A Jigsaw Query Processing Technique for Web GIS

Sung-Hee Kim[1], Chi-Young Yoo[1], Young-Geol Lee[2], Jong-Hoon Kim[2], Jae-Dong Lee[3], Jae-Hong Kim[4] and Hae-Young Bae[1]
[1] Dept. of Computer Science & Engineering, Inha University, Korea
[2] KGI Co., Ltd., Korea
[3] Dept. of Computer Science, Dankook University, Korea
[4] Dept. of Computer Science, Youngdong University, Korea
g1991237@inhavision.inha.ac.kr

Abstracts

As computing environments for Internet has been expanding, researches on Internet GIS has been actively made. The omnipresence and easiness of Web make the research being focused on Web GIS. Web GIS is classified into Server-side Web GIS and Client-side Web GIS. The former can easily be implemented but raise the overload of the server, the latter lessens the overload of the server, but raises some problems. To solve these problems, this paper proposes a Jigsaw query processing technique that processes user queries dynamically. For an evaluation, the detailed experiments such as map browsing, layer controlling, point query, and region query are conducted comparing the proposed technique with two other existent methods. Based on the result of experiments, the proposed technique can give the user fast response and increase system performance.

1. Introduction

Much recent attention has focused on developing GIS functionalities in the Internet, World-Wide Web, and private intranets. The Web GIS holds the potential to make distributed geographic information available to a very large worldwide audience. The Web GIS will make it possible to add GIS functionality to a wide range of network-based applications in business, government, and education. Many experiments are now under-way in Web GIS and related map server applications for interactive cartography. One of the important areas of innovation involves "pay-for-use" mapping and GIS services[1, 5, 6].

The challenge of Web GIS lies in creating software systems that are platform independent and run on open TCP/IP-based networks, that is on any computer capable of connecting to the Internet (or any TCP/IP-based network) and running a Web browser[9,12].

Many strategies can be employed to add GIS functionalities to the Web. Server-side strategy allows users(clients) to submit requests for data and analysis to a Web server. The server processes the requests and returns data or a solution to the remote client. This strategy can be easily implemented but raise the problems. Firstly, every request must be returned to the server and processed. Secondly, responses must then be returned to the client across the Internet. Performance will be affected by the bandwidth and network traffic on the Internet between the server and client particularly when responses involve transferring large files. Lastly, applications do not take advantage of the processing power of the user's own "client" computer, which is used merely to submit a request and display the response. Client-side strategy allows the users to perform some data manipulation and analysis locally on their own machines. Client-side strategy lessens the overload of the server but currently provides only some browsing-based geographic information, and raises some problems. Firstly, the response from the server may involve transferring large amounts of data as well as applets, causing delays. Secondly, large and complex datasets may be hard to process on the client if it is not very powerful. Thirdly, complex GIS analytical routines may run more slowly on the client if it is not very powerful. Lastly, users may not have the training needed to employ the data and analysis functions properly[5, 6, 9, 11, 12].

To solve these problems, this paper proposes a jigsaw query processing strategy that processes user queries dynamically selectively in the server or client based on system environments. The proposed system classifies the query into client class query, server class query or client-server class query. Client class query is processed in client itself. Server class query is transferred to the server and processed in the server and the result returns into the client. Client-server class query is processed as follows. The data required to the query is transferred from the server into the client. Next, the client processes the query with its original data and the transferred data. The proposed system uses some meta data on client-stored data and the communication speed between the client and the server to classify the class of the query.

This paper is organized as follows. In section 2, we investigate existent Web GISs. We propose the architecture of the proposed system in section 3 and describe the data storage structure of the client in section 4. We describe the management of meta data required for query class selection in section 5 and propose a jigsaw query processor. In section 7, we analyze the proposed system and last make a conclusion in section 8.

