Korean Journal of Remote Sensing, Vol.17, No.4, 2001, pp.359~368

Communication

Agent-Based Load Balancing Method for Web GIS Services; Web-Based Forest Fire Management System

Yun-Won Jo*, Myung-Hee Jo*, and Jeong-Soo Oh**

Dept. of Urban Information & Cadastral Engineering, Kyungil University, Korea*

Div. Forest Environment, Korea Forest Research Institute, Korea**

Abstract : The prototype of forest fire management system on Web was studied. In the architecture of this system, one of the most important concerns is to handle load upcoming to Web Server so that it provides Web service without any delay or failure. In order to solve this problem, the agent is designed on dispatcher in a Web Server cluster and implemented to distribute load dynamically by considering the information related to load coming to the Web Server such as the number of connection to the Map Server. The proposed forest fire management system has user-friendly interface with the GIS mapping functionality by selecting Map Objects Internet Map Server (MO IMS) as Map Server and is implemented using Java as programming language.

Key Words : Web GIS, Load Balancing, Agent, Forest Fire.

1. Introduction

Internet has grown so rapidly over the last few years and continued to expand at amazing paces. This situation results in heavy demands on Web Server and performance and scalability are a matter of great concern because single server cannot handle the client's requests.

Especially, in the case of the information related to natural disaster such as forest fire or flood damage, the first thing to be concerned is about how fast it is presented on the Web sites without delay or shutdown. Because these sites are supposed to have a number of hits from clients simultaneously when the disaster occurs, the methods to handle load dynamically should really remain big concerns. In other words, if this problem is not properly handled yet, clients become frustrated by slow respond time or refused connections, and perhaps never visit this site again. (Chris Gago, 1999) Moreover, this site becomes unstable or even turn out of fail under critical load conditions. Finally, the main goal of the site to provide information about disaster within seconds is achieved.

Received 5 November 2001 ; Accepted 5 December 2001.

To avoid such a problem in networks, it needs to ensure maximum scalability and availability with a solution that balances the load effectively and protects the user from these poor conditions. Therefore, you are probably looking for a way to handle your share of that traffic, and even to grow your share. (Chris Gago, 1999)

Nowadays in order to solve this problem, cluster-based architecture is utilized as a proper and effective alternative. In addition, the agent proposed in this paper is designed on dispatcher in a Web Server cluster and implemented to distribute load dynamically while current other servers typically has led bottlenecks in the server node. In order to select the proper server, the agent needs to collect the information from the server side such as the CPU load, the Disk load, the memory load and especially the number of access to GIS database. Considering this information, the agent can select a proper server by keeping load balancing dynamically so that it is possible to manage servers to share the workload effectively. Finally, the agent will reduce the probability that a client gets an error message when server fails to work at the Web sites. Thus, highly maintainable and expandable Web sites are continuously available to clients.

In addition, this paper proposes the prototype of Web-based Forest Fire Management system with an efficient user interface. It is possible to retrieve the attribute data in single database management system. Also, some thematic maps such as topographical map, forest type map and choropleth map can be presented on Web.

Web-based GIS has also attracted a lot of attention in the last few years since Jankowsku and Stasik (1997) introduced an internet-based GIS to make possible collaborative spatial decision-making via public participation. Keisler and Sundell (1997) presented an integrated geographic multi-attribute utility system with application to park planning. An extensive survey of applications and research issues for geographic information technologies in business is provided in Mennecke (1997). (Athanasios, 2000)

Based on these pre-research, some functional requirements should be considered before this system for the Internet is developed. First of all, the UI (user interface) has to be very logic as well as easy to use because the services are for the public ultimately. Secondly, a visually effective reporting functionality such as the result of query or statistical charts should be provided for users. Finally, this system should be developed to provide communication among users who have same interests. And it works not only as decision-making supporting system but also as information exchanging system. The rest of the paper is organized as followings. Section 2 presents the concept of load balancing. Section 3 discusses the network architecture and agent design. Section 4 discuses the systems design. Finally, conclusion remarks appear in section 5.

2. Concept of Load Balancing

Some large Web sites receive over a million hits per an hour. Therefore, they use a web cluster to support their services. A server cluster contains one domain with many servers, all of which have the same IP address and duplicate contents. Since each server's disk includes all the files that exist on the web site, all requests can be serviced using a mirroring technique.

