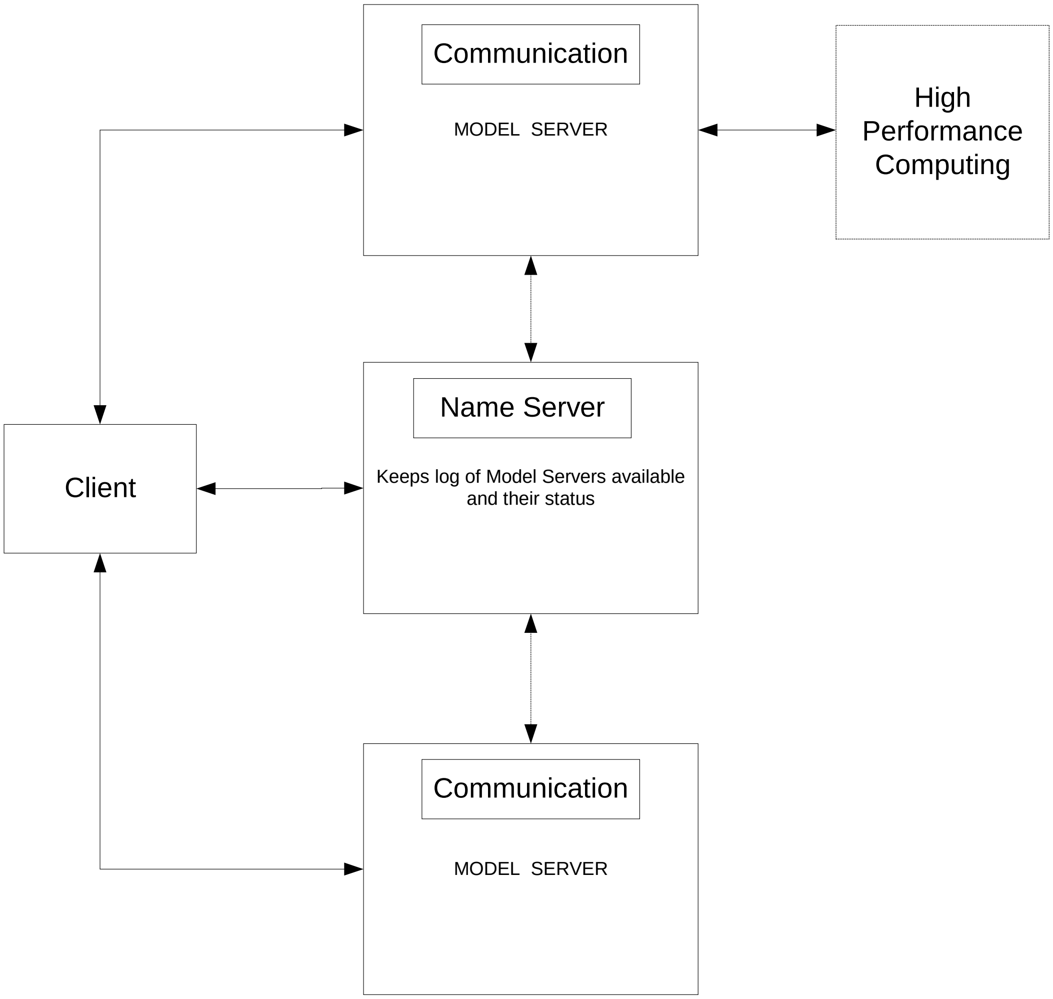
Communication
MODEL SERVER
High Performance Computing
Name Server
Keeps log of Model Servers available and their status
Client
Communication
MODEL SERVER

Figure 1. Structure of the proposed framework

is needed to take some action or report some result, or because they are intermediate or internal variables that help express the relationships between the rest of the variables.

4. TESTING OF THE PROPOSED FRAMEWORK

In this study, preliminary testing was done for the proposed framework for populating of probability network using Monte Carlo simulations for GAMES, hydrological model briefly described in section 3.

In the testing, the required model inputs were sent by the client using the name server to the model server running GAMES. Due to high computational needs for running Monte Carlo simulations the model server used a high performance computing cluster of SHARCNET for computations.

