J. Basic. Appl. Sci. Res., 4(1)21-26, 2014
© 2014, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

PFA: Parallel Filtration Algorithm for Query Technology in Spatial Database

Abbas Karimi[1,2*], Faraneh Zarafshan[2], and S.A.R. Al-Haddad[2], V. Saeidi[1], S. Morshed[1]

[1]Department of computer Engineering, Faculty of Engineering, Arak Branch, Islamic Azad University, Arak, Iran.
[2]Department of computer and communication systems Engineering, Faculty of Engineering, UPM, Serdang, Malaysia

Received: July 13 2013
Accepted: August 11 2013

ABSTRACT

Spatial database is used for storing large volume of complex data and information analysis. Query technology is one of the complex and time-consuming operations in spatial database. In this paper, a novel parallel algorithm for filtration stage of query technology has been proposed. The parallel filtration algorithm has been presented on EREW shared-memory systems. The analysis shows that the algorithm has improved speed up and efficiency. It especially has decreased time complexity from $O(n^2)$ to $O(n)$ which makes it appropriate for applications where the volume of input data is high.

KEYWORDS: Spatial Database, Query Technology, Filter-refining, Parallel Algorithm.

1 INTRODUCTION

Spatial database as a database system stores and manages large volume of data, and supports different types of data, index, query and database. Spatial database differs to other database for two reasons including its ability to store complicated data and using spatial operator to process them(Shubin, Jizhong et al. 2009). Spatial database have some basic features. Their data volume is high, reference to these data is more than the other databases, spatial data management and their attributes are complex and they are utilized in widespread. Geographic Information System (GIS), decision management systems, information analysis systems, health care applications and business intelligence applications are instances of applications using Spatial database (Yeung and Hall 2007).

Query is an important operation that searches for situation or coordination of spatial objects with special attributes in database. Total execution time of applications are normally utilizes for evaluating the performance of a query operation, however the response time to a query increases with dynamic growing of data and it is the main problem associated with query technology. One solution for decreasing the time complexity and improving the performance of spatial database is further improvements in designation and implementation level of systems (Lijing and Xuanhui 2010). This research aims to decrease the time complexity of query technology by improving its designation. To do so, parallelism is utilized in filtration stage to decrease the time complexity as an important factor for increasing special database performance.

The remainder of the paper is organized as follows. Section 2 introduces related works. The sequential and parallel filtration algorithms based on geometric intersection are presented in Section 3. The performance analysis of new parallel filtration algorithm is compared to sequential algorithm in Section 4. Finally, conclusions and future works are presented in Section 5.

2 RELATED WORKS

(Jun, Mamoulis et al. 2004; Zhang, Papadias et al. 2005; Shubin, Jizhong et al. 2009) classified Spatial Data Query to three kinds including spatial selection query (Zhang, Jagadish et al. 2010), nearest neighbor query (Gao, Zheng et al. 2009), and spatial join query (Shubin, Jizhong et al. 2009).

Spatial selection queries (Zhang, Jagadish et al. 2010) provide basic services for spatial operation. Therefore, effective spatial selection queries yield to efficiency of total spatial data management system. Point query and region query are two basic kinds of spatial selection query . Point query finds geometrical objects of M that contains P, while region query finds objects of M which have geometrical intersection. Region query is usually rectangle-like. The equation of Point query and Region query are brought in Equations (1) and (2), respectively.

$$Q(P) = \{O \mid Q\, contains\, (P), O \in M\} \quad (1)$$

$$Q(P) = \{O \mid Q, G \cap R, G \neq \phi, O \in M\} \quad (2)$$

[*]**Corresponding Author:** Abbas Karimi, Department of computer Engineering, Faculty of Engineering, Arak Branch, Islamic Azad University, Arak, Iran. Email: Akarimi@iau-arak.ac.ir

where *P, O, M,* and *R* are respectively a query point, spatial object, a set of objects, and a region query.

