Remote Thin-Client Access to Spatial Database Systems[*]

Hanan Samet
František Brabec
Computer Science Department
Center for Automation Research
Institute for Advanced Computer Studies
University of Maryland
College Park, Maryland 20742
(301)405-1755
{hjs,brabec}@umiacs.umd.edu
www.cs.umd.edu/{~hjs,~brabec}

Abstract

Numerous federal agencies produce official statistics that are made accessible to ordinary citizens for searching and data retrieval. This is often done via the Internet through a web browser interface. If this data is presented in textual format, it can often be searched and retrieved by such attributes as topic, responsible agency, keywords, or press release. However, if the data is of spatial nature, e.g., in the form of a map, then using text-based queries is often too cumbersome for the intended audience. We propose to use the capabilities of the SAND Spatial Browser to provide more power to users of these databases. Using the SAND Spatial Browser allows users to define the spatial region of interest with greater specificity, instead of forcing them to retrieve data just for a particular location or a region with a predefined boundary. They can also make use of ranking which is the ability to retrieve data in the order of distance from other instances of the data or aggregates of data that are user-defined. Work is distributed between the SAND server and the individual clients for query evaluation, data visualization and data management. This enables the minimization of the necessary requirements for system resources on the client side while maximizing the number of connections one server can handle concurrently. Concrete experience with interfacing the SAND system with FedStats data is also discussed.

1. Introduction

Various governmental agencies enable ordinary citizens to access and search their official statistics electronically, via Internet web browsers. The data includes forecasts, projections, statistical tabulations, surveys, and other collected or derived data. Data can be retrieved by topic, responsible agency, keywords, and press release. In this paper we demonstrate how capabilities of the SAND Spatial Browser can be utilized to provide more power to these individual users. In particular, instead of being able to retrieve data just for a particular location or a region with a predefined boundary, users can define the spatial region of interest with greater

[*] This work was supported in part by the National Science Foundation under Grants EIA-99-00268, IIS-00-86162, and EIA-00-91474.

specificity. They can also make use of ranking which is the ability to retrieve data in the order of distance from other instances of the data or aggregates of data that are user-defined. The SAND system partitions the workload between the client and the server in such a manner that the user's experience with the system is interactive, with minimal delay between the user action and appropriate response. The design works around potential bottlenecks for the information transfer such as the limited network bandwidth or resources available on the client computer. To support multiple concurrent clients, limited resources on the server must also be considered.

There has been a substantial amount of research on the remote access of spatial data. The images are often presented in raster format. Work described in (Chang et al., 1997) examines into a client-server architecture for viewing large images that operates over a low-bandwidth network connection. It presents a technique based on wavelet transformations that allows the minimization of the amount of data needed to be transferred over the network between the server and the client. While the server holds the full representation of the large image, only a limited amount of data needs to be transferred to the client to enable it to display a currently requested view into the image. On the client side, the image is reconstructed into a pyramid representation to speed up zooming and panning operations. Both the client and the server keep a common mask that indicates what parts of the image are available on the client and what needs to be requested. This also allows dropping unnecessary parts of the image from the main memory on the server.

Other related work has been reported in (Potmesil, 1997). The author describes a client-server architecture designed to provide end users with access to a central data server. It is assumed that this data server manages vast databases that are impractical to be stored on individual clients. His work blends raster data management (stored in pyramids (Williams, 1983)) with vector data stored in quadtrees (Samet, 1990a; Samet, 1990b).

GeomNet (Barequet et al., 1999) is a result of research in the area of distributed geometric computations. The architecture is based on a number of servers where each server provides its computational power to run various geometric algorithms. Typically, a certain server could read data provided by the client via a specified protocol, run an algorithm whose implementation is available on this server, and return the results via the communication protocol again. This approach is clearly beneficial for certain types of tasks. In particular, experiments and development of new algorithms can be done quicker by utilizing existing implementations of algorithms needed for these new solutions. This system may not be as helpful for many production-strength solutions where performance is a concern.

