THE KEY TECHNOLOGIC ISSUES OF PARALLEL SPATIAL DATABASE MAMAGEMENT SYSTEM FOR PARALLEL GIS

ZHAO Chun-yu[a], ZHAO Yuan-chun[b], MENG Ling-kui[a], DENG Shi-jun[a]

[a] The School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China -zcy7@263.net
[b] Chinese Academy of Surveying & Mapping, Hai-dian District, Beijing, China.

KEY WORDS: Parallel Computing, Parallel Spatial Database Management System, Parallel GIS, Decision Support Systems

ABSTRACT:

The technologies of organization and management of the massive volume spatial data are playing an important role in GIS. Especially, with the increasing of the volume of spatial data and the extending of the application of GIS, one of the requirements for the spatial database management systems (SDBMS) of the future is the ability to handle the huge volume spatial data. These applications involve spatial data mining and knowledge discovery, multi-dimensional and dynamic GIS, spatial-temporal GIS etc. Therefore, the paper presents a new software framework of spatial data management in GIS based on the parallel spatial database management system (PSDBMS) to solve the difficulties in storing and retrieval of the huge volume spatial data. The application of the parallel database management system in GIS will come forth lots of new issues. As we all know, the parallel GIS will provide a framework of high-performance computing for processing of the spatial data. In order to improving the integrated performance of parallel GIS, it is properly necessary to utilize the parallel database management system in the parallel GIS. The paper discusses the key technologic issues of the parallel spatial database management system, such as the software architecture of parallel GIS based on the PSDBMS, the technologies of the spatial data partitioning, and the related technologies of the optimal query in the PSDBMS etc. Due to the increasing requirements of GIS applications for high performance processing and storage of spatial data, we can foresee that the parallel GIS based on the PSDBMS should have the powerful ability to deal with the various complex applications in the future, and provide an effective software platform for processing, storage and retrieval of the spatial data.

1. INTRODUCTION

Because of the limitation of the software architecture and the functionalities, the traditional GIS cannot meet the application requirements gradually. With the increasing of the volume of spatial data and the extending of the application of GIS, the GIS users look forward to have a new GIS software platform to deal with the more complex problems in geo-spatial information science. Since the naissance of the first GIS platform – CGIS, the applications of GIS have not been only limited in the geographic field, more and more applications domain can benefit from the services and functionalities in GIS, these applications include data mining and knowledge discovery, multi-dimensional and dynamic GIS, spatial –temporal GIS etc. There is a common characteristic of these applications that the abilities for the high performance storage, retrieval and processing of spatial data should be provided by GIS.

The technologies of organization and management of the spatial data are one of the most key issues in GIS. As we all know, the development of spatial databases system will benefit from the progress of the commercial database management system, in other words, the advance of GIS technologies will depend on the improving of the technologies of database management system in a way. Some researchers started to study and develop the new software architecture of GIS under the parallel computing environment two decades ago, therefore, a new research field in GIS has emerged, the Parallel GIS. Richard Healey and Steve Dowers et al [1998] as the outstanding leaders have contributed to the development of the parallel GIS. With the rapidly developing of the technologies of computer hardware and software, people can obtain and build easily effectively the parallel computing environment, such as Beowulf Cluster etc. As such, it is feasible for the implementation and design of the parallel GIS based on the high value-cost ratio parallel computing environment.

The parallel GIS can provide a high performance processing of the spatial data to meet the requirements of computing intensive and I/O intensive applications, such as spatial data mining and knowledge discovery and decision support system etc. In fact, there are still many problems to need to be resolved in the parallel GIS. By analysing the existing research results of the parallel GIS, we can find that the organizing and management of the huge volume spatial data in parallel GIS still adopt the method of the file systems or parallel file systems provided by operating system. Obviously, the way of spatial data management by file systems have lots of limitations, such as the lower performance management of the huge volume spatial data, the data security and so on. Due to the reasons as above, the paper presents a new software framework of spatial data management in parallel GIS based on the parallel spatial database management system (PSDBMS).

