2nd National Conference Recent Innovations in Science and Engineering (NC-RISE 17) ISSN: 2321-8169
Volume: 5 Issue: 9 09 – 14

Critical Analysis of Query Processing Algorithms in Mobile Environment

Mr. Ajay P. Chendke
Department of Computer Science,
D.C.P.E.,H.V.P.M.,
Amravati, India.
achendke2012@gmail.com

Dr. Mrs. S. S. Sherekar
Department of Computer Science,
SGB Amravati University,
Amravati, India.
SS_Sherekar@rediffmail.com

Dr. V. M. Thakare
Department of Computer Science,
SGBAmravati University,
Amravati, India.
vilthakare@yahoo.co.in

Abstract: Performance of query processing becomes slow caused by increasing number of data. Four main components involved in algorithm are assigning initial query, exploit query, assign to any possible queryand query matching. This paper is work on how query processing can be done in efficiently by dissimilartechniques used by algorithm and how the query can be processed.Nowadays,akey challenging forprocessing the queries and query answering technique.An approach minimizes the delay, fast access of queries and query process provide more security. Itintroducedhow to evaluate their performance, in terms of both the CPU execution time and the total number of operations. The purpose of this paperis to understand how algorithms behave to process the queries according with the request of the users. The performance of the algorithms depends in terms of aggregation accuracy, energy efficiency, and querydelay. This paper provides many unique characteristics and challenges for query processing algorithms in mobile environment.

Keywords*- Query processing,mobile computing,query optimization, algorithm, mobile environment.*

*****

I. INTRODUCTION:

Nowadays more people have started using mobile phone to access information they need from anywhere at anytime. In advanced mobile technology, Location Service lets users quickly pinpoint their location. There are three prominent factors which are time, place and purpose and between theirrelationship. The basic rules are to analyze essential data and algorithms to query processing. Query processing can be recommended to push the appropriate message to users. Advanced query processing for mobile environment and its optimization is an area which comes under the field of mobile computing.The fast development of wireless communication technology and battery powered portable devices had led to emergence of mobile computing as new tool data communication and processing [1]. The mobile environments are composed of wireless technologies in which user asked for query to be processed on query processing.The mobile environment is collection of mobile heterogeneous hosts, which are enabled to communicate using wireless links. These wireless links may change according to the natures of mobile networks, moreover, nodes in the ad-hoc network have to communicate without any centralized or help. The usability for the user will also change using query processing mechanism Thus, this mechanisms that allows the sharing of functionality among differentdevices in same environment will change the way of user interaction for searching fast query time in mobile computing environment [2]. The increasing success of relational database technology in data processing is suitable, in part, to the availability of nonprocedural languages, which can significantly improve application development and end-user productivity. By hiding the low-level details about the physical organization of the data, relational database languages allow the expression of complex queries in a concise and simple fashion. In particular, to construct the answer to the query, the user does not exactly specify the procedure to follow. This procedure is actually devised by a DBMS module, called as query processor. This relieves the user from query optimization, a time consuming task that is handled properly by the query processor [3].

Due to the advances in wireless communication and hardware technology, nowadays the mobile phone has become an integral part of our private life. More and more people have begun using mobile phone not only to communicate with each other but also to access the information they need from every place and every time. Moreover, a variety of services and tremendous information have been recommended to customers. Occasionally, they are beyond customer’s satisfaction [4].

The algorithms are effective for a large number of cached data items with low update rates. It also illustrates that the algorithms can be used with other complementary techniques to address the problem of cache invalidation for data items with varied update and access rates. The effectiveness of the algorithms can, however, is improved by applying complementary invalidation techniques to deal with dynamically changed update rates and patterns. Advances in wireless and mobile computing environments allow a mobile user to access a wide range of applications. Mobile users may want to retrieve data about unfamiliar places or local life styles related to their location [5]. These queries are called location-dependent queries. Furthermore, a mobile user may be interested in getting the query results repeatedly, which is called location-dependent continuousquerying.One of main features in sensor networks is the function that processes real time state information after gathering needed data from many domains. The component technologies consisting of each node called a sensor node that are including physical sensors, processors, actuators and power have advanced significantly over the last decade[6].However, sensor nodes in sensor networks are considerably constrained because with their energy and memory resources they have a very limited ability to process any information compared to conventional computer systems. Thus query processing over the nodes should be constrained because of their limitations [7]. Due to the problems, the join operations in sensor networks are typically processed in a distributed manner.

