GEOBARN: A Practical Grid Geospatial Database System

Zhou HUANG[1,2], Yu FANG[1], Xuetong XIE[3], Mao PAN[2]
[1)]*Institute of Remote Sensing & GIS, Peking University*
Beijing, P. R. China 100871
[2)]*Key Laboratory of Orogenic Belts and Crustal Evolution, Ministry of Education, Peking University*
Beijing, P. R. China 100871
[3)]*National Satellite Ocean Application Service*
Beijing, P. R. China 100081
huangzhou@pku.edu.cn

***Abstract*—Recently, more and more geospatial data are generated and distributed with the promising data acquirement techniques like satellite imaging and digital cartography. So, it is an important issue to effectively make use of these huge geospatial data resources. This means that effective data processing technologies need to be developed to support distributed query and analysis operations onto the distributed resources. The emergence of grid computing technology offers a good solution to accomplish this goal. To combine the grid computing technique with geospatial database we design and implement a GGDS (Grid Geospatial Database System) prototype named GEOBARN. Key modules and detailed implementation methods of GEOBARN are discussed in this paper. By using the GEOBARN system, both reliable data management and effective distributed geospatial query can be achieved.**

***Index Terms*—Distributed database, Grid computing, GIS**

I. INTRODUCTION

Along with the wide use of information technology, the geospatial applications have attracted more and more attention [1]. In GIS (Geographic Information System) research realm, the processing of the huge amount of geospatial data distributed in network is one of the most concerned topics. The emergence of grid computing provides a good solution. Currently there are some related works. Xue [2] pointed out that grid computing is the best way to implement Digital Earth strategy and solve the problems it involved. Shen [3] analyzed the architecture of Grid GIS application scenarios on distributed image processing. The common structure of grid computing - OGSA (Open Grid Service Architecture) [4] was brought forward through absorbing the advantages of Web Services technologies. The Globus corporation, which is the largest research community on grid computing, has developed Globus Toolkit 4.0.5 as a standard architecture of grid based on OGSA [5].

Existing researches mainly concentrate on the synthetic architecture design of Grid GIS, and some conceptual analyses. However, few research papers discussed the detailed implementation approaches to effective geospatial data processing in grid. Grid is deemed as "super computer" since it offers a suite of techniques and standards for sharing the data resources and computational resources distributed everywhere. GGDS helps to achieve effective geospatial data processing and analysis. This paper mainly discusses how to implement GGDS. The architecture of GGDS and detailed implementation methods were presented, and then, an actual GGDS named GEOBARN was accomplished as well.

II. THE CONCEPTUAL ARCHITECTURE OF GGDS

To accommodate the grid framework OGSA, we design the typical structure of GGDS shown in Figure 1. Four layers, including geospatial resource layer, grid protocol layer, geospatial grid extension layer and user layer, construct a complete GGDS.

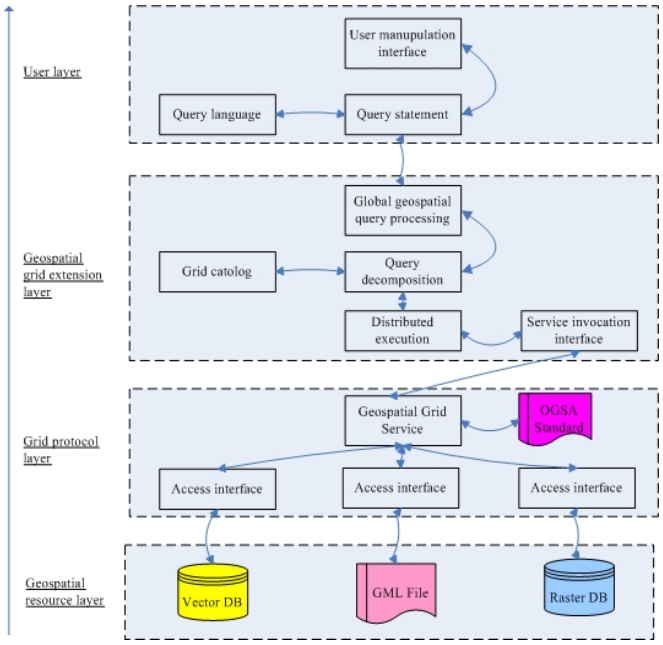

Figure 1. The conceptual architecture of GGDS.

