SRE: Search and Retrieval Engine of TerraScope Earth Science Information System

Enna Z. Coronado Pacora
Advisor: Dr. Manuel Rodríguez Martínez

Advance Databases Management Group
Electrical and Computer Engineering Department
University of Puerto Rico, Mayagüez Campus
Mayagüez, Puerto Rico 00681-5232
Enna.Coronado@ece.uprm.edu

Abstract

This paper presents the Search and Retrieval Engine (SRE) that we have developed as part of TerraScope that is, a Web based Earth Science Distributed Management System with a Peer to Peer Architecture where multiple data sources which can be integrated into a coherent system, and support the execution of different queries that gather useful data, with different formats and/or characteristics from each distributed data source.

The main contribution of SRE is the spatial search engine using Java Servlet technology and the Peer–to–Peer architecture that makes possible the transformation of the server into a client at any moment, and allows the communication among Web Servers.

Keywords

SRE, R-tree, Peer to Peer, Client Access Servlet, Data Broker Servlet and Information Gateway Servlet.

1. Introduction

In today's world we have the necessity of obtaining information in a quick, precise and instantaneous way, regardless the geographical localization of the given information. Distributed Database systems that already exist have experienced an important growth in the volume of data to process. Current applications such as Geographic Information Systems, multimedia and other, impose some demands on the efficiency of query processing.

MOCHA is a novel database middleware system designed to interconnect hundreds of data sources distributed over a wide area network [Rodriguez00], but have a problem of scalability as new sites are added to the system and a single site must manage system-wide interactions, while the *TerraScope's Search and Retrieval Engine* is based in peer to peer architecture.

The *MARIPOSA* distributed database management system is a research prototype developed at the University of California at Berkeley [Stonebraker96]. *Mariposa* addresses fundamental problems in the standard approach to distributed data management. However, *Mariposa* requires data to be removed from their existing server applications, and re-ingested into the federated database engine.

TERRASERVER is on a simple client-server model [Driftwood00]. TerraServer needs the data of interest to be fetched from remotes sites into the client site where most of the processing occurs, this kind of approach was used before in the MERCURY project [Itsc03]. The data processing options are limited to the client of the server site, but the SRE has more options for data processing such as the proper movement of data between sites. Then, the data can be filtered at the source sites before the required items could be moved over the network.

Peer – to – Peer (P2P) Systems provide an attractive alternative to client – server architectures [Franklin96]. It allows us dynamic and distributed search, manage and storage of information in distributed form, distributed and parallel processing. **SRE** defines a scenario where interaction between web servers is intrinsically *P2P* and guarantees security for transmission of the information.

Our research, emerge from the need to know how to recover and visualize graphical/textual information from images that are stored in multiple data sources

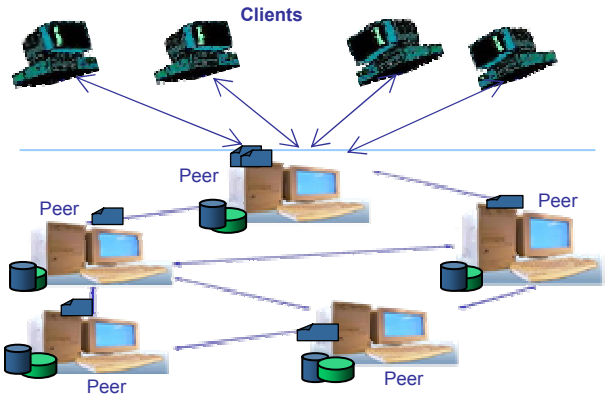
In this paper we present the *Search and Retrieval Engine* **(SRE)** as part of **TerraScope** with a Peer-to-

Peer architecture where multiple data sources can to be integrated in a coherent system. With SRE we can support the execution of different queries that make possible to gather information that are stored in each data center, with different formats or characteristics. Also, it's designed to support spatial indexes, parallel processing of images, distributed data recovery and visualization of images. It does not require existing data sources to be removed, since it can be run on top of existing storage servers.

We present in Section 2 a brief Architectural Overview of SRE. Section 3 presents Message with XML and finally Section 3 presents the conclusions.

2. Architectural Overview of SRE

SRE is based in Java Servlet Technology and on a decentralized Peer-to-Peer (P2P) architecture, where the clients can customize their own view of the system to define the remote data and computational services they wish to access. They can control which of their services and data products they are willing to share with others.



Picture No. 1. Peer – to – Peer Architectures

Picture No. 1, our P2P application will have the following characteristics:

- *To discover other partners*.

The application should be able to find other applications that share its information.

- *To consult partners for content*.

The application consults its peers about the content.

- *To share content with other partners*.