2. The Query Processing Strategy for Web GIS

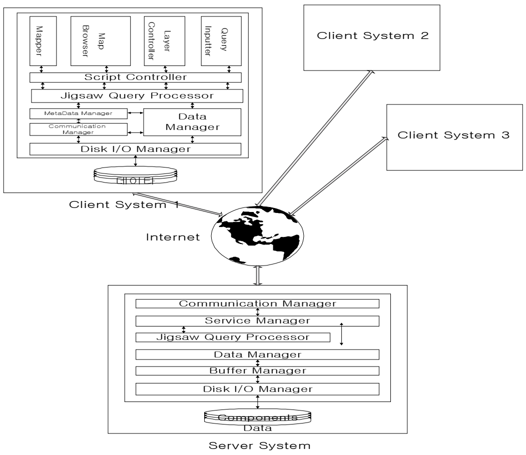
In this section, the existent query processing strategy for Web GISs is investigated. Existent Web GISs can be classified into server-side Web GIS or client-side Web GIS according to query processing strategy[9].

Server-side Web GIS configures the client by using Form-based HTML document to get query request. The client takes the query input and transmits the query to the server. The server processes the query and returns the processing result as HTML document with GIF and JPEG image to the client. Most server-side Web GISs extend traditional GIS using CGI and provide geographic information to Internet user[9, 12].

Client-side Web GIS configures the client so that it can process user query on web browser. The client processes map browsing facilities such as map scale-up, map scale-down, map move and so on, and provides user interactive geographic information. According to the configuration of the client, client-side Web GIS can be classified into Plug-in, Java applet and ActiveX control client-side Web GIS[7, 11].

3. The architecture of the proposed Query Processing Technique

The proposed query processing technique in this paper is a jigsaw architecture between server-side Web GIS and client-side Web GIS. It provides a jigsaw query processor. It processes a given query in the client, the server or mixed approach with considering the dynamic system environments. The system architecture is as following [Figure 1].



[Figure 1] The System Architecture

The server stores geographic data and client-module components, which configure the client. In addition, it takes the requests from the client, processes the requests and then transmits the result. As shown in [Figure 1], the server consists of communication manager, server manager, query processor, data manager, buffer manager, disk I/O manager. Communication manager provides communication facilities with the client. Service manager analyzes the request from the client and calls appropriate sub module. Query processor manipulates the query from the client. Data manager manages geographic data and includes spatial index. Buffer manager and Disk I/O manager get or put the required data from or to stable storage such as Disk device.

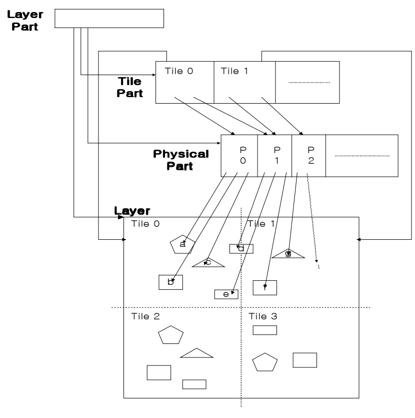
The client system provides on Web browser various geographic information through various facilities such as Map browsing, Map display, Layer control, point query processing and region query processing. As shown in [Figure 1], the client consists of Mapper, Map browser, Layer controller, Query inputter, Communication manager, Data manager and Disk I/O manager. Mapper displays map information on the display device of the client. Map browser provides the facilities such as scale-up, scale-down, move, drag and region-magnifying of the map on the mapper. Layer controller adds or deletes layers of the mapper. Query inputter takes the parameters of point query or region query. Script controller integrates various components with built-in script of the browser and so configures various client systems.

Jigsaw query processor manipulates the query with meta data. Meta data manager manipulates the information on the client-stored data and the speed between the server and the client. Communication manager transfers data and component from the server and transmits the query to the server. Data manager manages client-stored data and includes a spatial index. Disk I/O Manager processes disk input or output of geographic data and meta data.

4. Data Storage Management

4.1 Tile-divided Data Storage Management

When existent Web GISs transfer geographic data from the server to the client, they use whole data or layer as the unit of data transfer. Therefore, though the client requires only some part of geographic data, the server must transfer unnecessary data to the server. The client waits until the client finishes not only the required data but also unnecessary data, which decrease the total system performance due to very large initial set-up time. Therefore, the proposed system manages geographic data by unit of layer. In addition, the system manages the layer by unit of tile that is a small rectangle area and transmits geographic data from the server to the client by the unit of tile. Such a tile-based data management method can decrease the large initial delay time of the client in existent Web GIS. The proposed system manages layer as following [Figure 2].