Geospatial resource layer. The resource layer is composed of a number of separated geospatial data resources distributed on different sites in different forms. This layer is used for geospatial data storage deployment. The representation of geospatial data in the backend can be various. For example, ORDB (Object Relational Database) products like PostgreSQL, Informix, Oracle or even GML (Geography Mark-up Language: an XML compatible format) might be adopted for geospatial data storage.

Grid protocol layer. This layer provides basic implementation of standard grid protocols, such as resource finding, monitoring and messaging between different sites in the grid system. In OGSA this layer is incorporated with Web Services technologies. Therefore, standards like

Digital Object Identifier 10.4316/AECE.2009.03002

SOAP/WSDL are used here. All local geospatial database functions are wrapped into standard grid service interfaces. Through this layer the various geospatial data resources could be accessed using a uniform method.

Geospatial grid extension layer. This middle layer mainly refers to processing the user-defined geospatial query. Usually the users submit global geospatial query without knowing the distribution information of various resources. So this layer is designed to parse the global geospatial query, generate equivalent distributed sequences according to grid catalog that records geospatial resources' distribution information, and then execute them via parallel invocation onto various local geospatial grid services in network. In the following system design section we shall introduce implementation approaches to this layer in detail.

User layer. The user layer is on the top of GGDS. Firstly, a friendly graphic interface should be provided to users to submit request. Then an appropriate geospatial query language should be designed or selected to organize query statements.

III. DESIGN AND IMPLEMENTATION

Based on the conceptual architecture of GGDS, we implemented an actual prototype system named GEOBARN. The system architecture was shown in Figure 2. The modules are designed and developed according to the aforementioned four layers.

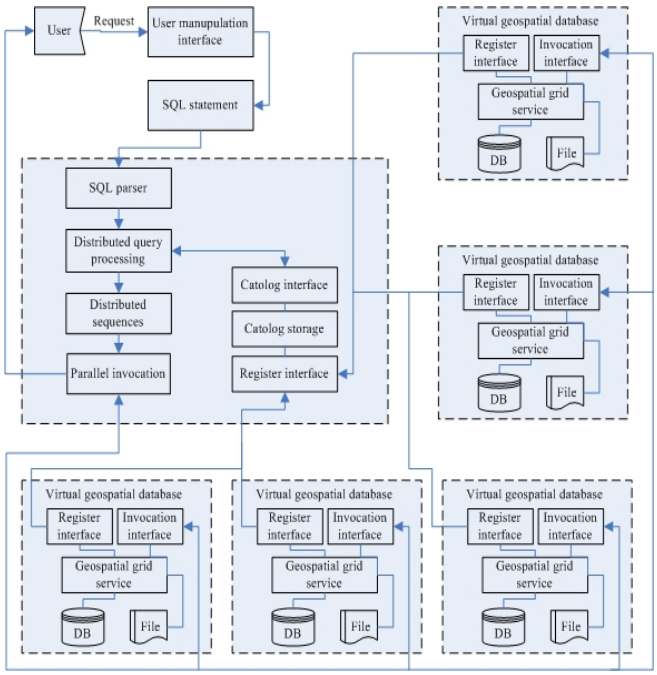

Figure 2. System architecture of GEOBARN.

Among these four parts, the geospatial grid extension layer is our research focus because the others could be accomplished via some traditional mature software tools. For example, grid protocol layer can be accomplished by common grid operation system products like Globus Toolkit 4.0.5; user layer can be developed by Swing components (Java graphic interface); as for the resource layer, PostgreSQL, Oracle and GML tools are prepared for geospatial data storage; standard web service can easily be built and deployed by Apache Axis. So, this paper focuses on those modules which cannot be accomplished by conventional method. These mainly include geospatial query language, grid catalogue and distributed geospatial query processing.