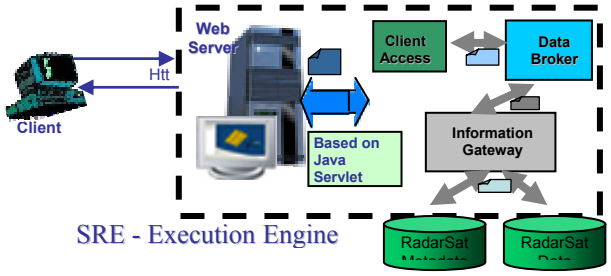
The application can share information after having been discovered.

The components of the architecture are as follows:

2.1 Web Server

The Web server is the most important part in any Internet site. In our case, we required a Web server that could handle Java Servlet technology. Our choice was the Apache Tomcat because its own JWSDP (Java Web Service Developer Pack) offers us the best ratio of easy use/communication cost.

The Search and Retrieval Engine **(SRE)** has a group of servlets that communicate between them through XML messages. Each Servlet has a different function that will process each request sent by the client through the GUI (Graphical User Interface) obtaining a final result from the diverse data sources that our server has access.



Picture No. 2. Architecture of SRE

For example, **Picture No. 2,** consider what happens when a user requests the following through GUI (Graphical User Interface): "Get all the Radarsat Images of CENSSIS acquired between 03/16/2001 and 03/20/2001". Then our **SRE** subsystem will act as a data and resource broker for the application, where the client application will send this query to:

a. **The *Client Access Servlet (CAS)*** is on charge of interacting directly with requests made by the client. The requests are encoded in XML format which is interpreted and validated by the CAS using a proper parsing function and then send over to the next Servlet, the Data Broker.

b. **The *Data Broker Servlet (DBS)*** receives the client's requests, processes the information, controls and coordinates the communication that exists between the others Data Broker Servlets of the other Web-Server (peers), determining a strategy to solve the queries in a quick and precise way. It verifies what data sources have been invoked for the clients, consulting in a catalog elaborated in XML. Once the ***DBS*** has found groups of sites, it will negotiate the access or will send the query request to be resolved.

c. **The *Gateway Servlet (GWS)*** will receive the query request that the Client sends through the ***DBS***, allowing access to the database and extracting the significant information according to the queries

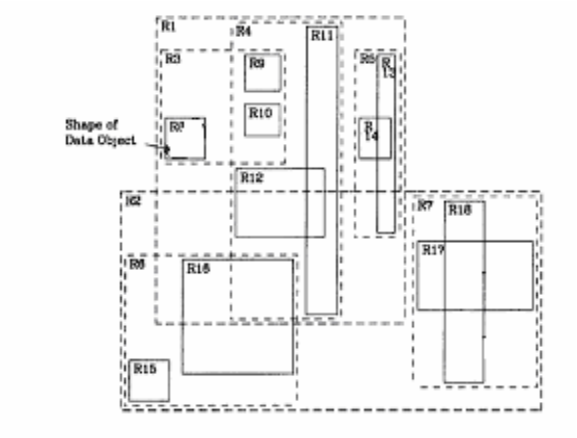
received, creating a new XML document with the final result containing the respective data and metadata.

2.2 Database System

In our design we have implemented databases created with PostgreSQL because allow us to use R-tree indexes in order to obtain a more effective access to a large volume of spatial data. Our database is composed by several tables where the data of each image available will be stored.

- Handling of spatial data

We use the R-tree index method because covers areas on a multidimensional space that are not too well represented by locations points [Corral02, Williams82, Mohan86]. Our database allows us to build R-tree indexes in order to place spatial searches and get an efficient response from the spatial query.



Picture No. 3. Overlap relationship

Picture No.3 shows the overlapping relationships that can exist between spatial searches. Spatial searches are represented in picture 2 as rectangles. Spatial searches are achieved using spatial indexes or polygons that overlap in a given range. Those queries require the use of a spatial index because will result extremely difficult in a system that single support B-tree. The use of the operator overlap in index R-tree will allow us to bring all images that are overlapped inside a selected range, and finally queries recursive allows us to bring the images belonging to a certain geographical point.

- Database Access

To access the database, SRE used the JDBC (Java Database Connectivity) interface that supports the execution of SQL sentences. The main advantage of the JDBC is the interoperation between multiples data sources.

- Specifics – Query execution

SRE has some methods that will make possible for the user to search information of any image obtaining its respective data and metadata.

For example:

SELECT *
FROM swaths
WHERE '(90,50),(70,50),(90,30),(70,30)' && bound
AND start_date = '03/14/2001'
AND end_date = '03/28/2001;

The queries can be expressed in terms of:

- *Spatial Data*. It is an exact query that looks for spatial objects determined by their location.. The queries can be restricted by date or time period.
- *Sensor type or some other characteristics*.
- *Geographic area.*
- *Data Sources*.