The rest of this paper is organized as follows. We first give an an overview of SAND as the spatial database kernel system in Section 2. Section 3 discusses the client-server architecture employed to provide remote access to SAND-managed databases. Section 4 contains an example application using FedStats data, while Section 5 contains some concluding remarks.

2. SAND

SAND (Esperança and Samet, 2002) is a GIS system developed at the University of Maryland to deal with spatial and non-spatial data. It can handle two-dimensional data such as country boundaries, river paths, and city locations and facilitates the response to queries involving them such as finding the closest hazardous waste site to the border of a particular state. A major feature of SAND is the ability to find not just the closest hazardous waste site to a particular location or spatial object, but, instead, also to incrementally produce a list of all hazardous waste sites ordered by their distance from the particular location or spatial object.

In addition to aiding exploration of the data set by queries on spatial attributes, SAND also permits queries that involve the non-spatial attributes of the data. Taking our hazardous waste site example a bit further, we might have stored with each hazardous waste site some other attribute such as the level of pollutants or the nature of the dangerous chemicals that are present. Now, instead of just examining the hazardous waste sites that are near the California border, we could just look at the ones that have at least a particular level of pollutants. Thus we see that SAND aids in the exploration of the data set. Presumably such a capability would be useful to scientists exploring other data as well.

The class of queries currently implemented in the SAND Browser is restricted to spatial selections and distance semi-joins (Hjaltason and Samet, 1998). The user specifies queries by choosing the desired selection conditions from a variety of menus and dialog boxes. Spatial values can either be drawn on the appropriate display pane or be input by typing them in by filling forms. Query results can be either displayed interactively using the *First* and *Next* buttons or saved in relations for use in subsequent SAND queries.

One of the key features of the SAND system is the support of the *ranking* operation. It enables finding objects in the order of their proximity to other objects. Ranking can be viewed as a spatial analog to sorting. For example, it is not possible to order a collection of points in the same sense that a collection of numbers can be sorted in ascending or descending order. However, it is possible to *rank* points in ascending or descending order of distance from a given point or spatial object.

In SAND, ranking is performed incrementally(Hjaltason and Samet, 1999). This means that once the parameters for a ranking operation are given, the first (and closest) tuple is returned as soon as possible. This is a better solution than initially sorting the entire data set especially when we may only need a few of the closest elements rather than all of the elements in which case sorting the set would have been a good idea. This characteristic is extremely important in an interactive environment.

3. SAND Client-Server Approach

Traditionally, common Geographic Information Systems (GIS) such as ArcInfo from ESRI (Arc, 2002) and spatial databases are designed to be stand-alone products. The spatial database is kept on the same computer or local area network from where it is visualized and queried. This architecture allows for instantaneous transfer of large amounts of data between the spatial database and the visualization module so it is perfectly reasonable to use large-bandwidth protocols for communication between these two. There are however many applications where a more distributed approach is desirable. In these cases, the database is maintained in one location while users need to work with it from possibly distant sites over the network (e.g., the Internet). These connections can be far slower and less reliable than local area networks and thus it is desirable to limit the data flow between the database (server) and the visualization unit (client) in order to get a timely response from the system.

One approach has been adopted by numerous web-based mapping services (MapQuest (Map, 2002a), MapsOnUs (Map, 2002b), etc.). Their goal is to enable remote users typically only equipped with standard web browsers to access the company's spatial database server and retrieve information (in the form of maps) from them. The solution presented by most of these vendors is based on performing all the calculations on the server side and transferring only bitmaps that represent results of user queries and commands. Although the advantage of this solution is the minimization of both hardware and software resources on the client site, the resulting product has severe limitations in terms of available functionality and response time (each user action results in a new bitmap being transferred to the client).

The original design of SAND was as a stand-alone system. In the Digital Government scenario, end users would need remote access to SAND's functionality, and the best way to provide this is to make SAND available over the Internet. Our newly developed client-server version of the system (Figure 1) allows the actual database engine to be run in a central location maintained by spatial database experts, while end users acquire a Java-based client component that provides them with a gateway into the SAND spatial engine.