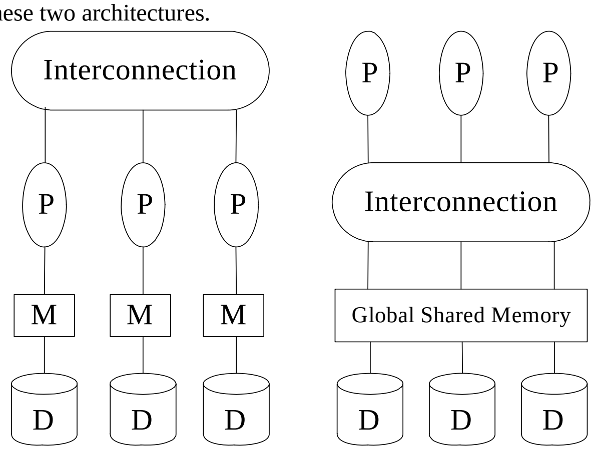
The development of database management systems has coincided with significant developments in distributed computing and processing technologies. The merging of these two resources has resulted in the emergence of parallel database management systems. Parallel database management systems are being used nowadays in a wide variety of systems, right from database applications to decision systems. Some commercial parallel database management systems will be obtained from the database vendors, such as Oracle 10g RAC etc. In the paper, we try our best to construct the parallel spatial database management systems based on the commercial parallel database systems and discuss the key technologic issues.

The paper is organized as follows. Section 2 proposes the architecture of the parallel spatial database management systems in parallel GIS. Section 3 describes the related key technologies of PDBMS, such as spatial data partitioning, query processing and optimization methods etc., and gives several basic techniques for these issues. Finally, section 4 lists the conclusions and highlights directions future work.

2. ARCHITECTURE OF PSDBMS IN PARALLEL GIS

2.1 Architecture of Parallel Database Systems

The parallel database management systems architecture range between two extremes: the shared-nothing and the shared memory architectures. In the shared nothing architecture, each processor has its own main memory and disk units. In shared memory architecture, every processor has access to any memory module or disk unit through a high-speed interconnection. Based on these two system architectures, other modified architectures are available for use. One of them is the shared disk architecture. In this architecture, processors have their own individual main memory space but they share common global disk storage. The shared disk could however become a bottleneck for limiting scalability, especially during large data retrieval. Fig. 1 illustrates these two architectures.



(a) Shared-Nothing (b) Shared-Memory

Figure 1. Two Architectures of Parallel Database Systems

As mentioned above, the processors in a shared-nothing architecture have their own memory and disks. Nodes communicate by passing messages to one another through an interconnection network. Shared-nothing architectures can incorporate hundreds of processors within a single system and this is one of its main advantages. However, its dependence on an interconnection network for message passing can be a cumbersome issue. Also, a high number of processors within a single system make it imperative that a good load-balancing strategy be in place. Also, if a processor fails then access to data owned by that processor is lost. A solution to this is a shared-disk architecture where every processor can read and write to any of the disks in the system but manages its own memory.

The shared memory architecture has zero communication cost: data and messages are exchanged through the shared memory area. Advantages of such architecture include easier process synchronization and minimal overhead for load balancing correction. But this architecture can be scaled effectively up to 40 processors. Beyond that interference affects the rate at which memory is processed.

The shared-nothing architecture is currently the more popular implementation, mainly due to its high reliability and ease of scalability to hundreds of processors. It is also the most cost effective means of construction a parallel database system as it can be built from existing sequential machines using a simple interconnection network.

2.2 Architecture of PSDBMS in Parallel GIS

In general, parallel GIS software is developed based on the two architectures mentioned above. As such, parallel GIS can also obtain high reliability and ease of scalability from shared-nothing architecture. To using the new hardware platform makes the storage, processing and management of a huge volume spatial data more intricate. In traditional GIS, the massive volume spatial data would be managed and organized by file systems provided by operating system or by commercial database management systems, and logically stored on a single disk or on big volume size storage equipment, for example disk array system and SAN (Storage Area Network) etc. All functions and services provided by traditional GIS are being serially run on the single processor system. Obviously, the implementation of all algorithms in traditional GIS is effective and efficient rather than in the parallel GIS.