3. Message Passing with XML

The communication among the servlets uses XML messages. For example:

Schema of the request in XML. Schema of the request in XML. This schema is an XML document containing all the requests placed through the GUI. The corpus of the XML file is composed by detailed information such as query type, sources, dates and searching coordinates.

```
<?xml version="1.0"?>
<terrascope>

   <query_type>1</query_type>
   <source>
         <source1>TCESS</source1>
         <source2>CENSSIS</source2>
    </source>
    <coordinates>
         <longitud1>98.4000</longitud1>
         <latitud1> 20.4873</latitud1>
         <longitud2>68.4000</longitud2>
         <latitud2> 10.0000</latitud2>
    </coordinates>
    <start_date>12/10/200</start_date>
    <end_date>12/20/2000</end_date>

</terrascope>
```

Schema of the response in XML. Schema of the response in XML. This schema is an XML document containing all the responses founded in either

local or remote databases. The corpus of this XML file is composed by important data for the user such as a url where the image is stored, condition, date, image coordinates, source and other data that will be send to the GUI and checked by the user.

```
<?xml version="1.0"?>
<terrascope>
  <TCESS>
    <id_swath>22787_1</id_swath>
<url>http://icarus.ece.uprm.edu/~ecoronado/images/
22787_1.jpg</url>
     <condition>Ascending</condition>
     <resolution>F1</resolution>
     <date>Mar 16 2000</date>
     <time>23:16:12</time>
     <bound>
        <longitud1>6.614653N</longitud1>
        <latitud1>75.504684W</latitud1>
        <longitud2>6.411222N</longitud2>
         …
    </bound>
  </TCESS>
  <CENSSIS>
    <id_swath>22922_1</id_swath>
<url>http://icarus.ece.uprm.edu/~ecoronado/images/
22922_1.jpg</url>
    <condition>Descending</condition>
    <resolution>F1</resolution>
    <date>Mar 16 2000</date>
    …..
    …..
  </CENSSIS>
</terrascope>
```

4. Conclusions

SRE is a distributed query execution engine based on Java Servlet Technology and a peer to peer architecture where multiple data sources can be integrated in a coherent system that can handle search operations, spatial queries, and other operations on data sets stored on different data sources. We have created the databases using PostgreSQL because it allows us to create spatial indexes that facilitate the search and access at the same time.

Our **SRE** has a group different Servlets that communicate among them using XML messages. Each peer site submitting a query request it will indicate the kind of services it expects, where SRE will satisfy those requests. Each peer site will be running the following servlets: Client Access Servlet, Data Broker Servlet and Gateway Servlet

REFERENCES

[Rodriguez00] Manuel Rodríguez and Nick Napol Roussopoulus., **"MOCHA:** A Self - Expandable Database Middleware System for Distributed Source Dates", Technical ReportUMIACS-TR 2000-05, CS-TR4105, University of Maryland, January 2000.

[Stonebraker96] Stonebraker M., Aoki P., Devine R., Litwin W. and Olson M., "**Mariposa:** A New Architecture for Distributed Data**",** University of California Berkeley, California.

[Stonebrake96] Stonebraker M., Aoki P., Pfeffer, A., Sah, A., Sidell, J., Staelin, C. and Yu, A., "**Mariposa:** A Wide-Area Distributed Database System", in *VLDB Journal*, (1996).

[Driftwood00] Microsoft's **TerraServer** http://driftwoodkey.com/LocalStuff/terraserver.htm

[Itsc03] **Mercury** Search Engine http://mercury.ornl.gov/esip/

[Franklin96] Franklin, M.J., Jonsson. B.T. and Kossmann, D. Performance Tradeoffs for Client-Server Query Processing. in *Proc. ACM SIGMOD Conference*, Montreal, Quebec, Canada, 1996, 149-160.

[Corral02] Corral A., **"Tésis Doctoral:** Algoritmos para el Procesamiento de Consultas Espaciales utilizando R – Tree. La Consulta de los Pares Más Cercanos y su Aplicación en Bases de Datos Espaciales**"**, Almería January - 2002.

[Williams82] Williams, R., Daniels, D., Haas, L., Lapis, G., Lindsay, B., Ng, P., Obermarck, R., Selinger, P., Walker, A., P.W., and Yost, R. R*: An Overview of the Architecture. (RJ3325), 1982.

[Mohan86] Mohan, C., Lindsay, B.G. and Obermarck, R., Transaction Management in the R* Distributed Database Management System. *TODS*, *11* (4). 378-396, 1986.