Nearest Neighbor Query (Gao, Zheng et al. 2009) includes two basic types. They are K Nearest Neighbor (KNN) query and All Nearest Neighbor (ANN) query. The purpose of KNN is to find K objects in dataset M that are limited by query point q. Present algorithm suppose that dataset is indexed by R-tree and recession search space with different metrics (Manolopoulos, Nanopoulos et al. 2005). Nearest neighbor query is a special case of KNN where K=1, q is query point, and o and o' are spatial objects.

$$1NN(q) = \{o | \forall o' : dist(q,o) \leq dist(q,o'),\ o \in M, o' \in M\}. \quad (3)$$

ANN query finds each object of *A* that is the nearest neighbor to *B* as presented in Equation (4). Given that *A* and *B* are two collections of spatial data, and *dist(a,b)* is a distance metric. Note that *ANN(A,B)* ≠ *ANN(B,A).*

$$ANN(A,B) = \left\{ < a_i, b_j > \ | \forall a_i \in A : \exists b_j \in B, \neg \exists b_k \in \&A\{dist(a_i, b_k) < dist(a_i, b_j)\}\right\}. \quad (4)$$

Spatial join query *(Shubin, Jizhong et al. 2009)* is used for implementing map shield Spatial join operation, as presented in Equation (5). It combines the objects in spatial data source based on their geometrical attributes. These geometrical attributes are the basis of some spatial predictions on the existence of intersection, consisting or distance range. Spatial join queries are usually intersection joins.

$$SJ(R,S) = \{(r,s) | r.join(s), r \in R, s \in S\}. \quad (5)$$

There are three types of Spatial join query including Topological intersection query, Sequence intersection query, and Measurement intersection query each of which has two main steps of filtering and refining. In this research, the concentration is on filtration in spatial join query.

2.1 Filtration

Objects are normally irregular shapes in spatial database but are considered as similar regular shapes for the purpose of Filtration. As shown in Fig. 1, the objects are allocated in a rectangle with minimal area, and the objects outside the area will be removed.

As seen in Fig.1 (top), objects *A, B, C* are located in query area and object *D* is out of query area. So, *D* will be removed. (Fig.1 (middle)).Only four operations are required to find the intersection between two rectangles.

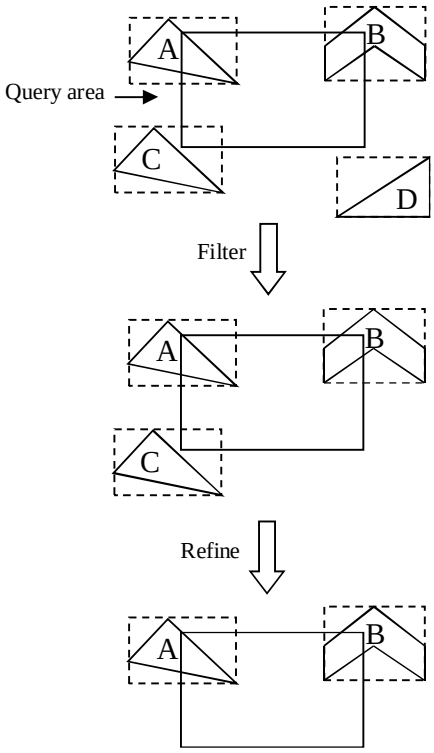

Fig. 1: filtration and refinement of Spatial Object Query.

2.2 Refinement

Refinement processes the outcome of filtration by taken into account the exact regular geometric shapes of objects. As shown in Fig. 1 (middle), the irregular objects *A* and *B* are located in query area, while irregular

object *C* is located out. Refinement operation removes object *C* as illustrated in Fig.1(bottom). Although, the number of objects concluded from this step is severely reduced compared with filtration step, the amount of required processing is high (Limited 2010).

Two main sequential filtration algorithms are presented in literature for filtration by utilizing geometric interactions. Map Scan technique presented by (Zhang and Zuo 2009) processes the intersection between objects in sets. time complexity of this algorithm is $O(n^2)$. = (Lijing and Xuanhui 2010) used divide-and-conquer method to decrease the time complexity of $O(n^2)$ to $O(nlogn)$. In this paper, an optimized parallel filtration algorithm based on EREW Shared-memory systems is proposed with time complexity of $O(n)$.

3 METHODOLOGY

The sequential algorithm presented by (Zhang and Zuo 2009) is the basis of new parallel algorithm. Prior to introduce new parallel algorithm, it is required to briefly overview the sequential filtration algorithm.