As an Internet server, the computer can receive a dramatically different amount of Internet traffic, as the web sites grow more popular. However, a powerful computer may not be necessary initially, even though it is predicted that such a computer will be needed later. Also, installing a very expensive computer as a web server is just not economical when the hits to a site are only few. Moreover, an Internet server should not be down at any time because there is no nighttime on the Internet. Also, if a single computer is used as the server, Internet services are unavailable while it is checked for maintenance or broken.

With those reasons, a substantial amount of time should be needed for system backup, database restoration, and sanity checking, etc. Updating the server can also interrupt Internet services. In contrast, there is much more flexibility if a cluster server is provided. However, if there is no load balancing of requests, some servers even in a cluster can also have heavy web traffic.

Therefore, some load balancing methods have to provide clients with access to server IP addresses. As such, when the load balancing method detects that certain servers have a heavy load, the client can request to access to another server. This then allows the resources of server to be used more efficiently and prevents connection delays.

3. Network Architecture and Agent Design

In this section, the agent-based load balancing method is described and simulated considering Web Server clustering shared with Map Server architecture.

1) Network Architecture

In order to respond faster to a client's request, the way to select the best-suited server is considered as very important. Also, recently several mechanisms have been discussed to maintain load balancing. In this study, the load balancing mechanisms using dispatcher is introduced to react to a client's request in real time based on the information about load balancing such as the number of connection and between a Map Server and Web Servers.

The basic operation of load balancing method preformed on dispatcher, which is working with the Map Server, is shown in Fig. 1. As you see, Web Sever accesses to database through Map Server. The reason to use this architecture is to reduce processing load at Web Server.

Fig. 2 shows the message processing when client requests a desired map in Fig. 1. At this time, dispatcher selects the best-suited server

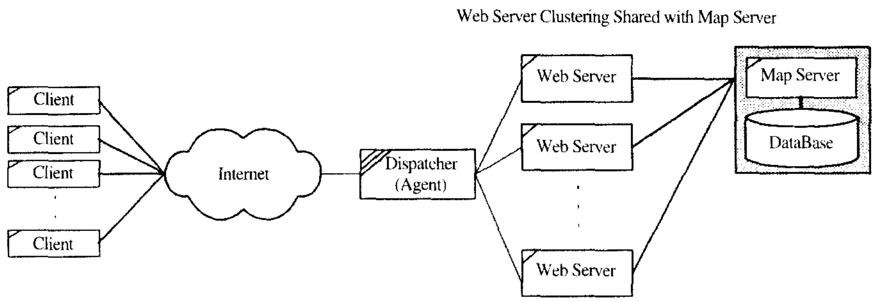


Fig. 1. Architecture of web server cluster shared with map server.

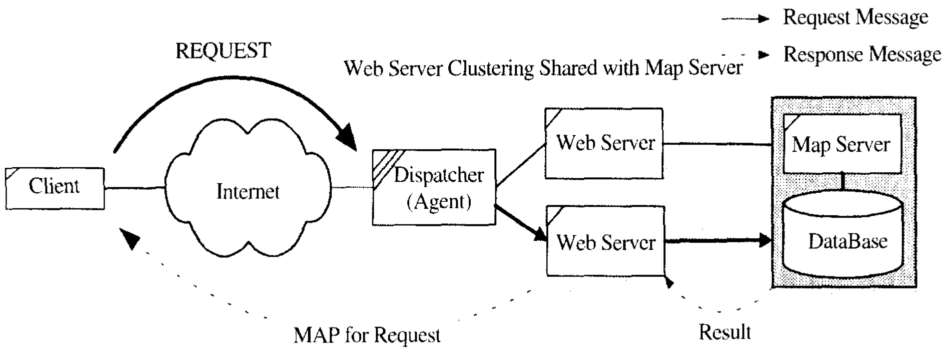


Fig. 2. Overview of messages operation.

based on the load information. Then, Web Server receives a desired map from Map Server and returns to client as the result of processing.

2) Agent Design

This paper proposed an ideal agent for Web-based GIS services. The agents proposed in this study are a new kind of agent that aims at prediction of the load and selection of the most proper Web Server in a Web Server cluster.

In this section, the concept of agent to reduce the Internet traffic is described. The architecture of agents' technology is outlined and the performance for this architecture is discussed. The policies of agents proposed in this paper bases their decision on the estimated load rates. Fig. 3 summarizes the additional software components needed for an efficient agent based on load-balancing method. The agent on a dispatcher has a load monitor and a load table. These monitors track the feedback from servers to avoid assigning requests to an overloaded server. When the agent is in learning mode, the data is generated as the load information. In order to compute the load, agent collects CPU load, memory load, disk using and the number of connection between Map Server and Web Server from each Web server.