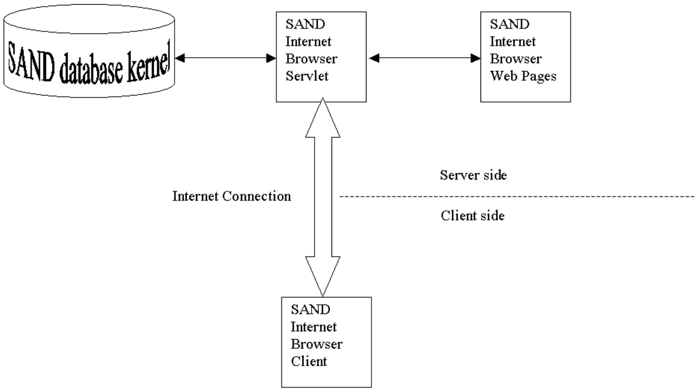

SAND database kernel
SAND Internet Browser Servlet
SAND Internet Browser Web Pages
Server side
Internet Connection
Client side
SAND Internet Browser Client

Figure 1: SAND Internet Browser — Client-Server architecture.

Our client is more than a simple image viewer. Instead, our client operates on vector data which allows the client to execute many operations such as zoom in/out or locational queries locally. In essence, a simple spatial database engine is run on the client. This database keeps a copy of a subset of the whole database whose full version is maintained on the server. This is a concept similar to 'caching' as known from other areas of computer science. Notice, however, that this is a more complex architecture than what is known as "web caching". In our case, the client acts as a lightweight server in that given raw data, it evaluates queries and provides the visualization module with objects to be displayed. It initiates communication with the server only in case it does not have enough data stored locally. This is different from web caching where static web pages or images are kept on the client temporarily in order to allow them to be reused in case in the near future the very same page or image needs to be displayed. In web caching, the client does not attempt to create new content from the data it stores.

Since the locally run database is only updated when additional or newer data is needed, our architecture allows the system to minimize the network traffic between the client and the server when executing the most common user-side operations such as zooming and panning. In fact, as long as the user explores one region at a time (i.e., he or she is not panning all over the database), no additional data needs to be retrieved after the initial population of the client-side database. This makes the system much more responsive than the web mapping services discussed above. Due to the complexity of evaluating arbitrary queries (i.e., more than window queries needed for database visualization), we do not perform user-specified queries on the client. All user queries are still evaluated on the server side and the results are downloaded onto the client for displaying. However, assuming that the queries are selective enough (i.e., there are far fewer elements returned from the query than there are elements in the database), the response delay is usually within reasonable limits.

In the FedStats environment, special provisions need to be made to enable Federal agencies to import their own data into the system. This also includes the ability to build indexes on it to facilitate efficient query responses. In this client-server scenario, all the data is stored on the server side in SAND's internal format so steps need to be taken to move the agencies' datasets from their location to the server location and to convert the data from the format in which it was gathered and/or delivered to SAND into a format that SAND can understand and use.

4. FedStats Collaboration and an Example Application

FedStats (Fed, 2001) enables ordinary citizens to access and search official statistics of numerous Federal agencies without knowing in advance which agency produced them. We have been involved in collaboration with FedStats in order to provide more power to users of FedStats by utilizing the SAND Spatial Browser.

As an example, we used two Excel files corresponding to EPA-regulated facilities that have Chlorine and Arsenic, respectively. For each file, we had the following information available: EPA-ID, Name, Street, City, State, Zip Code, Latitude, Longitude, followed by flags to indicate if that facility is in the following EPA programs: Hazardous Waste, Wastewater Discharge, Air Emissions, Abandoned Toxic Waste Dump, and Active Toxic Release. Each of the programs was represented in the Excel file by a separate column and the appearance of the entry 'Y' indicates that the facility participates in the EPA program.

We converted this data to a SAND relation having the spatial attribute 'location' corresponding to the latitude and longitude, which was stored using a PMR quadtree (Nelson and Samet, 1986; Nelson and Samet, 1987) for points. We added an attribute 'tuple-id' to distinguish between the tuples in the relation thereby making them unique.