There are lots of functions in parallel GIS must to be programmed in parallel computing, and run effectively on the parallel computer. A parallel computer, or multiprocessor, is itself a distributed system made of a number of nodes (processors and memories) connected by a fast network within a cabinet. The advantages of parallel computing in parallel GIS based on the parallel computing environment are that GIS platform can get a high throughput for spatial data and an efficient and effective running performance. But the cost of the software development, system maintenance and operating of parallel GIS is further higher than that of traditional GIS.

The organization and management of huge volume spatial data is a key technologic issue not only for traditional GIS but also for parallel GIS. At present, commercial database management systems, especially emergence of the object-relational database management systems, have had the ability to manage and store the spatial data effectively supported by spatial data management module appended in DBMS, and provided a high performance spatial data service for traditional GIS. The construction of spatial database systems based on commercial database management systems will improve the efficiency of spatial data management and strengthen the security of spatial data in traditional GIS, it makes the traditional GIS more reliable and scalable.

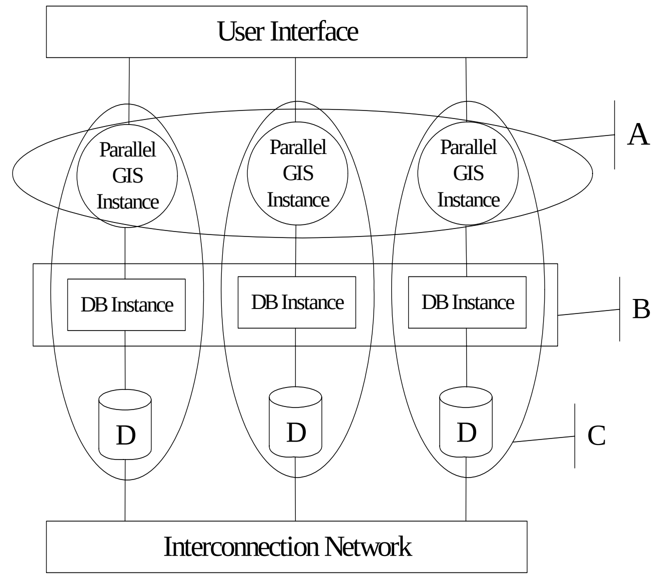
Early, the popular file systems and parallel file systems have been used to manage and organize the spatial data in parallel GIS. Under the temporal level of hardware and software in computer science, the method of spatial data management in file systems has provided a high performance I/O processing and a flexible and simple way of spatial data organization. However, the shortcoming in data management brought by file systems has reduced the performance of the integrated system, and is not fit for a huge volume spatial data management.

The maturation of database management system technology has coincided with significant developments in distributed computing and parallel processing technologies. The end result is the emergence of distributed database management systems and parallel database management systems. These systems have started to become the dominant data management tools for highly data-intensive applications. In the field of spatial information science, most of the functions and services provided by spatial information systems, commonly geographic information system, belong to the data-intensive and I/O intensive applications. With the increasing of the volume of spatial data, GIS users need a new GIS that have the more high performance processing and effective data management. Fortunately, the emergence of parallel GIS and parallel database management systems benefits the progress of spatial information technology.

Few of commercial parallel database management systems provide the ability of spatial data management on the parallel computing environment, such as shared-nothing architecture. Therefore, some technologic issues must to be considered when we built the parallel spatial database management systems, for example, spatial data partitioning on the multiprocessors or multi-nodes computers, spatial indexing under the parallel computing environment, and query optimization in the parallel spatial database management system etc.