This information is sent from Web Server to a dispatcher periodically. Agent on dispatcher becomes aware of this load information delivered from server at load monitor and gathers them at load table. As shown in Fig. 3, a set of components in one web server cluster is consisted in HTTPD, Load Checking and Connection Counter. The Load Checking tracks the server utilization of CPU load, memory load and disk using. The Connection Counter estimates the number of connections between Map Server and Web Server. That information estimated from Load Checking and Connection Counter is periodically provided to agent.

The basic operation rule is that dispatcher selects one sever using the load information when

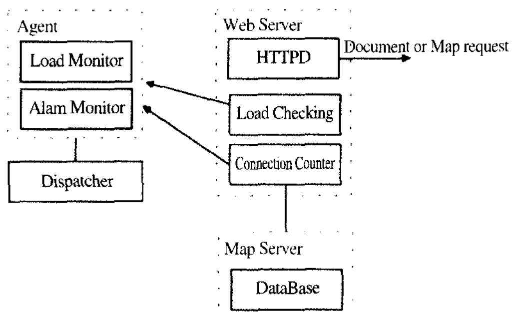


Fig. 3. Relation among web server and map server and agent.

a client requests a map. The selected server will be the one that have probably the lowest load value. If the proper Web Server is selected, the message is forward to server. If client's request message is requesting map, Web Server is accepting the result from Map Server then responding it to client. The formula to compute load to select the best-suited server for its web services is the following.

$$\text{Total\_Load} = \text{weight} \times (\text{cpu\_Load} + \text{memory\_Load} + \text{disk\_using} + \text{num\_Map\_Server\_connection}) \quad (1)$$

While cpu_load, memory_load, and disk_using mean the utilization of each Web Server side. num_Map_Server_connection means the number of connection between Map server and Web Server at each Web Server.

Here, the weight can be granted differently depending on each load. For the convenience of computing load, cpu_Load, memory_Load, disk_using, num_Map_Server_connection are only considered in this paper. The sum of weight is $1 (0 \leq \text{weight} \leq 1)$. The formula to grant Weight depending on each load is following.

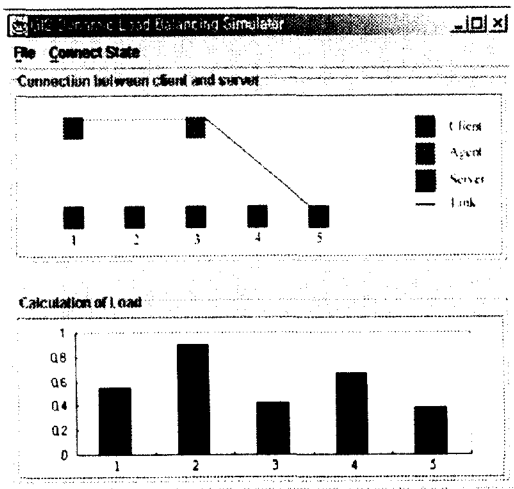


Fig. 4. The result of agent simulation.

$$\text{Total\_Load} = W_{cpu} \times \text{cpu\_Load} + W_{mem}\ \text{mem\_Load} + W_{disk} \times \text{disk\_using} + W_{conn} \times \text{Map\_Server\_connection} \quad (2)$$
$$(W_{cpu} + W_{mem} + W_{disk} + W_{conn} = 1)$$

Fig. 4 shows the result of simulation that agent selects the server having the lowest load value based on total load after agent learns load information. In the Fig. 4, there are two parts. The upper part presents that client's request is sent to the server that is selected by agent on dispatcher and the lower shows the chart for load information acquired from formula (2). Here, the weight of cpu_Load, memory_Load, disk_using, num_Map Server_connection are defined in 1/4, 1/4, 0, 1/2, respectively.