Advances in wireless and mobile computing environments allow a mobile user to access a wide range of applications. Mobile users may want to retrieve data about unfamiliar places or local life styles related to their location. These queries are called location-dependent queries. Furthermore, a mobile user may be interested in getting the query results repeatedly, which is called location-dependent continuous querying. This continuous query emanating from a mobile user may retrieve information from a single-zone or from multiple neighboring zones. The problem of handling location dependent continuous queries with the main emphasis on reducing communication costs and making sure that the user gets correct current-query result.

I.How hierarchical database framework (tree architecture and supporting continuous query algorithm) for handling location-dependent continuous queries.

II.How analyzing the flexibilityof framework for handling queries related to single zone or multiple zones and intelligent selective placement of location dependent databases [8].

III.How intelligent selective replication algorithm to facilitate time and spaceefficient processing of location-dependent continuous queries retrieving single zone information.

IV. How intelligent selective placement and selective replication model in terms of communication cost and storage constraints, considering various types of queries [9].

In many applications of mobile sensor networks, such as water flow monitoring and disaster rescue, the nodes in the network can move together or separate temporarily. The dynamic network topology makes traditional spanning-tree-based aggregation algorithms invalid in mobile sensor networks [10]. A distributed clustering algorithm which divides mobile sensor nodes into several groups, and then propose two distributed aggregation algorithms, Distance-AGG (Aggregation based on Distance), and Probability-AGG (Aggregation based on Probability). Both of these two algorithms conduct an aggregation query in three phases: query dissemination, intra-group aggregation, and inter-group aggregation. These two algorithms are efficient especially in mobile networks. Toevaluate the performance of the algorithms in terms of aggregation accuracy, energy efficiency, and query delay. Distance-AGG and Probability-AGG can obtain higher accuracy with lower transmission and query delay than the aggregation algorithms. Data sensing and retrieval in wireless sensor systems have a widespread application in areas such as security and surveillance monitoring, and command and control in battlefields. In query-based wireless sensor systems, a user would issue a query and expect a response to be returned within the deadline [11]. While the use of fault tolerance mechanisms through redundancy improves query reliability in the presence of unreliable wireless communication and sensor faults, it could cause the energy of the system to be quickly depleted.Therefore, there is an inherent trade-off between query reliability versus energy consumption in query-based wireless sensor systems [12]. The adaptive fault-tolerant quality of service (QoS) control algorithms based on hop-by-hop data delivery utilizing “source” and “path” redundancy, with the goal to satisfy application QoS requirements while prolonging the lifetime of the sensor system [13].

II. BACKGROUND OF QUERY PROCESSING:

The role of query processing in a mobile environment is to form a high level query on a distributed database, which is seen as a single query by to different devices by two different users; into an efficient execution strategy and finding the execution time take by the devices [14]. Location management is an important issue in query processing in mobile environment. Since the number of user population carrying mobile devices increases linearly with the service demand, the communication traffic for locating users also increases accordingly. This situation requires an efficient strategy for location tracking and management. Location management is a very essential factor for optimization of query processing mobile based environment [15].

The main duty of a relational query processor is to transform a high-level query (in relational calculus), into an equivalent lower level query (in relational algebra). The distributed database is of major importance for query processing since the definition of fragments is based on the objective of increasing reference locality, and sometimes parallel execution for the most important queries. The role of a distributed query processor is to map a high level query on a distributed database (a set of global relations) into a sequence of database operations (of relational algebra) on relational fragments. Several important functions characterize this mapping [16]:

- The calculus query must be decomposed into a sequence of relational operations called an algebraic query.
- The data accessed by the query must be localized so that the operations on relations are translated to bear on local data (fragments)
- The algebraic query on fragments must be extended with communication operations and optimized with respect to a cost function to be minimized. This cost function refers to computing resources such as disk I/Os, CPUs, and communication networks [17].