A. *Geospatial query language*

Geospatial query language is the user interface for expressing geospatial query task in GGDS. Current field researches lack one uniform geospatial query language in grid, but all of them could be included in three categories.

Those compatible with SQL. This method is completely compatible with SQL standard, using added geospatial operations and geospatial functions to manipulate the geospatial objects [8]. Those extending SQL. This approach used to extend SQL grammar for supporting geospatial semantic. Those expressed by GML. This method named GridGML was firstly brought forward by Shen [3], who uses GML to express the geospatial query task composed of a set of services and related operations.

Since the ORDB was widely adopted to store and manage the geospatial data, the first approach is more practical than others. Usually it is called GSQL (Geographic Structured Query Language) [9], e.g., in the GSQL statement *"select county.name from river,county where crosses(river.geometry,county.geometry)"* the function '*crosses*' implies the geospatial semantic "county crossed by river".

B. *Grid catalog*

In general, the common protocol LDAP (Lightweight Directory Access Protocol) is often used to set up global catalogue in distributed system [10]. Although the LDAP seems to be one good approach to implement global catalogue in grid systems, there are still some intrinsic disadvantages [11]. Firstly, the LDAP is designed without considering the geospatial operation on the catalogue level, e.g., user may want to find data services which serve in specified geographical MBR (Minimum Bounding Rectangle). Common LDAP based grid catalogue like Globus MDS (Monitoring and Discovery System) cannot support such geospatial judgment functions. Secondly there is no synchronization interface for LDAP based products. Often several catalogue servers are set up in a geospatial grid system, so there is a need for synchronizing the catalogue data between different servers [12].

Then, we developed a new relation model based solution to the grid catalogue for GGDS. The proposed grid catalogue mainly contains three components, including catalogue storage, catalogue interface and catalogue update.

Catalogue storage. It refers to the storage strategy of catalogue data, where we used relational tables to store the catalogue items. Catalogue interface. The catalogue interface is invoked by distributed geospatial query processing module. It offers descriptive information on what the distributed geospatial query processor needs [13], e.g., site list or service list using specified geospatial data source. And, the extern interface supports geospatial operation and optimization, which means that the optimal site list and service list could be obtained by user defined geographical MBR. Catalogue update. This module is quite important since the site list of GGDS might be variational, which means that this module should deal with the situations of

site join/exit grid system, should update and synchronize the global grid catalogue through adding or deleting some records.

C. *Distributed geospatial query processing*

In GEOBARN, distributed geospatial query processing algorithm is based on the heuristic principles combining with the "semi-join" strategy. As mentioned in the query language section, GSQL is the manipulation interface for geospatial query and analysis. So the distributed query processing is a process of GSQL equivalent transformation.

To reduce data transmission and to enlarge the number of parallel operations, we conclude five basic heuristic rules in the distributed geospatial query processing: execute the pre-select operations as soon as possible; execute the project operations as soon as possible; avoid to execute only Descartes accumulate and combine it with selection and project operations; make the parallel operations at maximum count; select the plan with less data transmission.

Based on the aforementioned heuristic rules and considering the characteristics of geospatial query, we brought out a new distributed geospatial query processing algorithm named HHOA (Hybrid Heuristic Optimization Algorithm) [15], which is briefly described as follows:

TABLE I. HHOA ALGORITHM

```
Algorithm: SUBS_Generate(query)
Input: the global geospatial query string submitted by user
Output: final result
{
  Step1(Build the query tree):
    Carry out the lexical and syntax analysis to build one query
    tree;
  Step2(Generate the pre-selection strategy):
    According to the components of the query tree, attract the
    separated conditions which could be executed firstly without
    need of data transmission;
  Step3(Generate the join strategy):
   Check if join operation appears in the global query
   {
    Case "True"
        Semi-join strategy is taken account, the less data
       transmission as the chief goal is the principle to generate the
       optimal plan. The characteristics of huge volume geospatial
       data are taken account into the selectivity factor calculation;
        break;
    Case "False"
       Skip to Step 4 directly;
   }
  Step4(Union the useful temporary results):
    The site with the most volume of temporary results is selected
    as the final result site, which means that other useful temporary
    results should be sent to it;
  Step5(Confirm the final plan and execution):
     The execution plan mainly contains three steps (Step 2 to 4),
    execute them on the separated sites and get the final result.
}
```

In addition, we brought forward EDP (Equivalence Distributed Program) [16] to formulate the sub-queries, demonstrating the execution flow, sub-query statement, service name, running site, etc. To adapt the RPC and SOAP mechanism, the EDP file is formatted in XML as well.