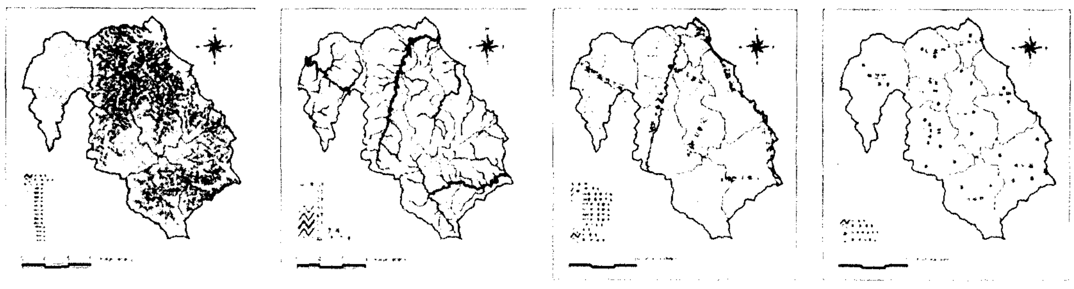
4. System Design

MO IMS is employed as the interactive GIS mapping server and Java is used as programming language for visual interface such as displaying map and reporting statistical chart and retrieving information from the central database.

1) Study Area and Materials

The study area is Sam-Chuck city where recently has frequent and large sized forest fire. For the spatial database, four layers were constructed based on 1/25,000 topographical map. There are contour layer with intervals of 100m, stream layer classified by degrees, road layer classified as highway, state road, local road and railroad, and administrative boundary layer with places of forest fire and government offices. Fig. 5 shows the spatial data set in study area.

In order to manage and prevent present forest fires, situation data of the past forest fire is



(a) (b) (c) (d)

Fig. 5. Spatial data sets in Sam-Chuck city, (a) contour line, (b) stream, (c) road, (d) administrative boundary with places of forest fire and government offices

considered as very important materials. For the attribute database, situation data of forest fire for a decade (1991 – 2001.8), which were constructed at Korea Forest Services and Korea Forest Research Institute, was used. Also, this data item is worked as retrieval condition in proposed system. Table 1 shows the capital item and detail item.

2) Web Server

Internet Information Services (IIS) 5 handles the primary process that manages requests from the client browser for the information such as HTML pages and maps. Depending on the type of requested data, IIS 5 will either fetch and return the data without calling on any other component or will co-operate with other components to generate the required response for the browser. (Jankowski, 1997)

As you see in Fig. 6, the architecture of this system is expected to have two kinds of audience; one is for the general users and the other is for specific users. It means the public is allowed to access without authentication while the other requires user verification. The purpose of this two-step architecture is expected to protect the data from illegal use by providing specific data items to those who are permitted. Also, this is supposed to classify appropriate type of service to appropriate users. Thus, people who want only simple information related to services can have brief theme map within seconds. Also, they don't have to lose their primary target during their web surfing. Otherwise, people who are verified can have more specific information such as not only desired theme map but also the result of various data retrieval. Of course, those users will wait for

Table 1. Capital item and detail item.

NO / Item	Item	Detail Item
1	Location	the name of cities or countries
2	Time	00:00-08:00 (dawn), 08:00-12:00 (morning), 12:00-18:00 (afternoon), 18:00-24:00 (night)
3	A day of the week	weekday (Mon-Fri), weekend (Sat-Sun)
4	Ownership	a private forest, a state forest
5	Forest type	a pine tree, a big cone pine, a kind of oak, a species of pine, a thicket of assorted trees, etc.
6	The loss	a won unit
7	Area	ha unit
8	Reason	by mountain-climbers, by burning up the rubbish, by visitors to his ancestral tomb, by setting on fire

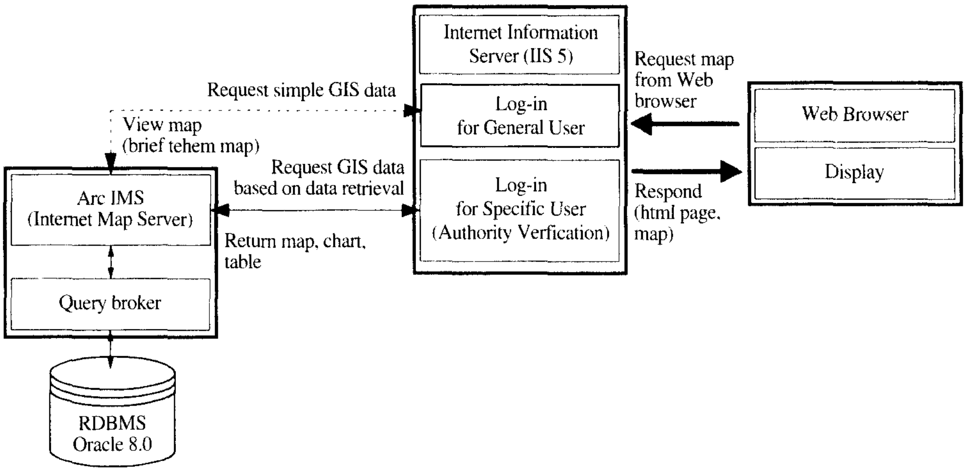


Fig. 6. System design of forest fire information management system.

a little bit much time for their request.