The Paradise project [1998] is the first organization that works on the parallel spatial database system for GIS application on the world. The objective of the Paradise project is to design, implement, and evaluate a scalable, parallel geographic information system that is capable of storing and manipulating massive data sets. By applying object-oriented and parallel database technologies to the problem of storing and manipulating geographic information to significantly advance the size and complexity of GIS data sets that can be successfully stored, browsed, and queried. Whereas, it is a pity that the Paradise project cannot continue to work on this research direction. As we can learn about from the Paradise project, the research direction is the sub-project under the Database Research Group, UW Madison, its truly objective of the project is to test and validate the system validity of the STORE object-relational database management system. In other words, the Paradise project cannot release the potential integrated system performance by integrating both GIS and parallel spatial database management system.

Through integrating parallel GIS and parallel spatial database management systems, we can establish a new GIS software architecture that is capable of high performance data processing and massive spatial data management. The paper gives the architecture of parallel GIS supported by parallel spatial database management systems for the first time in the field of high performance GIS, and discusses the related technologic issues. The architecture given by the paper is shown as fig. 2.



A: The Logical Function Layer of Parallel GIS
B: The Logical Function Layer of Parallel Spatial DBMS
C: Computing Node of Parallel Computing Environment

Figure 2. The Architecture of Parallel GIS based on Parallel Spatial Database Management System

As the figure 2 illustrated, parallel GIS based on parallel spatial database management system is established on the shared-nothing architecture, however, each of parallel GIS instance and parallel spatial DBMS instance will be deployed on the computing node. Figure 2 describes the relationship of the whole software and hardware architecture. As we can see, from upper (User Interface) to bottom (Interconnection Network) of figure 2, section A illustrates the logical function layer of parallel GIS in which all functions will be running and implementation, and section B is the logical function layer of parallel spatial DBMS in which every DB instance will be deployed and manipulates the spatial data within a single logical DBMS, that is, parallel spatial DBMS shows itself a single DBMS to users. Section C presents a computing node of parallel computing environment that includes the parallel GIS instance and DB instance.

3. THE KEY TECHNOLOGIC ISSUES OF PSDBMS

Spatial information has the characteristics of massive volume size, complex structure, and variety of entity etc. It is different with common digital data object, such as text, image, video/audio and so on. Therefore, it is inevitable to adopt different strategy for organizing and managing the huge volume spatial data.

In practice, it is an inevitable trend that a huge volume spatial data would be stored and manipulated by the commercial DBMS. Furthermore, lots of main commercial DBMS vendors provide the appropriative functional module to support spatial data in their DBMS product, such as, Oracle Spatial, IBM DB2, and ESRI ArcSDE etc.

As mentioned above, parallel DBMS is different with common DBMS on the architecture. There are many technologic issues need to be considered in parallel DBMS, for example, data placement and query optimization under the parallel computing environment. Similarly, as the particularity of spatial data, these issues listed above will bring lots of new problems when we establish the parallel spatial DBMS based on commercial parallel DBMS. This section of the paper will discuss several key technologic issues of construction of parallel spatial DBMS.

3.1 Spatial Data Partitioning Techniques

In parallel DBMS, the database can be physically distributed across data sites by fragmenting and replicating the data. Given a relational database schema, fragmentation subdivides each relation into horizontal (by a selection operation) or vertical (by a projection operation) partitions. Fragmentation is desirable because it reduces the size of relations involved in user queries. Based on the user access patterns, each of the fragments may also be replicated.

Data partitioning (or called data placement) strategies act as a significant aid in improving the performance of a parallel database system. Partitioning and distributing the data allows for the use of improved data processing techniques so as to allow each data section to be processed individually. Similarly data replication in a parallel database system can act as a backup strategy as well as provide an opportunity for carrying out separate processing on each individual data set.

These described strategies are static methods i.e. they are carried out at periodic intervals when data is not being processed. Likewise, dynamic data partitioning techniques are available which rearrange the data when data processing algorithms are also executing on the data. This is turn takes care of data access skew by carrying out periodic data redistribution. Such a dynamic data allocation method identifies data fragments that are accessed the most (so called hot nodes) and moves data from these nodes onto other segments so that there is even data access throughout the system, thereby reducing the chances of system failure through processor overburdening.