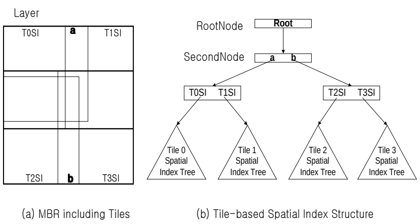
[Figure 2] Tile-based Data Management Method

As shown in [Figure 2], the system manages geographic data in layer part, tile part and physical part. Layer part stores whole information on the layer. Tile part manages stores tile information and mapping information of a tile into spatial objects that are included in the tile area. Physical part stores spatial objects in the disk page. In [Figure 2], spatial object a, b, c, e is included in tile 0, spatial object f, g is included in tile 1. Because spatial object d crosses on tile 0 and tile 1, spatial object d is managed in both tile 0 and tile1. Page P0 stores spatial object a, b and c. Page P1 stores spatial object d, e, f. Page P2 stores spatial object g. Tile 0 manages some information on spatial objects which are included in its own area by pointing P0 and P1.

4.2 Tile-based Spatial Index Structure

Web GIS must provides spatial index structure to support general point query and region query. There are many researches on spatial index structures such as R*-tree[2], Cell trees[4], R-tree[2], Grid file[4], and so on. It is general decision that R-tree is easy to implement and efficient to point and region query. Therefore, the proposed system processing point query or region query using R-tree.

When the proposed system transmits geographic data from the server to the client, it transmits corresponding spatial index. The reason of this approach is that the system can provide fast system response to client query without additional communication load with the server. The following [Figure 3] shows the dual spatial index structure proposed in this paper.



[Figure 3] Spatial Index Structure Using Tile Spatial Index

As shown in [Figure 3], this system calculates MBR(Minimum Boundary Rectangle) including each tile consisting of the target layer at first. This work determines T0SI area, T1SI area, T2SI area and T3SI area and then determines corresponding tile spatial index tree. After this work, above level index structure is configured by using the lower level tile index tree. This dual spatial index tree requires some more storage space compared to original R-tree. However, the client system can calculate the exact transfer amount from the server. So, it is used in jigsaw query processor to be described in section 6.

5. Meta Data Management Structure

5.1 Meta Data Manager on Client-stored Data

Most existent Web GIS do not store the transferred geographic data from the server in the client[3, 5, 8, 10]. When the client requires the data transferred in the past client session after the client exits, this approach causes re-transmission problem of the data. Especially, the Internet which is communication base in Web GIS is very slow relatively to LAN, thereforre re-transmission problem decreases the system performance largely.

In order to solve the above problem, this system stores the transferred data from the server to the client. In addition, the client manages meta data on the client-stored data by unit of layer to assist jigsaw query processor. The meta data on geographic data is established when the geographic data of a layer is transferred to the client. A meta data is established using layer part and tile part of a transferred geographic data. In addition, as geographic data of a layer is transferred, the meta data on the layer is updated. The meta data on client-stored data that is managed in the client is as following [Table 1].

As shown in [Table 1], MaxScale and MinScale represents the scale of a layer that limits the maximum and minimum scale on mapper. The client determines by comparing TileCnt and ClientTileCnt whether all geographic data of a layer is stored in the client. AverObjectsize is an average size of all objects consisting of a layer and this is used to calculate expected query processing cost of server class query. TileMetaTable is a set of meta data on each tile. RegionQuerySelectivityTable is a meta data to be used in expecting the number spatial objects satisfying the region query condition. The record of TileMetaTable is as following [Table 2].