There are several ways of implementing the nature of the program in which the facility participates.

1. Have an attribute called 'program' that contains the names of the programs in which the facility participates. The drawback of this solution is that the result is not in first normal form (e.g., (Elmasri and Navathe, 2000)).

2. Make use of five Boolean attributes of the form 'is-program' to indicate if the facility participates in the program (e.g., 'is-hazardous-waste', etc.). The drawback of this solution is that users must formulate their queries in terms of this construct and its Boolean value which requires them to know how the Boolean value is specified. In particular, there are many ways of specifying Boolean values (e.g., '0' and '1', 'false' and 'true', 'no' and 'yes', etc.).

3. Have one tuple for each program in which the facility participates, and use the field 'program'. This is fine when each facility participates in just one or a few programs. The drawback is that in the case of queries on the basis of the values of other attributes, a facility will be retrieved as many times as the number of programs in which it participates. This is a classic problem in databases known as the duplicate problem (Aref and Samet, 1992; Aref and Samet, 1994). In the SAND Spatial Browser, this drawback is alleviated by using the 'group by attribute-name' mechanism, which retrieves all tuples having the same value of the 'attribute-name' attribute simultaneously. In our example, we can use the 'EPA-ID' attribute. Note that it is not advisable to use the 'name' attribute in this case as there is no guarantee that two different EPA facilities with the same name do not exist.

We chose the third solution as it appeared to be the easiest to use and it did not require the user to know much about the underlying implementation.

Some queries that can be handled include:

1. Find all EPA-regulated facilities that have Arsenic, and which participate in the ”Air Emissions” program in states from Georgia to Illinois, alphabetically.

2. Find all EPA-regulated facilities that have Chlorine, and which participate in the ”Air Emissions” program that lie within the state of Arkansas or 30 miles within its border.

3. Find all EPA-regulated facilities that have Chlorine, and which participate in the ”Air Emissions” program that lie within 30 miles of the border of Arkansas (i.e., both inside and outside Arkansas).

4. For each EPA-regulated facility that has Arsenic, find all EPA-regulated facilities that have Chlorine which are closer to it than to any other EPA-regulated facility that has Arsenic. In order to avoid reporting a particular facility more than once, we use the ‘group by EPA-ID’ mechanism. Note that the result of this operation is analogous to a discrete Voronoi diagram where the sites are the EPA-regulated facilities that have Chlorine.