The effect of these data partitioning techniques, either static or dynamic, is positive as throughput is improved with re-allocation. Both re-allocation types show a significant improvement in database processing if they are supported with multiprogramming techniques. At lower levels of multiprogramming, the advantage of dynamic re-allocation is not substantial due to the fact that the resources outweigh the demands for information from each individual node. At a higher level of multiprogramming re-allocation is not very effective as the gain in performance is offset by excessive data transfer and transaction restart. Thus the conclusion is that re-allocation is observed to improve throughput but not always significantly as might be naturally assumed.

A significant aim of parallel DBMS is to provide I/O parallelism so as to get a high performance parallel data processing. By partitioning the relation, we can use the simplest method to obtain I/O parallelism. In parallel DBMS, there are several techniques is to be used for data partitioning, as follows. Firstly, we assume that data will be partitioned onto *n* disks, such as D_0, D_1, … , D_{n-1}.

- Round - robin: This strategy is to scan the relation in random sequence, and deliver tuple *i* onto the $D_{i \bmod n}$ disk. Round-robin strategy can ensure the approximate equal numbers tuple on every disk.
- Hash partitioning: In the fragmenting policy, there are one or more attributes in a given relation schema is to be specified act as partitioning attribute. By selecting a hash function, its value domain is {0, 1, … , n-1}, each of tuple in the row relation will be mapping into specified value domain by hash function based on the partitioning attribute, and if the return result is *i*, corresponding tuple will be sent to D_i disk.
- Range partitioning: The strategy is to distribute a segment that its value is of sequential onto different disk. Firstly, selecting a partitioning attribute A and forming partitioning vector. Set $\{v_0, v_1, \dots, v_{n-2}\}$ as the partitioning vector that meet the situation, that is, when $i < j$, and $v_i < v_j$. Considering tuple t, t[A] = *x*. if $x < v_0$, then put tuple t onto the D_0 disk. If $x >= v_{n-2}$, then tuple t will be put onto the D_{n-1} disk. Otherwise, if $v_i =< x < v_{i+1}$, then tuple t will be distributed onto the D_{i+1} disk.

For example, there are three disks that specified respectively by 0, 1, and 2. By using range partitioning, we may distribute all the tuples that these value is less than 5 onto disk 0, and put all the tuples that these value is between 5 with 40 onto disk 1, then distribute all the tuples that these value is more than 40 onto disk 2.

All these three data partitioning methods mentioned above have being used in commercial parallel DBMS widely. Once one relation is been distributed onto multi-disks, we can retrieval our interest information through querying from all over the disks. To the different query, each of these data partitioning techniques has different performance.

Spatial data has special characteristics rather than common digital object. The length of spatial data record is variety, that is, the volume size of spatial data is not fixed. For example, a spatial point object can be described using two double values in specified spatial reference system, X and Y. That is, the shortest length of a point object is 16 bytes as a kind of length-fixed object. Line object is composed of multi points, and the number of points is not fixed, therefore, the length of a line object is length-variant. Similarly polygon object is same with line entity.

The parallel spatial DBMS presented in the paper will adopt the object-relational DBMS, spatial data and its corresponded attribute data will be organized within a relation. Lots of advantages brought by the method have had discussed in many other related documents. The structure of the relation is illustrated as figure 3.

Object ID	Name	DataType	– – – – –	Geometry
001	P1	Point	– – – – –	Geometry
002	P2	Point	– – – – –	Geometry
⋮	⋮	⋮	⋮	⋮
1001	L1	Line	– – – – –	Geometry
1002	L2	Line	– – – – –	Geometry
⋮	⋮	⋮	⋮	⋮

Figure 3. The Structure of Relation Based On Object-relational DBMS

Figure 3 describes the structure of relation based on object-relational DBMS. As we can see, *Object ID*, *Name* etc., all these attribute field are saved with its spatial object, *Geometry*, in one relation table.