[Table 1] Meta Data on Layer

Name	Type	Meaning
LayerName	String	Layer name
Origin	Coordinate	Coordinates of layer origin
TotalArea	MBR	Total area of layer
TileSize	MBR	Size of a tile
MaxScale	Real	Maximum map scale of layer
MinScale	Real	Minimum map scale of layer
TileCnt	Integer	Number of tiles consisting of the layer
ClientTileCnt	Integer	Number of tiles stored in the client
PageSize	Integer	Size of Page storing the physical data
AverObjectSize	Real	Average object size
TileMetaTable	Table	Meta information on each tile
RangeCnt	Integer	Size of RegionQuerySelectivityTable
RegionQuery SelectivityTable	Table	Selectivity Information on Region query

[Table 2] Meta Data on Tile

Name	Type	Meaning
TileNo	Integer	Tile Number
TileArea	MBR	Area information of this tile
ClientStoread	Bit	Tag indicating if this tile is stored in the client or not
DataPageCnt	Integer	Number of spatial objects
IndexPageCnt	Integer	Number of the pages of spatial index

In [Table 2], ClientStored is a meta data on whether the tile is stored in the client or not. DataPageCnt is the number of disk pages to be used in storing spatial objects in this tile. Using this information, the client can determine how many disk pages to be transferred from the server when it receives a tile. IndexPageCnt is the number of index pages to be used in storing spatial index part corresponding to this tile. The client finds the spatial index file size corresponding to this tile.

5.2 Meta Data Manager on Average Communication Speed

The communication speed between the server and the client can be expected using past the communication speed information. The variation of bandwidth of Internet is large in the system using time zone. For an example, in daytime the communication is slow due to much population, but in nighttime, the communication is fast due to relatively small population. Therefore, this system manages not only recently communication speed information but also communication speed information of each time zone in order to expect the exact communication speed between the server and the client. Given the number of managed past communication speed, information is 3 and the number of time zone is 5, the meta data on communication speed is as following [Table 3].

[Table 3] Meta Data on Communication Speed

Name	Type	Meaning
TimeZoneCnt	Integer	Number of time zone dividing a day
Time0	Real	Average communication speed in first time zone
Time1	Real	Average communication speed in second time zone
Time2	Real	Average communication speed in third time zone
Time3	Real	Average communication speed in fourth time zone
Time4	Real	Average communication speed in fifth time zone
LatestCTCnt	Integer	Number of managed recent communication speed
LatestCT1	Real	Most recent communication speed
LatestCT2	Real	second recent communication speed
LatestCT3	Real	third recent communication speed

6. Jigsaw Query Processor

6.1 Query Class

This system supports not only map browsing facilities, layer control facility that is generally supported in existent Web GIS, but also point query and region query processing facilities which is essential to general GIS. Map browsing provides the facilities such as scale-up, scale-down, move, region magnification, panning of map displayed on mapper. Layer control facility adds or deletes layers on mapper. Point query retrieves spatial objects the area of which include a given point coordinate. Region query retrieves spatial objects that satisfy the given topological condition on given area and the area of the spatial object. This system processes point query and region query using dual spatial index structure.

This system classifies query into following three classes according to query processing method.

(1) Client class : Queries which are processed by using only client-stored data without the connection of the server

(2) Client-Server class: Queries which are processed as follows. At first, the client receives the required geographic data from the server. Next, the client processes the query using the received data and client-stored data.

(3) Server class: Queries which are processed as follows. At first, the client sends the query itself to the server. The server processes the query and returns the result into the client.

The queries to be supported in this system can be classified as following [Table 4].

[Table 4] Query Class Table

Query	Sub Query	Query Class		
		Client Class	Client-Server Class	Server Class
Map Browsing	Scale-up	O	O	
	Scale-down	O	O	
	Move	O	O	
	Region Magnifying	O	O	
	Panning	O	O	
Layer Control	Layer add	O	O	
	Layer delete	O		
Point Query	Point query	O		
Region Query	Disjoint	O	O	O
	Meet	O	O	O
	Overlap	O	O	O
	Covered_By	O	O	O
	Covers	O	O	O
	Inside	O	O	O
	Contains	O	O	O
	Equal	O	O	O

As shown in [Table 4], map browsing query can be classified as client class or client-server class. The reason of this is that the geographic data required for map browsing must be exists in the client to display the map information on the mapper. Layer control query also has the required data as the map browsing, so it can be classified into client class or client-server class.

To make a point query, the user takes a point on mapper and the geographic data of mapper must be exists in the client. Therefore, point query must be client class.