3.1 Sequential Filtration Algorithm

There are two sets of objects labeled with $A = \{A_1, A_2, \dots, A_a\}$, $B = \{B_1, B_2, \dots, B_b\}$. Each object is shown as a rectangle. The objects are sorted on the left down corner of coordination regarding to their value on *x* axis. All sorted objects are later located in set *AUB*. Then, the following steps are applied.

1) Scanline, as illustrated in Fig.2, sweeps from left to right along with *y* axis. It stops once an object in set *AUB* is visited.
2) If visited object is a member of set *A*, then scanline will scan all rectangles of set *AUB* until it reaches to an object of set *B*.
3) If the visited object of set *B* intersects the object already visited by scanline from set A, both objects will be recorded. Otherwise, after checking all objects of set *B* the object visited by scanline will be removed.
4) Scanline sweeping continues until next rectangle is visited. Then, algorithm returns to Step 2.

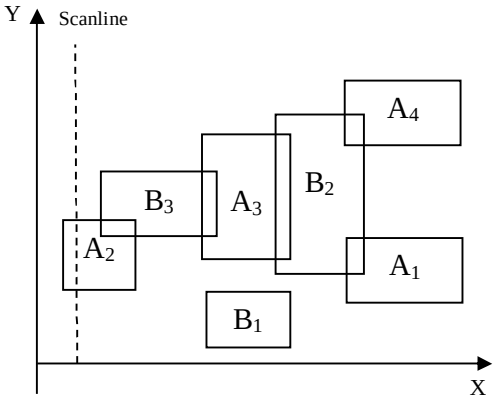

Fig. 2 Method of scanline.

3.2 Parallel Filtration Algorithm Based on Geometric Intersection

There are two architectures for multi-processor systems. One is shared-memory and the other is message passing. In a shared-memory parallel system, *n* processors are assumed to share public working space or to have a common public memory (Karimi, Zarafshan et al. 2012).

new proposed parallel algorithm uses geometric intersection on EREW shared-memory system. Assumptions are as follows.

1) There are two sets $A = \{A_1, A_2, \dots, A_a\}$, $B = \{B_1, B_2, \dots, B_b\}$, number of objects in set *A* is *a* and number of objects in set *B* is *b*.
2) *P* is the number of processors and *P=a*.
3) Each object is considered as a rectangle with four points as shown in Fig. 3.
4) Intersection between two objects is there are in common area, even if it is one point only.

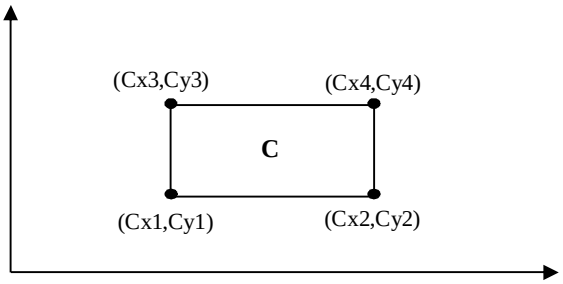

Fig. 3 C rectangle's description.

Pseudo-code of new parallel filtration algorithm is presented in Fig. 4.

Algorithm: filtration stage of Parallel algorithm based on geometric intersection	
Input: $A=\{A_1, A_2, \ldots, A_a\}$ and $B=\{B_1, B_2, \ldots, B_b\}$ as sets of spatial objects.	
Output: The pairs of objects from different sets that have intersection	
1.	**Begin**
2.	**For $i=1$ to b Do**
3.	**For all P_j, where $1 \le j \le a$ Do in parallel**
4.	**IF (($A_j x1 \ge B_i x1$ AND $A_j y1 \ge B_i y1$ AND $A_j x1 \le B_i x4$ AND $A_j y1 \le B_i y4$)**
5.	**OR ($A_j x2 \ge B_i x1$ AND $A_j y2 \ge B_i y1$ AND $A_j x2 \le B_i x4$ AND $A_j y2 \le B_i y4$)**
6.	**OR ($A_j x3 \ge B_i x1$ AND $A_j y3 \ge B_i y1$ AND $A_j x3 \le B_i x4$ AND $A_j y3 \le B_i y4$)**
7.	**OR ($A_j x4 \ge B_i x1$ AND $A_j y4 \ge B_i y1$ AND $A_j x4 \le B_i x4$ AND $A_j y4 \le B_i y4$)**
8.	**OR ($B_i x1 \ge A_j x1$ AND $B_i y1 \ge A_j y1$ AND $B_i x1 \le A_j x4$ AND $B_i y1 \le A_j y4$)**
9.	**OR ($B_i x2 \ge A_j x1$ AND $B_i y2 \ge A_j y1$ AND $B_i x2 \le A_j x4$ AND $B_i y2 \le A_j y4$)**
10.	**OR ($B_i x3 \ge A_j x1$ AND $B_i y3 \ge A_j y1$ AND $B_i x3 \le A_j x4$ AND $B_i y3 \le A_j y4$)**
11.	**OR ($B_i x4 \ge A_j x1$ AND $B_i y4 \ge A_j y1$ AND $B_i x4 \le A_j x4$ AND $B_i y4 \le A_j y4$)) then**
12.	**Write ($<A_j, B_i>$)**
13.	**EndIF**
14.	**EndFor**
15.	**EndFor**
16.	**End**