3) Map Server

To choose a suitable Map Server, users have to match a Map Server and an application in the following area: data format used for geographic data, Web Server platform, GIS functionality and direct data accessibility. (Yafang Su, 2000) This site has MO IMS obtained from ESRI site license. This Map Server will make Web GIS applications more flexible and capable than ever. MO IMS is the out-of-the-box Map Server available for multiple Windows-based system.

MO IMS is responsible for serving requests for maps from the client, and is the cornerstone of data retrieval. Requests from the client are received by IIS 5 and are routed to MO IMS for processing. This usually involves interaction between MO IMS and Oracle because all data are stored in the database and are retrieved when map is requested. Therefore, the developers should select an IMS product, which can best incorporate their organization's hardware and software environment and best perform their tasks. (Yafang Su, 2000)

4) Client Interface Design

This forest fire management system helps Map Server not only deliver data to the specific community but also make GIS available for general users through the Internet. This system is using Microsoft's Internet explorer as web browser. After logged on the system, user can access to GUI within seconds. Fig. 7 shows the general user interface of forest fire management system in the browser.

If user ID and password are verified, then they can meet data retrieval part and acquire the result as table form and chart. This interface may be generally simple and intuitive to even people who visit first time because it is based on the familiar Microsoft web browser format.

HTML is used to make multi-part interfaces and MO IMS is used to develop the client interface. Also, GUI is built in JAVA to retrieve and manage data and use through Internet. Of course, users interface with basic GIS capabilities such as zooming, panning, buffering, identifying,

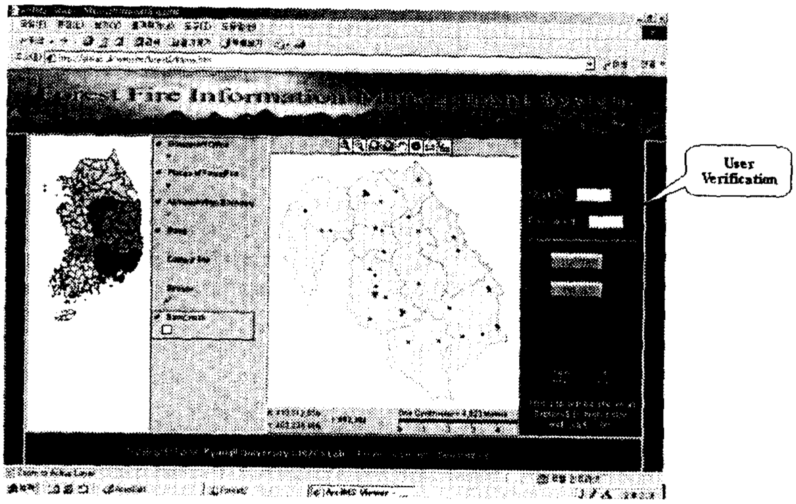


Fig. 7. GUI of forest fire information management system.

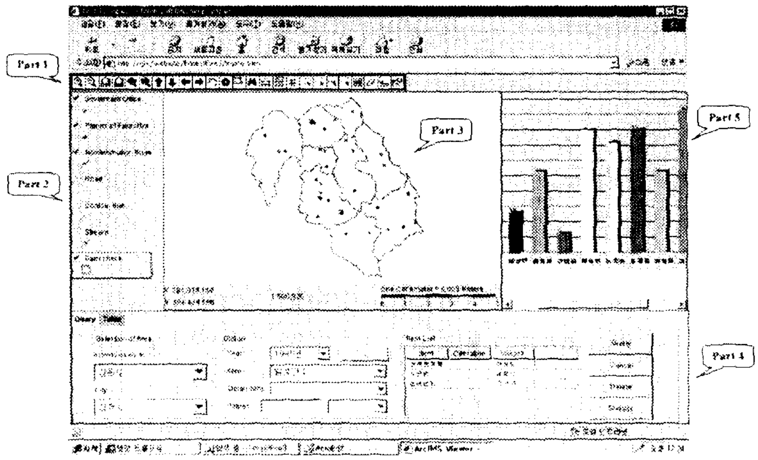


Fig. 8. GUI of system after user verification.

measuring, and finding can be executed over the World Wide Web.