SHARCNET stands for Shared Hierarchical Academic Computing Network. Established in 2000, SHARCNET is the largest high performance computing consortium in Canada, involving eleven universities and colleges across southern Ontario. SHARCNET also refers to a grid of high performance clusters of thousands of processors on a dedicated, private high-speed wide area network with a throughput of 1 Gigabits per second. Powered by the Ontario Research Innovation Optical Network (ORION) and the state-of-the-art operating system environments, the grid of SHARCNET enables researchers to run a single parallel application across

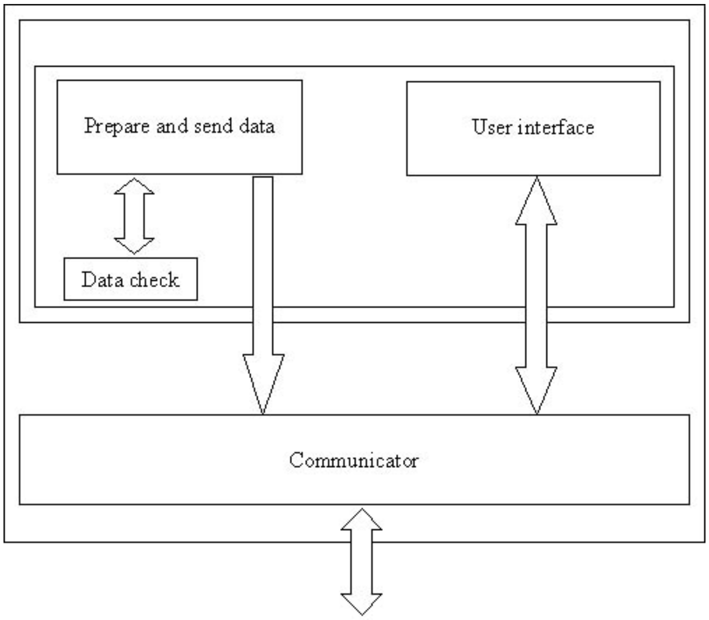

Figure 2. Structure of a client

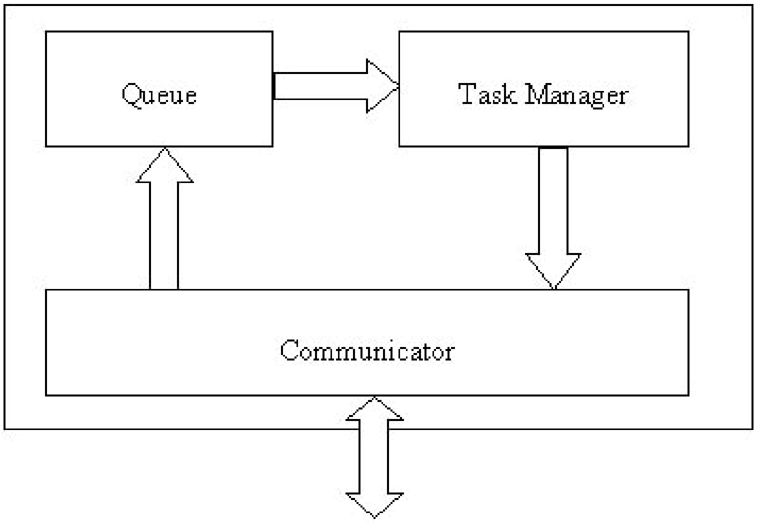

Figure 3. Structure of the Name Server

multiple clusters deployed at different institutions seamlessly.

The model server is a multithreaded server, which prepares the data to be sent to SHARCNET. It also receives the results and populates the probability network using Netica API software. When request comes to the model server it is stored in a queue, and when SHARCNET cluster becomes available it sends data there for further calculations. After SHARCNET sends back results Bayesian network is populated using Netica API. Then acknowledgement is sent to Client with the information that probability network is constructed and ready to work with. Work with Netica is performed through application, which Client connects to. The application is responsible for constructing Bayesian network, compiling, reflecting changes made by user, sending and receiving data from client. Communication continues through communicator until Client decides