Fig. 4: Improved parallel algorithm.

Each iteration of first loop (Line 2) selects one object from set *B*. At second loop (Line 3), a processor is assigned to each object of set *A*. In other word, P_1 to P_a processors are assigned to objects of set *A* and work in parallel. Then, selected object from set *B* is compared in parallel to each object of set *A*. In step i ; $1 \le j \le b$; each processor P_j (allocated to object A_j) compares object A_j to object B_i. If A_j and B_i intersected, then the pair $< A_j, B_i >$ is generated as the output (Fig. 4).

4 RESULTS AND DISCUSSION

To describe the time complexity of new algorithm, $T_s(n)$ is defined as function of execution time of sequential filtration algorithm and $T_p(n)$ as function of the execution time of parallel filtration algorithm.

The time complexity of sequential algorithm equals to $T_s(n)=O(ab)$ *(Lijing and Xuanhui 2010)*, while parallel algorithm needs time of $O(b)$ for first loop and constant time of $O(1)$ for second loop. In other words, loop executes $b+1$ times. procedure of finding intersection between objects in Fig.4 (Line 4) takes time of $O(b+1)$. Execution time in Fig.4 (Line 12) is $O(b)$. Thus total execution time of operation is:

$$T_p(b) = 3b + 3 \quad (6)$$

In other words, total time complexity of new algorithm is $T_p(n)=O(b)$, while $O(ab)$ for sequential algorithm. Note that $n=a+b$.

Speed-up and efficiency are two factors to compare performance of parallel and sequential algorithms (Rajaraman and Murthy 2004). Speed-up shows how fast the program is running on multi-processor system in comparison with single processor system (Karimi, Zarafshan et al. 2011). Efficiency denotes the average time to keep each processor busy while running a parallel algorithm (Rajaraman and Murthy 2004). The equations of speed-up and efficiency of parallel filtration algorithm are presented in Equations (7) and (8).

$$Speed\ up = \frac{T}{T_p} = \frac{ab}{b} = a \quad (7)$$

$$Efficency = \frac{Speed\ up}{P} = \frac{a}{a} = 1 \quad (8)$$

Since the efficiency is 1, new parallel algorithm is optimal.

Assume the number of objects in both sets is equal and is considered as b. In Fig. 5 time complexity of both algorithms are compared where $0 \le b \le 10$.

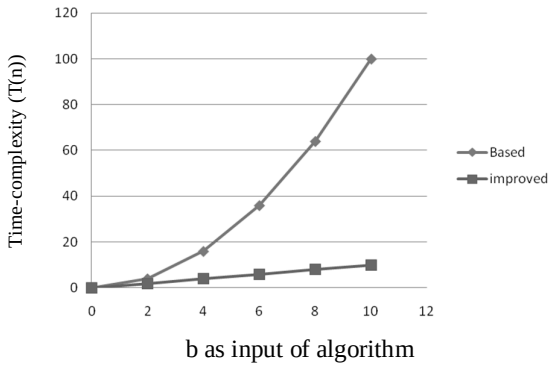

Fig. 5 Comparison of two algorithms for low amount of b.

The number of operations of both algorithm are shown in Fig. 6 when *0≤b≤1000*.