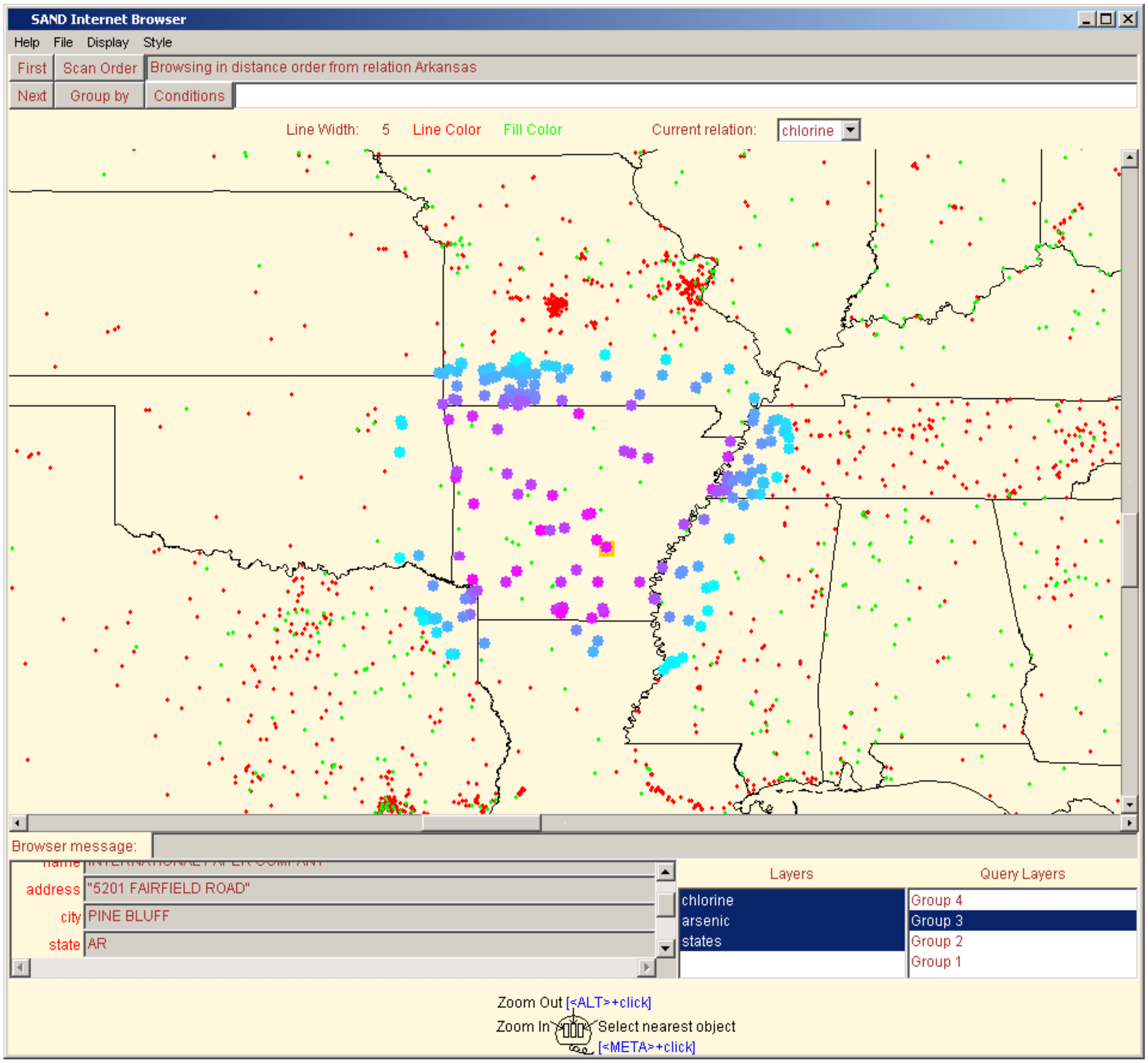
5. For each EPA-regulated facility that has Arsenic, find all EPA-regulated facilities that have Chlorine, and which participate in the Air Emissions program that are closer to it than to any other EPA-regulated facility that has Arsenic. In order to avoid reporting a particular facility more than once, we use the ‘group by EPA-ID’ mechanism. Note that the result of this operation is analogous to a discrete Voronoi diagram where the sites are the EPA-regulated facilities that have Chlorine and that participate in the Air Emissions program.

Figure 2 illustrates the output of an example query that finds all Chlorine sites within a given distance of the border of Arkansas. The sites are obtained in an incremental manner with respect to a given point. This ordering is shown by using different color shades.

While the import of data in Excel format as outlined in the above example is satisfactory for occasional work, the process can get cumbersome and unreliable when accessing data from different sources in different formats. In order to access multiple data sources in real time reliably, it is desirable to look for another mechanism that would support data exchange by design. The XML protocol (XML, 2002) and its Document Type Definitions (DTDs) have emerged to become virtually a standard for describing and communicating arbitrary data, and GML (GML, 2002) is becoming increasingly popular for exchange of geographical data. We are currently working on making SAND XML-compatible so that the user can instantly retrieve spatial data provided by various agencies in the GML format via their web services and then explore, query or process this data further within the SAND framework.

5. Concluding Remarks

We have presented our research results in the area of remote spatial database access using SAND. We have shown how this system could be utilized within the Digital Government framework to allow ordinary citizens to access government-published spatial data more efficiently than if only static viewing was available. We have outlined the architecture of the system and introduced its individual elements. Finally, we have provided an example where we have shown how adding a map browsing capability to FedStats by making use of the SAND Spatial Browser increases the information value of the data for end users.