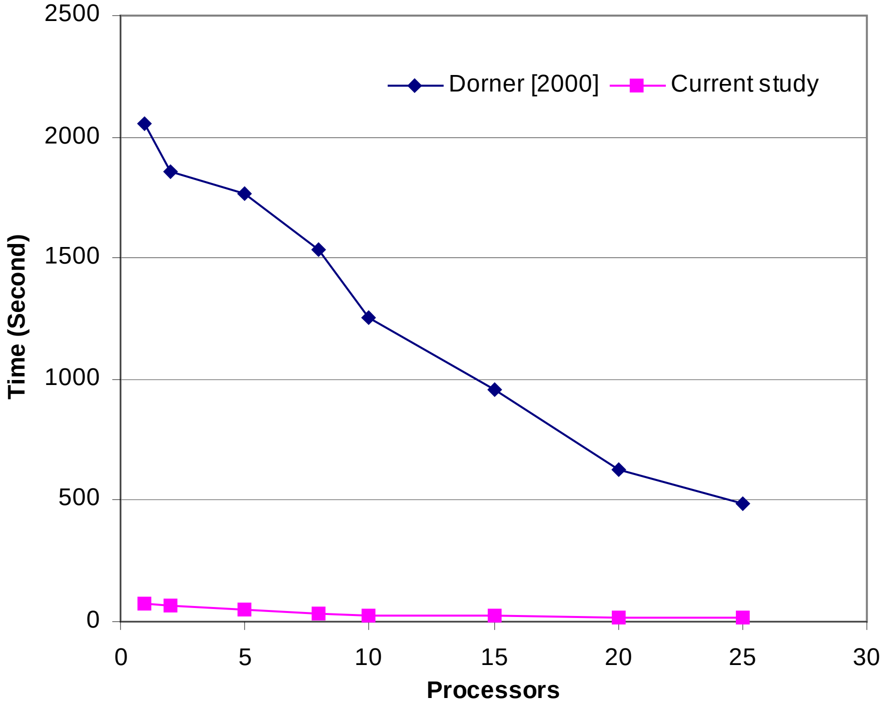

Figure 4. Comparison of timing results with results from Dorner [2000]

to close the connection by quitting the application or some error in connection occurs. A log is kept on the server from every thread. In that way server is doing most of the job, building the probability network and responding to modifications that are requested from the client side. Files from Client and to Server are sent and received using file transfer protocol (ftp). Data sent to and from SHARCNET is transferred using secure file transfer protocol (sftp).

5. RESULTS AND DISCUSSION

Dorner [2000] showed the possibility of building a belief network using the Monte Carlo simulation with GAMES using a single processor computer. In this study, for preliminary testing of the proposed framework, probability network was populated using multiple processors (SHARCNET). It was observed that using SHARCNET, the most computationally intensive part of constructing the probability network could be done in significantly lesser time (Figure 4). The savings in time observed as compared to Dorner [2000], were because of limiting the use of the file I/O (Table 1). The results were stored in the memory rather than files on disk, and broadcasted to processors whenever necessary. The data sets used for testing were the same (Stratford Avon Watershed).

Number of Processors	Time (seconds), (Dorner [2000])	Time (seconds)
1	2052.49	71.19
2	1859.28	69.80
5	1768.94	47.11
8	1535.82	30.90
10	1252.09	26.75
15	959.19	22.05
20	624.37	19.41
25	489.76	16.50

Table 1: Comparison of timings for populating the belief network

With the current framework not only belief network was populated in a significantly lesser time, but also this approach provided more reliability and confidence in running the model. The user does no longer have to worry about knowing the in depth coding of the model but just needs to ensure the data validity for the specific site and provide specified data. The framework can be connected by a user from anywhere in the world. It can easily accommodate collaboration with modelers interested in providing their models for user through the proposed framework. The server running their model need not be present in the same geographic location. This also increases the reach of the expensive resources like HPC to the users.

6. CONCLUSIONS

In this study, a framework is proposed that moves the computing from the user end to the server end. This gives the user indirect access to resources like high performance computing, which otherwise are too expensive for a single user. This framework also enables the users to concentrate their resources for actual analysis of the problem rather than technical in-depth of coding and running the hydrological models for calibration or populating belief networks. The preliminary testing has shown successful use of the proposed framework.

7. ACKNOWLEDGEMENTS

This research is supported by Canadian Foundation for Climate and Atmospheric Sciences, The Canadian Natural Sciences and Engineering Research Council, and Environment Canada.

8. REFERENCES

Bobba, A. G., V.P.Singh, and L.Bengtsson , Application of First-Order and Monte Carlo Analysis in Watershed Water Quality Models, *Water Resources Management*, 10, 219-240, 1996.

Dickinson, W. T., R.P.Rudra, and G.J.Wall , Targetting Remedial Measures to Control Non-point Source Pollution, *Water Resources Bulletin*, 26(3), 499-507, 1990.

Dickinson, W. T., R.P.Rudra, and G.J.Wall , Identification of Soil Erosion and Fluvial Sedimentation Problems, *Hydrologic Processes*, 1, 111-124, 1987.

Dorner, S. M. *Graphical Probability Models for Evaluating Best Management Practices for Agricultural Watersheds*, M.Sc. Thesis, University of Guelph, ON, 2000.

Eckhardt, K., and Arnold, J. G. Automatic calibration of a distributed catchment model, *Journal of Hydrology*, 251(1/2), 103-109, 2001.

Haas, T. C. A Bayesian Belief Network Advisory System of Aspen Regeneration, *Forest Science*, 37(2), 627-654, 1991.

Rudra, R. P., W.T.Dickinson, D.J.Clark, and G.J.Wall , GAMES - A Screening Model of Soil Erosion and Fluvial Sedimentation in Agricultural Watersheds, *Canadian Water Resources Journal*, 11(4), 58-71, 1986.

Sloboda, M. *Parallelization of the Uncertainty Analysis in Environmental Models*, M.Sc. Thesis, University of Guelph, ON, 2005.

Wischmeier, W. H., and Smith, D. D. Predicting rainfall erosion loss – A guide to conservation planning, *Agriculture Handbook No. 537. Washington D. C. : USDA-ARS*, 1978.