TABLE.2 A SIMPLE INSTANCE OF EDP FILE ("select * from river": Fetch all data from source 'river' distributed on site 'gis4g01' and site 'gis4g02')

```
<EDP>
<SECTION_EXEC>
  <FLOW>
   <STCQL COPYTO="T0_1@gis4g02" INTO="T0" RUNON="gis4g01"
SERVICE="Query Execution”>SELECT * INTO T0 FROM river;</STCQL>
<STCQL COPYTO=" " INTO="T1" RUNON="gis4g02" SERVICE=" Query
Execution ">SELECT * INTO T1 FROM river;</STCQL>
  </FLOW>
  <STCQL COPYTO=" " INTO="T_result" RUNON="gis4g02" SERVICE="
Query Execution ">SELECT * into T_result from T0_1 UNION select * from
T1;</STCQL>
</SECTION_EXEC>
<SECTION_RESULT>
  <RESULT host="gis4g02" result ="T_result" />
</SECTION_RESULT>
</EDP>
```

IV. EXEPRIMENT AND DISCUSSION

A. *Test Environment*

For demonstrating the application of GEOBARN, we used it onto ten sites in WAN (Wide Area Network), with nine data sources distributed according geographical range. On each site several basic geospatial data services were accomplished, like query execution, data transmission and transformation between GML and relational table. The total data volume exceeds 20 GB with about 210,000 polygons, 12,000 lines and 6,000 points. The sample query "*select county.name, county.geometry from county, rail where Crosses(rail.geometry,county.geometry) and rail.flux>800*" execution process and final result could be shown as Figure 3 (Friendly visualization user interface was accomplished as well, through which the users could see the data source distribution, result visualization and even the execution status of the sub-queries in the distributed plan).

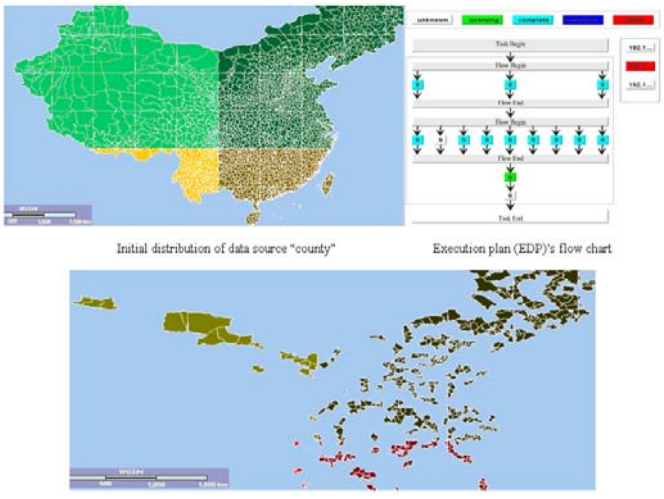

Figure 3. The sample query execution process demonstration (different colours represent from different data sites).

B. Efficiency comparison

Firstly, we compared the grid system with a central system on query processing efficiency with the same data, the results reveal that the grid system is much faster than the central system, especially when many parallel queries exist simultaneously or the global query involves geospatial join operation between different data sources.