Browsing in distance order from relation Arkansas

Line Width: 5

Current relation: chlorine

name INTERNATIONAL PAPER COMPANY

address "5201 FAIRFIELD ROAD"

city PINE BLUFF

state AR

Layers: chlorine, arsenic, states

Query Layers: Group 4, Group 3, Group 2, Group 1

Figure 2: Sample output from the SAND Spatial Browser — Large dark dots indicate the result of a query that looks for all chlorine sites within a given distance from Arkansas. Different color shades are used to indicate ranking order by the distance from a given point.

References

(2001). Fedstats: The gateway to statistics from over 100 U.S. federal agencies. `http://www.fedstats.gov/`.

(2002). Arcinfo: Scalable system of software for geographic data creation, management, integration, analysis, and dissemination. `http://www.esri.com/software/arcgis/arcinfo/index.html`.

(2002). Extensible markup language (xml). `http://www.w3.org/XML/`.

(2002). Geography markup language (gml) 2.0. `http://opengis.net/gml/01-029/GML2.html`.

(2002a). Mapquest: Consumer-focused interactive mapping site on the web. `http://www.mapquest.com`.

(2002b). Mapsonus: Suite of online geographic services. `http://www.mapsonus.com`.

Aref, W. G. and Samet, H. (1992). Uniquely reporting spatial objects: yet another operation for comparing spatial data structures. In *Proceedings of the Fifth International Symposium on Spatial Data Handling*, pages 178–189, Charleston, SC.

Aref, W. G. and Samet, H. (1994). Hashing by proximity to process duplicates in spatial databases. In *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM)*, pages 347–354, Gaithersburg, MD.

Barequet, G., Duncan, C. A., , Goodrich, M. T., Bridgeman, S. S., and Tamassia, R. (1999). GeomNet: geometric computing over the internet. *IEEE Internet Computing*, 3(2):21–29.

Chang, E., Yap, C., and Yen, T. (1997). Realtime visualization of large images over a thinwire. In Yagel, R. and Hagen, H., editors, *Proceedings IEEE Visualization'97 (Late Breaking Hot Topics)*, pages 45–48, Phoenix, AZ.

Elmasri, R. and Navathe, S. B. (2000). *Fundamentals of Database Systems*. Addison-Wesley, Reading, MA, third edition.

Esperança, C. and Samet, H. (2002). Experience with SAND/Tcl: a scripting tool for spatial databases. *Journal of Visual Languages and Computing*. To appear.

Hjaltason, G. R. and Samet, H. (1998). Incremental distance join algorithms for spatial databases. In Hass, L. and Tiwary, A., editors, *Proceedings of the ACM SIGMOD Conference*, pages 237–248, Seattle, WA.

Hjaltason, G. R. and Samet, H. (1999). Distance browsing in spatial databases. *ACM Transactions on Database Systems*, 24(2):265–318. Also Computer Science TR-3919, University of Maryland, College Park, MD.

Nelson, R. C. and Samet, H. (1986). A consistent hierarchical representation for vector data. *Computer Graphics*, 20(4):197–206. Also *Proceedings of the SIGGRAPH'86 Conference*, Dallas, TX, August 1986.

Nelson, R. C. and Samet, H. (1987). A population analysis for hierarchical data structures. In *Proceedings of the ACM SIGMOD Conference*, pages 270–277, San Francisco.

Potmesil, M. (1997). Maps alive: viewing geospatial information on the WWW. *Computer Networks and ISDN Systems*, 29(8–13):1327–1342. Also *Hyper Proceedings of the 6th International World Wide Web Conference*, Santa Clara, CA, April 1997.

Samet, H. (1990a). *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*. Addison-Wesley, Reading, MA.

Samet, H. (1990b). *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, MA.

Williams, L. J. (1983). Pyramidal parametrics. *Computer Graphics*, 17(3):1–11. Also *Proceedings of the SIGGRAPH'83 Conference*, Detroit, July 1983.