Region query can belong to one of all three classes. Region query retrieves spatial objects that satisfy the given topological condition on given area and the area of the spatial object. If the spatial objects in the area to be retrieved are all stored in the client, then the query is client class. If not, the query can be client class or server class.

6.2 Jigsaw Query Processing Algorithm

The proposed query processing techniques manipulates given query according to its query class. The jigsaw query process classifies the query class of the given query using following <Algorithm 1>.

```
<Algorithm 1> Query Class Determining Algorithm
Input: Given Query and its parameter
Output: Query class

    SpatialQueryClassify()
    {
        if ( input query is layer delete or point query ) then
            return client class query
        end if
        if( input query is map browsing query ) then
            if ( all required data are in client system ) then
                return client class query
            else if
                return client-server class query
            end if
        else
            if (all required data are in client system) then
                return client class query
            else if ( the client-server class processing cost is
                    less than server class one ) then
                return client-server class query
            else
                return server class query
            end if
        end if
    }
```

As shown in <Algorithm 1>, the jigsaw query processor classifies point query and layer delete query as client class. Map browsing query and layer add query is client class if all required data exist in the client and they client-server class if not.

If all required data exist in the client, region query is classified as client class. If not, the jigsaw query processor compares the expected costs of client-server class processing and of server class and it classifies the query as query class requiring small cost. Such an expected processing cost will be calculated as follows.

If region query is classified as client-server class, the jigsaw query processor must receive the required geographic data and corresponding spatial index part from the server. Then, it processes the region query. Therefore, the expected processing costs will be defined as following Formula 1.

$$\text{ClientServerClassCost} = \text{Cdt} + \text{Cit} + \text{Ccp} \qquad \text{(Formula 1)}$$

In Formula 1, Cdt is the cost required to receive the required data from the server. Cit is the cost required to receive the corresponding spatial data from the server. Ccp is the cost required to process the region query in jigsaw query processor in the client. To simplify the above formula, this system considers only time factor in calculating the cost.

If region query is classified as server class, the jigsaw query processor must send the query itself to the server. Then, the server processes the query and returns the results into the client. Therefore, the expected processing costs will be defined as following Formula 2.

$$\text{ServerClassCost} = \text{Cqt} + \text{Csp} + \text{Crt} \qquad \text{(Formula 2)}$$

In Formula 2, Cqt is the time to required to send the query into the server. Csp is the time required to process the query in the server. Crt is the time required to send the processing result into the client.

In Formula 1 and Formula 2, the value of Ccp is similar to one of Csp. This is because the processing step of the client is same as one of the server and the difference of processing times between the server and the client will be very small and can be ignored. Therefore, Formula 1 and Formula 2 can be simplified as follows.

$$\text{SimpleClientServerClassCost} = \text{Cdt} + \text{Cit} \qquad \text{(Formula 3)}$$

$$\text{SimpleServerClassCost} = \text{Crt} + \text{Cqt} \qquad \text{(Formula 4)}$$

Using the above formulas, the proposed jigsaw query processor determines the query class of given query that can be classified as client-server class or server class.

7. Performance Evaluation

7.1 Performance Evaluation Environments

The performance of the proposed jigsaw query processing technique is analyzed compared to existent Web GISs – Client-side Web GIS and Server-side Web GIS. At first this simulation generates random query pattern that consists of map browsing, layer controlling, point query and region query. Then, it compares the time needed in processing the query and storage space to be required as to architectures of each Web GIS. By doing so, we analyze the performance of the proposed system as to existent Web GISs.

The simulation system is implemented under Window NT 4.0. The system environments are as following [Table 5].

[Table 5] Simulation System Environments

Item	Value
Platform	IBM PC Compatible
CPU	Pentium 400
Main Memory	128 MB
HD Disk	8GB
OS	Window NT 4.0
Language	Visual C++ 6.0

The simulation factors are as following [Table 6].