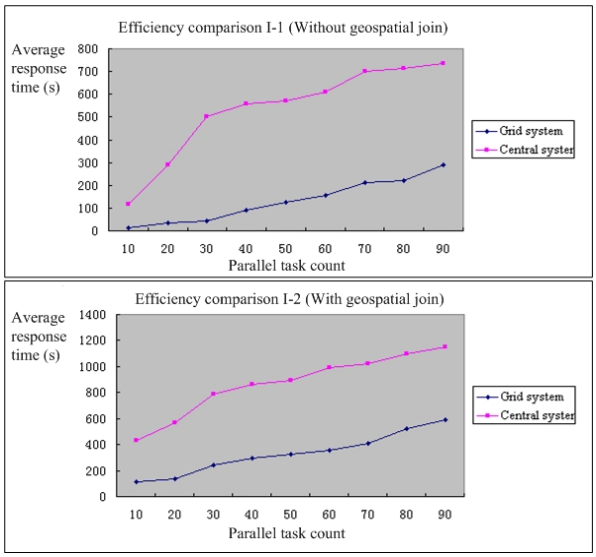

Figure 4. Efficiency comparison between grid system and central system.

Then we compared distributed query processing algorithm HHOA's efficiency with traditional algorithms MST and SSD-1 [15]. Extra implementation of the two algorithms was accomplished and through the execution time of generated plans appropriate judgments could be made (See Figure 5). HHOA is more effective than the traditional two, especially when the query involves the geospatial join operations between different data sources.

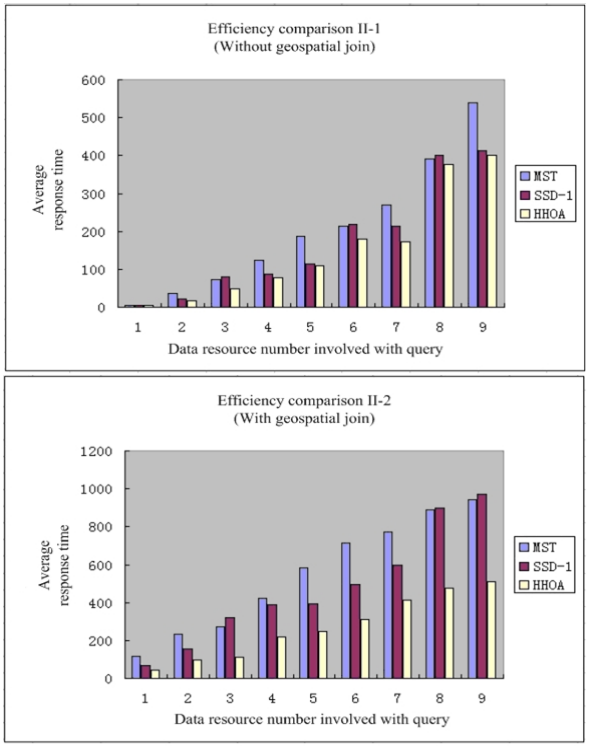

Figure 5. Efficiency comparison between different distributed query processing algorithms.

Moreover, the proposed grid catalogue was tested as well. We selected Globus MDS as an alternative, which could be used for comparison with the grid catalogue based on relation model. In practice, we had done an experiment to record the average response time of information retrieval to the grid catalogue. The response time under these two approaches was shown in Figure 6. Increasing with the sites number in the grid system, the average response time in the way of Globus MDS increases much more rapidly than our proposed approach. The experiment demonstrates that the approach based on relation model has much better efficiency.

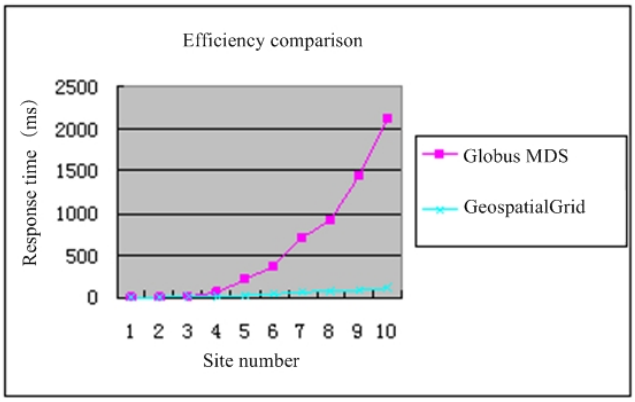

Figure 6. Efficiency comparison between Globus MDS and relation model based grid catalogue.