The low-level query actually implements the execution strategy for the query. The transformation must achieve both correctness and efficiency. The well-defined mapping with the above said functional characteristics makes the correctness issue easy. But producing an efficient execution strategy is more complex. A relational calculus query may have many equivalent and correct transformations into relational algebra. Since each equivalent execution strategy can lead to different consumptions of computer resources, the main problem is to select the execution strategy that minimizes the resource consumption [18].

III.QUERY PROCESSING ALGORITHM:

In mobile environment query processing algorithms plays its important role, it affects the complete performance of the system. The main parameters to be considered for performance measures are initial query, exploit query, assign to any possible query and query matchingin mobile environment. Query processing in a mobile environment is to form a high level query on different algorithms [19].

The query processing algorithms are able to give answers of different types of queries by exchanging less number of messages among the various component of the network [20].

Following are the various algorithms discussed for query processing in mobile environment.

3.1.Bongki Moonet. al.[20]presenteda five competent algorithmslike balanced tree algorithm for count aggregation, merge sort algorithm for max aggregation,bucket algorithm for large scale aggregation, parallel bucket algorithm, and parallel algorithm for large scale aggregation are recommended for query processing in mobile environment like a variety of temporal aggregation algorithms that overcome major drawbacks.

I.For small-scale aggregations, both are the worst-case and average-case processing time has been improved significantly.

II.For large-scale aggregations, the suggested algorithms can deal with a database that is substantially larger than the size of available memory.

III. The parallel algorithm designed on a shared-nothing architecture achieves scalable performance by delivering nearly linear scale-up and speed-up, even at the presence of data skew. The rate of increase in database size and response time requirements has out-paced advancements in processor and mass storage technology.

Another newsequential and parallel bucket algorithm suggested for large-scale aggregations relies on a novel data partitioning scheme, so that it can deal with a database substantially larger than the size of available memory.

The balanced tree and merge-sort aggregation algorithms have improved the worst-case and average-case processing time significantly for small databases that fit in memory. A parallel algorithm has been suggested for shared-nothing architectures for large-scale aggregations.

3.2 KyriakosMouratidiset.al. [16] presented acompetent algorithm is that shortest path algorithm based on nearest neighbor technique. If the object can compute it based on a shortest path algorithm and a stored map of the area. Although movement is continuous, we observed that updates are discrete; for example, there is a minimum time between two subsequent message transmissions from the same object. Concerning the server, it can either broadcast messages to all objects, or send unicast messages to objects individually.

As in most real-life applications, the cost of broadcasting is much lower than that of sending a large number of individual messages. Our micro level observation is how to minimize the communication cost, i.e., the number of messages exchanged between the server and the moving objects. In accordance with the study we can avoid the continual transmission of a large number of rapid data streams corresponding to location updates.

3.3 Murat Ali Bayiret.al. [15] presented the most popular metaheuristic techniques used for solving complex optimization problem is the genetic algorithm. GA simulates the evolution concept of biology. GA iteratively searches for an optimal solution by using evolutionary operations (also called genetic operators). At the beginning, a pool of random chromosomes is generated representing a set of different solutions. Then, genetic operators are applied to the pool of chromosomes to generate new chromosomes for the next iteration. Two genetic operators used in GA are as follows. *Crossover Operation:* In the crossover operation, sections of the parent chromosomes are exchanged to create new offspring chromosomes.

*Mutation Operation:*New chromosomes are also generated by randomly modifying a few genes of an existing chromosome. A selection technique must be used to determine which chromosomes survive to the next generation.The most commonly used selection techniques are as follows.

Roulette Wheel: All chromosomes of a generation are placed on a roulette wheel such that the area of the section of the wheelcorresponding to an individual chromosome is proportional to its fitness value.

Tournament: Randomly, r (r is the tournament size, and it must be at least two chromosomes are picked from the population, and the chromosome with the best fitness value is chosen for the next generation from the r-element group. This process is repeated as many times as the population size of the next generation. Using this technique, too, it is possible to select the same chromosome several times.

Truncation Selection: First, all the chromosomesare rankedfromthebesttotheworst according to their fitness values. Then, n chromosomesarechosenamongthetopTchromosomes each with the same probability. In addition, the selection function determines which solutions survive. These three parameters determine the efficiency of the solution in terms of its speed and the solution quality.