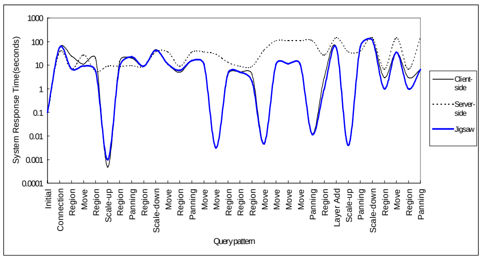
[Table 6] Simulation Factors

Item	Values
DB Size	880KB
Layer Size	220KB
Layer Number	4
Tile Number per Layer	25
Page Size	1KB
Page Number of Layer	220

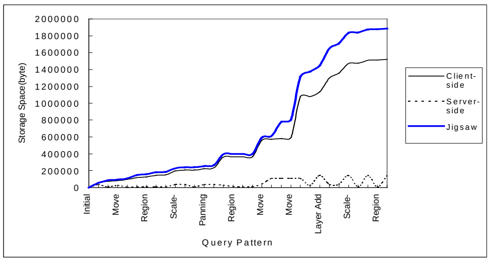
As shown in [Table 6], this simulation will be performed as to four layer geographic data. In general, the size of layer will range from several kilobytes to several megabytes. This simulation fixes the size of a layer as 220 kilobytes. Each layer consists of 25 tile areas. Assumed that the average Internet communication speed between the server and the client are 1KB and the user can endure 9 seconds to be received on tile area geographic data. The size of disk page is 1KB. Therefore, the number of pages in a layer will be 220.

7.2 Simulation Results

As shown in [Figure 4], the proposed jigsaw query processing technique required extra storage space in the client. However, as the client continues to process queries, it shows good performance.



(a) System Response Time



(b) Required Storage Space

[Figure 4] System Performance

8. Conclusion

This paper proposed a Jigsaw query processing technique which supports Jigsaw query processor to perform as in the client itself, in the server or in cooperated method. In addition, this system supports point query and region query that are not supported generally in existent Web GIS. The proposed Jigsaw query processing technique considers client-stored data and the average communication speed between the server and the client and classifies the given query as client class, client-server class or server class. By doing so, it give the user fast response.

In future, because Web GIS run on Internet, which is very weak to security, the research will be required on incorporating the security into Web GIS.

Reference

[1] Abel, D. J., "Spatial Internet Marketplaces: A Grand Challenge?," Proc. of the 5th International Symposium on Large Spatial Databases, pp.3-8, 1997.

[2] Beckmann N., Kriegel H. P., Schneider R., Seeger B., "The R*-tree:An Efficient and Robust Access Method for Points and Rectangles," Proc. of ACM SIGMOD, Vol. 19, pp.322-331, 1990.

[3] Gunther, O., "Efficient Structures for Geometric Data Management," Lecture Notes in Computer Science #337, Springer-Verlag, 1988.

[4] Nievergelt, J., Hinterberger, H., Sevcik, K. C., "The grid file: An adaptable, symmetric multikey file structure," ACM Trans. on Database System, Vol. 9, No. 1, pp.38-71, 1984.

[5] O.Guenther, A.Buchmann, "Research Issues in Spatial Databases", ACM SIGMOD RECORD, Vol.19, No.4, 1990, pp61-68

[6] Peng, Z., "An Assessment of Internet GIS Development," Proc. of the 1997 ESRI User Conference, 1997, http://www.esri.com/library/userconf/proc97/PROC97/TO550/PAP526/P526.HTM

[7] Plewe, Brandon, Stuber, Chris, "TIGER Map Service," http://tiger.census.gov/cgi-bin/mapbrowse-tbl

[8] R.H.Guting, "An Introduction to Spatial Database Systems", VLDB Journal, Vol.3, No.4, 1994, pp357-399

[9] R. Peng: "An Assessment of the Development of Internet GIS. Proceedings of the ESRI User Conference, 1997

[10] VOLKER GREDE, "Multidimensional Access Methods", ACM Computing Surveys, Vol.30, No.2, June 1998

[11] Yeong-Seob Cho, Hong-Yeon Kim, Jong-Hoon Kim, Hae-Young Bae, "The Design and Implementation of Componentized Web-enabled GIS," ICIMU 98, September, 1998

[12] YeongSub Cho, HongYeon Kim, JaeHone Kim, HaeYoung Bae, "GIS Solutions Supporting WWW Environments," Korea Information Science Society, Review, Vol. 16, No. 3, pp.28-33, 1998