The main objective of this homepage's GUI is to provide services for viewing the information in the database. After user verification, the five parts that make up the main client interface are shown as you see in Fig. 8. All parts are scrollable and resizable.

• **Part 1**: the standard toolbar of the system. Within this environment it is possible to zoom, pan, measure scaled distances, identify, find on the map.

• **Part 2**: the layer information of the map viewer. Here, spatial data information such as contour line, stream, road, places of forest fire, administrative boundary, and government offices in study area are shown as layers labels.

• **Part 3**: the main mapping frame of the system. For the GIS mapping, Arcview shape, Arc/Info coverage formats and the images related to forest fire can be displayed. For instance, here, the layers of government offices, an administrative boundary and places of forest fire are shown for Sam-Chuck city.

• **Part 4**: the retrieving functionality of the system. Information retrievalis based on the frequency, area, the loss, ownership, time, reason, date, and forest type about forest fire. For instance, the retrievaloptions can be spatially performed in Sam-Chuck city, Korea and temporally from 1999.1 to 1999.12. The result of this frame is available at the part 3.

• **Part 5**: the report window of the system. On the report window, most of requested information is provided to the user generally in the form of charts. For instance, here the result of part 4 is presented as a chart graph.

5. Conclusions

The information system implemented over Internet has a big demand to present information on Web within seconds without bottleneck. In this study agent-based load balancing is designed and simulated considering Web Server cluster shared with Map Server. In architecture of this system, agent watches load coming to each Web Server side and gathers it. Then, an agent is able to select the best-suited Web Server in the cluster by maintaining load balancing.

In addition, forest fire information management system implemented in this study has user-friendly interface. Through this GUI, user can retrieve information via database, see spatial data and use static report such as chart or table form.

Here, some of the expected merits from our system are briefly discussed. First of all, accessing to the system over Internet helps people easily obtain the results at any place and at any time. So, information of the system will support time-sensitive decisions at within a much shorter time. This will allow information at another country or state office with a click of a button.

In the future, the work done here can be extended for implementation in larger networks with agent performing more useful and efficient tasks.

References

Athanasios, K., Ziliaskopoulos, S., and Travis Waller, 2000. An Internet-based geographic information system that integrates data, models and users for transportation application, *Transportation Research Part C*, 8: 427-444.

Bouguttaya, A., M.Ouzzani, B. Benatallah, L. Hendra, and J. Beard, 2000. Supporting Dynamic Interactions among Web-Based Information Sources, *IEEE Transactions on knowledge and data engineering*, 12(5): 779-801.

Chris Gago, 1999. Scaleability, Availability andLoad-balancing for TCP/IP applications, *IBM Secure Way Network dispatcher Version 2.1*, White Paper.

Huang, G. Q., J. Huang, and K. L. Mak, 2000. Agent-based workflow management in collaborative product development on the Internet, *Computer-Aided Design*, 32: 133-144.

Jankowski, P., and Stasik, M., 1997. Design Considerations for space and time distributed collaborative spatial decision-making, *Journal of Geographic Information and Design Analysis*, 1(1): 1-8.

Kimmance, J. P., M. P. Bradshaw, and H. H. Seetoh, 1999. Geographic Information System (GIS) application to construction and geotechnical data management on MRT construction projects in Singapore, *Tunneling and Underground Space Technology*, 14(4): 469-479.

Keisler, J. M., and Sundell, R. C., 1997. Combining multi-attribute utility and geographic information for boundary decisions: an application to park planning, *Journal of Geographic Information and Design Analysis*, 1(2): 101-118.

Mennecke, B., 1997. Understanding the role of GIS in business: application and research directions, *Geographic Information and Design Analysis*, 3(1): 44-68.

Cardellini, V., M. Colajanni, and Philip S.Yu., 1999. DNS Dispatching Algorithms with State Estimators for Scalable Web-Server Clusters, *World Wide Web Journal Blather Science*, 2(2): 1-25.

Yafang Su, Joan Slottow, and Avi Mozes, 2000. Distributing proprietary geographic data on the World Wide Web-UCLA GIS Database and Map Server, *Computing & Geosciences*, 26: 741-749.