To the commercial database systems, in general, data stored in the parallel DBMS is common data type, such as Integer, Text, String and so on, therefore, the length of each of tuples in the relation is fixed. In other word, these three data partitioning techniques mentioned above can appropriately solve the data skew during partitioning data among the different disks. Whereas, the length of each of tuples in the relation table of spatial DBMS is not fixed, the existing data partitioning techniques will not meet the requirement for the spatial data fragment among all the disks.

In parallel DBMS, generally, selecting a field of the relation as a partitioning attribute, we can make use of one of three partitioning techniques to subdivide the relation onto different disks. However, to the parallel spatial DBMS, using the traditional data partitioning techniques will bring serious data skew so as to reduce processing performance and I/O parallelism. Of course, we can use the common data partitioning techniques to fragment the relation if only *point* objects are saved in the relation table. Due to the status, some new spatial data partitioning techniques need to be utilized to resolve the spatial data skew.

The paper presents a simple method to solve the data skew. The method is described as follows: when constructing the relation, appending a new field domain in the relation, such as, the name of the new field domain is *DATASIZE*. When loading the spatial data into the parallel DBMS, the new field domain will save the size of the corresponded spatial data object. For example, the field domain can store the number of points that belong to the spatial object. As the relation includes the field domain, traditional data partitioning techniques can chose the value of the field domain act as the partitioning attribute so as to fragment the relation. Figure 4 describes the simple strategy for spatial data partitioning techniques.

Object ID	Name	DataType	*DataSize*	Geometry
001	P1	Point	2	Geometry
002	P2	Point	2	Geometry
⋮	⋮	⋮	⋮	⋮
1001	L1	Line	5	Geometry
1002	L2	Line	8	Geometry
⋮	⋮	⋮	⋮	⋮

Figure 4. The Structure of Relation Based On Object-relational DBMS including the spatial data partitioning attribute --- DataSize

As illustrated in Figure 4, the volume of *DATASIZE* is a common field domain act as a partitioning attribute. Based on the partitioning attribute, using hash partitioning or range partitioning to fragment the relation can get a better partitioning result, that is, databases can be distributed among all the disks.

Nevertheless, there are several issues need to be discussed. In fact, the *DATASIZE* volume is similar with other common field domain. The data skew issues still exist if the relation is been partitioned based on the *DATASIZE*. Practically, a new field domain created in relation, *DATASIZE*, is to be used to save the number of points of corresponded spatial object. The value of the field domain can be used to describe the size of every spatial object. For example, a polygon is composed of 10 points, and then its size is about 10*2*8 bytes.

During partitioning spatial data relation based on *DATASIZE*, if we do not take the size of every spatial object into account, then a direct result is to bring the serious data skew status. Especially, the data skew issue will be more serious when using range-partitioning technique. It is considerable that a new high performance technique must be used to solve the spatial data skew issue in parallel spatial DBMS. In the paper, we give another method to avoid spatial data skew. In the new method, a histogram of spatial data frequency will be used to indicate the frequency of each spatial data object, and record the size of each one, and then average the histogram to subdivide the spatial data relation into a balanced status.

3.2 Query Optimization in PSDBMS

In parallel database systems, query processing and optimization techniques have to address difficulties arising from the fragmentation and distribution of data. To deal with fragmentation, data localization techniques are used where an algebraic query, which is specified on global relations, is transformed into one that operates on fragments rather than global relations. In the process, opportunities for parallel execution are identified and unnecessary work is eliminated. Localization requires the optimization of global operations, which is undertaken as part of global query optimization. This in turn involves permuting the order of operations in a query, determining the execution sites for various distributed operations and identifying the best distributed executing algorithm for distributed operations.

Query execution techniques can be classified into two forms of parallelisms:

- Intra-operator parallelism - One operation is parallelized over several processors. This can be achieved by partitioning the data among the processors. Each processor then carries out the query on its own data set and the results are then combined for the eventual result.
- Inter-operator parallelism – Here, several processors are executes simultaneously, each processor carrying out a process. There are two forms of inter-operator parallelisms: independent and pipelined.