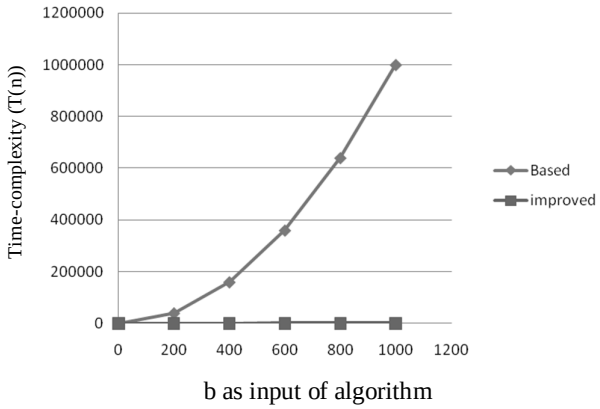

Fig. 6 Comparison of two algorithms for low amount of b.

As shown in Fig. 5 and Fig. 6, increasing number of objects in parallel algorithm yields to significantly deduce in cost of operations while it is opposite in sequential algorithm.

5 CONCLUSION

Spatial query operation can process databases in various size. The volume of spatial data is normally high and data structures are abstruse. As a result the cost of query in spatial database is high and optimization of spatial query becomes more important. In this paper, a parallel algorithm for finding intersections between objects of sets in filtration step of query technology was proposed. The parallel system was shared-memory EREW model. As seen in the results and discussion, the time complexity of sequential algorithm was $O(n^2)$, whereas parallel filtration algorithm has improved time complexity to $O(n)$. Furthermore, speed-up and efficiency as major performance evaluation factors of parallel algorithm are high and the algorithm is optimal.

Design of an optimal algorithm in refining step of query technology is one of the triggered issues for research in future.

Acknowledgment

The authors declare that they have no conflicts of interest in this research.

REFERENCES

Gao, Y., B. Zheng, G. Chen and Q. Li (2009). "On efficient mutual nearest neighbor query processing in spatial databases." Data & Knowledge Engineering 68(8): 705-727.

Jun, Z., N. Mamoulis, D. Papadias and T. Yufei (2004). All-nearest-neighbors queries in spatial databases. 16th International Conference on Scientific and Statistical Database Management, 2004. Proceedings. .

Karimi, A., F. Zarafshan, A. b. Jantan and S. A. R. Al-Haddad (2011). "A New parallel N-input Voting for Large Scale Fault-tolerant Control Systems " Journal of Electronic Science and Technology(JEST) 9(2): 1-6.

Karimi, A., F. Zarafshan, A. Jantan, A. R. Ramli, M. I. B. Saripan and S. A. R. Al-Haddad (2012). "Exact parallel plurality voting algorithm for totally ordered object space fault-tolerant systems." Pertanika Journal of Science and Technology 20(1): 89-96.

Lijing, Z. and H. Xuanhui (2010). Research on query technology in spatial database. 2010 Second Pacific-Asia Conference on Circuits,Communications and System (PACCS).

Limited, I. E. S. (2010). Introduction To Database Systems, Pearson Education.

Manolopoulos, Y., A. Nanopoulos, A. N. Papadopoulos and Y. Theodoridis (2005). R-Trees: Theory and Applications, Springer.

Rajaraman, V. and C. S. R. Murthy (2004). parallel computers: architecture and programming Prentice-Hall of India Pvt.Ltd

Shubin, Z., H. Jizhong, L. Zhiyong, W. Kai and F. Shengzhong (2009). Spatial Queries Evaluation with MapReduce. Grid and Cooperative Computing, 2009. GCC '09. Eighth International Conference on.

Yeung, A. K. W. and G. B. Hall (2007). Spatial Database Systems: Design, Implementation and Project Management, Springer.

Zhang, J., D. Papadias, K. Mouratidis and Z. Manli (2005). "Query processing in spatial databases containing obstacles." International Journal of Geographical Information Science 19(10): 1091-1111.

Zhang, P.-S. and X.-Q. Zuo (2009). "Research on Dealing with Query in Spatial Database." Geospatial information 7(6): 104-106.

Zhang, R., H. V. Jagadish, B. T. Dai and K. Ramamohanarao (2010). "Optimized algorithms for predictive range and KNN queries on moving objects." Inf. Syst. 35(8): 911-932.