3.4 Nicolas Bruno et.al. [13] presentedon a top-k query is an ordered set of tuples, where the ordering is based on how closely each tuple matches the query.

Threshold Algorithm (TA) is the most well-known instance due to its simplicity and memory requirements. TA is based on an early termination condition and can evaluate top-k querieswithout examining all the tuples. This top-k query model is ubiquitous not onlyover middleware systems, but also over plain relational data.

The challenges are that must be addressed to adapt TAto a relational database system depending on the available indices, many alternative TA strategies can be used to answer a given query. Choosing the best alternative requires a cost model that can be effortlessly integrated with that of current optimizers.

Thesechallengesand evaluation of theresulting techniques by characterizing which scenarios can take advantageof TA-like algorithms to answer top-k queries in relational database systems.

IV.ANALYSIS AND DISCUSSION:

We proposed critical analysis and discussion of various competent algorithms for query processing in mobile environment on various parameters that are given as follows:

4.1.Bongki Moonet.al. [20] worked on algorithms like balanced tree algorithm for count aggregation, merge sort algorithm for max aggregation, bucket algorithm for large scale aggregation, parallel bucket algorithm, and parallel algorithm for large scale aggregation.

The suggested algorithms provide significant benefits over the current state-of-the-art in different ways. We have studied new sequential and parallel bucket algorithms based on novel data partitioning schemes. These algorithms can be used to compute temporal aggregates for databases that are substantially larger than the size of available memory, by processing data partitions in a sequential or parallel fashion.

In particular, with the adaptivedata partitioning scheme and the local and global meta arrays for partitioned data, we have also studied that the parallel bucket algorithm achieves scalable performance for large-scale databases by delivering nearly linear scale-up and speed-up, even at the presence of data skew.

We have observed that there are a few factorsthat affect the performance. It includes the percentage of long-lived tuples and the number of buckets used for data partitioning.

Although the suggested algorithms outperformedpreviousapproach consistently, irrespective of such conditions, we observed it is worth elaborating further on the issues. Additionally, we plan to study how to increase performance impacts ofsuch factors as initial data placement.

We also observed how to extend the data partitioning approach tospatio-temporal databases, which requires computing aggregates for data objects with two or more dimensional extents. Unlike the temporal aggregation, finally, the process of data partitioning and generating meta arrays will be more sophisticated.

4.2 KyriakosMouratidiset.al. [16]worked on shortest path algorithm which introduces and solves the problem of continuous k-NN monitoring over moving objects. Given a set of geographically distributed clients, the goal of the server is to constantly report the NN set of each client with the minimumcommunicationoverhead.

We observed that such methods which candeal withmultiple, staticor movingqueriesindependentlyof theunderlying distance definitions. Extensive evaluation, using Euclidean and road network settings used, it confirms that they are robust and efficient, even when compared with a, almost infeasible, optimal method.

We observed that the continuous monitoring of spatial queries will play a central role in numerous applications related to mobile computing and spatio-temporal databases. At the same time, it poses several interesting research issues. A direction for further analysis refers to the extension of the study other query types.

The client could require continuous reports about the value of some functions (e.g., number of cars, average traffic, and maximum speed) in its vicinity. Another promising direction refers to approximate k-NN monitoring for situations where the exact results are not necessary and the communication resources are limited.

We also observed how to used threshold-based techniques that adapt based on the trade-off between accuracy and network overhead.

4.3 Murat Ali Bayiret.al. [15]Worked on the following parameters have been used for generating different input sets: 1) number of queries (Q); 2) number of tasks in tasks pool (T); 3) number of plans in plans pool (P); 4) minimum and maximum and numbers of plans per query (PQm and PQM); 5) minimum and maximum numbers of tasks per plan (TPm and TPM); 6) minimumandmaximumvaluesforexecutiontimeoftasks(ETm and ETM).

We observed that how to use the minimum and the maximum numbers of tasks per plan and plans per query in generating random MQO instances. However, since mostly the averages of these two parameters (called as PQA and TPA, respectively), together with the number of queries, affect the size of the MQO problem instance.

The input generator generates the MQO problem instances by using the above parameters in the following order:

I. Tasks are generated randomly by using two parameters, namely, the total number of tasks and minimum and maximum execution time values of tasks. Then, for each task generated, its execution time is selected as a value between minimum and maximum execution time values of the tasks.