Running effects and comparison experiments indicate that the GGDS implementation approach that we provided in this paper offers an effective way for geospatial data processing and analysis. In particular, the new method for organizing the grid catalogue and HHOA algorithm both adapt the characteristics of geospatial query processing well.

V. CONCLUSION

In this paper, we discussed key GGDS technologies, including some new methods: relation model based grid catalogue, distributed query processing algorithm HHOA, etc. After accomplishing a GGDS prototype GEOBARN, several experiments had been done and the comparison effects indicate that our proposed methods work well for geospatial data processing and analyses in network. Thus, we observed that system availability and efficiency can be improved after applying grid computing to geospatial data processing and sharing.

ACKNOWLEDGMENT

This research was supported by the following grants from the National High Technology Development 863 Program of China (2007AA120502, 2007AA120503).

REFERENCES

[1] M. Erwig, M. Schneider and R.H. Guting, "Temporal Objects for Geospatial Data Models and a Comparison of Their Representations", Proceedings of the Workshops on Data Warehousing and Data Mining (1998)

[2] Y. Xue, A.P. Cracknell and H.D. Guo, "Telegeoprocessing: The integration of remote sensing, geographic information system (GIS), global positioning system (GPS) and telecommunication", International Journal of Remote Sensing. 23: 1851-1893(2002)

[3] Z. Shen, J. Luo and C. Zhou, "Architecture design of grid GIS and its applications on image processing based on LAN", Information Sciences. 166:1-17(2004)
[4] I. Foster, C. Kesselman, J.M. Nick and S. Tuecke "Grid services for distributed system integration", Computer. 6:37-46(2002)
[5] "The Globus Project Argonne National Laboratory USC Information Sciences Institute: Grid Architecture", Available from: http://www.globus.org
[6] N. Giannadakis, A. Rowe, M. Ghanem, Y. K. Guo, "InfoGrid: providing information integration for knowledge discovery", Information Sciences. 155:199-226(2003)
[7] C.C. Aggarwal, "A framework for diagnosing changes in evolving data streams", Proceedings of the 2003 ACM SIGMOD international conference on Management of data (2003)
[8] H. Cao, O. Wolfson and G. Trajcevski, "Geospatial data reduction with deterministic error bounds", Proceedings of the 2003 joint workshop on Foundations of mobile computing (2003)
[9] M. J. Egenhofer, "Spatial SQL, A query and presentation language", IEEE Transactions on Knowledge and Data Engineering. 6(1): 86-95(1994).
[10] S.K. Harms, J. Deogun, S. Goddard, "Building knowledge discovery into a geo-spatial decision support system", Proceedings of the 2003 ACM symposium on Applied computing (2003)
[11] C.R. Lin, K.H. Liu, M.S. Chen, "Dual Clustering: Integrating Data Clustering over Optimization and Constraint Domains", IEEE Transactions on Knowledge and Data Engineering. 17(5): 39-48(2005)
[12] K. Hornsby, "Workshop Report - International Workshop on Temporal, Spatial, and Geospatial Data Mining", Proceedings of the First International Workshop on Temporal, Spatial, and Geospatial Data Mining (2000)
[13] Z. Huang, Y. Fang, B. Chen, D. Yin and X. Peng, "A Global View Oriented Approach to Directory Management in Distributed Spatial Database", Proceedings of SPIE-The International Society for Optical Engineering, Vol.6418, GNSS and Integrated Geospatial Applications. 64181O(2006)
[14] L. Savary, T. Wan and K. Zeitouni, "Geospatial Data Warehouse Design for Human Activity Pattern Analysis", Proceedings of the Database and Expert Systems Applications, 15th International Workshop (2004)
[15] Z. Huang, B. Chen and Y. Fang, "An SQL Transformation Based Optimization Algorithm of Distributed Spatial Query", High Technology Letters. 17(10): 1013-1018(2007)
[16] Z. Huang, X. Peng and K. Zhang. "An Approach to Geographic SQL Global Parsing under Distributed Computation Circumstance", Geography and Geographic Information Science. 21(3): 18-21(2006)