Various implementations exist for achieving intra-operator parallelism. An approach known as de-clustering can be used to partition the query into fragments that can then be allocated to the processors. This approach is most useful when optimizing complex queries. Various sections of the queries can be broken and each part can be allocated to a separate processor. The allocation of the various query subparts can also be done such that those sharing the same data set can be allocated to adjacent or nearby processors. This helps in reducing the costs associated with moving the query result from one stage to another.

It is clear that query processing and optimization depends on the data partitioning and the distribution of data. In parallel databases systems, lots of approaches for query processing and optimization have been developed that can achieve well query operation parallelism. However, in parallel spatial database systems, due to the different data partitioning techniques and the complex characteristics of spatial data spatial query and retrieval will face many new difficulties in query parallelism and parallel execution of processing.

In parallel spatial DBMS for parallel GIS, spatial query is the most important function involving range query, spatial join query, nearest neighbor query etc. The using frequency of range query is the highest in GIS. In parallel DBMS, several techniques have being used for parallel query, for example, parallel join based partitioned, fragment-and-replicate join, parallel hash join based on partitioned and so on. Currently, there is little research result for parallel spatial query, but lots of techniques in parallel DBMS can be applied into the parallel spatial DBMS.

4. CONCLUSIONS

The paper discusses the related key technologic issues in the parallel spatial DBMS for parallel GIS. Considering the foregoing sections, it is clear that for a parallel database system to perform at an optimum level, various operational and non-operational aspects need to be given careful consideration. In this regard, architectural design, data partitioning and query optimization can be considered the main factors. Besides these, other factors also exist which have due weight-age in determining the performance of the parallel database system. For the complex characteristics of spatial data, many new issues should be considered in parallel spatial DBMS for parallel GIS. Of course, these existing techniques in parallel DBMS can be used to develop the new approaches for the parallel spatial DBMS. In the future, our research will focus on the parallel spatial query, parallel query optimization, and architectural design of parallel GIS supported by parallel spatial DBMS.

BIBLIOGRAPHY

M. Tamer Ozsu, Patrick Valduriez, 1996. Distributed and Parallel Dataabase Systems. *ACM Computing Surveys.* pp. 1-21.

Abdelsalam Helal, David Yuan, Hesham El-Rewini, 1997. Dynamic Data Reallocation for Skew Management in Shared-Nothing Parallel Databases. *Distributed and Parallel Databases*, 5(3).

Mansih Methta, David J. Dewitt, 1997. Data Placement in Shared-Nothing Parallel Database Systems. *The VLDB Journal*.

S. Dowers, B.M. Gittings, M.J. Mineter. 2000. Towards a framework for high-performance geocomputation: handling vector-topology within a distributed service environment. *Computers, Environment and Urban Systems,* 24, pp.471-486.

Charles Bookman,2002. *Linux Clustering Building and Maintaining Linux Clusters*. USA: New Riders Publishing.

Jang Ho Park, Kihong Kim, Sang K.Cha. 1999. A High-Performance Spatial Storage System Based on Main-Memory Database Architecture. *DEXA'99*, pp.1066-1075.

Terence M. Sloan, Michael J. Mineter, Steve Dowers, Connor Mulholland, Gordon Darling, and Bruce M. Gittings, 1999. Partitioning of Vector-Topological Data for Parallel GIS Operations: Assessment and Performance Analysis. *Euro-Par'99, LNCS 1685*,pp.691-694.

Brian Tierney, William Johnston, et al. 2000. A Data Intensive Distributed computing architecture for Grid application. *Futrue Generation Computer System.* 16, pp. 473-481.

Michael J. Mineter, Steve Dowers, Bruce M. Gittings, 2000. Towards a HPC Framework for Integrated Processing of Geographical Data: Encapsulating the Complexity of Parallel Algorithms. *Transaction in GIS*, 4(3), pp. 245~262.

Michael J. Mineter. 2003. A software framework to create vector-topology in parallel GIS operations. *International Journal of Geographical Information Science*, 17(3), pp. 203~222.