II. After the tasks have been generated, the input generator decides thenumber oftasksbyusingtheminimumandmaximumnumber of tasks per plans parameters. Although many plans may share the same task, it is guaranteed that no two plans have exactly the same set of tasks.

III. Finally, queries are generated randomly by using the parameters defining the number of queries and minimum and maximum number of plans per query.

Each query has its own distinct plan set, and therefore, a plan cannot solve more than one query. However,taskscanbesharedbydifferentqueriessincetherecan be plans solving different queries using the same task(s).

4.4 Nicolas Bruno et.al. [13] Worked on algorithm top-k queries in middleware systems could be adapted to work in the context of a RDBMS.

We observed how adapting the TAalgorithm to work inside the relational engine. As with traditional relational processing, the choice of indices critically determines the execution cost of TA strategies.

We also observed the space of execution alternatives, designed a cost model based on power laws to approximate the cost of each alternative, and provided pruning rules to speed up query optimization.

TAstrategies based on single-column indices are rarely more efficient than simple sequential scans over the underlying table unless the data is extremely correlated.

However, if one or more covering indices are available, the improvement given by the best TAstrategy can be of orders of magnitude. We also observed that for given query workloads, the cost of building a useful set of indices (once) would be remunerated by a large number of queries that subsequently benefit from such indices.

TA strategies to the set of algorithms that a relational optimizer can take advantage of to efficiently evaluate queries. One needs to be careful since each new index adds overhead for update statements in the workload, but this is the same trade-off that occurs in traditional query processing.

Threshold Algorithm is the most well-known instance due to its simplicity and memory requirements. TA is based on an early termination condition and can evaluate top-k queries without examining all the tuples. This top-k query model is ubiquitous not only over middleware systems, but also over plain relational data.

V. COMPARATIVE ANALYSIS OFDIFFERENT ALGORITHMS:

Different Algorithms behaviors and key factors on the basis of different parameter			
Balanced tree algorithm	Shortest path algorithm	Threshold algorithm	Genetic algorithm
1. Balanced tree algorithmis used for count aggregation, merge sort algorithm for max aggregation, and bucket	1. This algorithm is based on nearest neighbor technique. If the object can compute it based on a shortest path	1.Threshold Algorithm is the most well-known instance due to its simplicity and memory requirements. TA is based on an	1. The most popular meta heuristic techniques used for solving complex optimization problems is the genetic algorithm. GA

algorithm for large scale aggregation, parallel and parallel bucket algorithm, for large scale aggregation are recommended for query processing like a variety of temporal aggregation algorithms that overcome major drawbacks.	algorithm. The minimum time between two subsequent message transmissions from the same object. Concerning the server, it can either broadcast messages to all objects, or send unicast messages to objects individually.	early termination condition and can evaluate top-k queries without examining all the tuples. This top-k query model is ubiquitous not only over middleware systems, but also over plain relational data. .	simulates the evolution concept of biology. This simulation includes probabilistic methods using the principles of evolution.
2. For small-scale masses, both are the worst-case and average-case processing time has been improved significantly.	2. The cost of broad casting is much lower than that of sending a large number of individual messages.	2. TA strategies can be used to answer a query. Choosing the best other requires a cost model that can be effortlessly integrated with that of current optimizers. TA like algorithms to answer top-k queries in relational database systems.	2. In GA, the main data structure is a simple vector (chromosome) presenting a solution instance of a problem. The quality of the solution instance is defined by its closeness to the optimal solution
3. For large scale masses, the algorithms can deal with a database that is substantially larger than the size of available memory.	3. The number of messages exchanged between the server and the moving objects.	3. TA algorithm works for top-k queries in middleware systems could be adapted to work in the context of a RDBMS. It works on how adapting the TA algorithm to work inside the relational engine.	3. GA iteratively searches for an optimal solution by using evolutionary operations. Genetic operators are applied to the pool of chromosomes to generate new chromosomes for the next iteration.
4. Parallel algorithm designed on sharednothing architecture achieves scalable performance by delivering nearly linear scale-up and speed-up, even at the presence of data skew. The rate of increase in database size and response time requirements has out-paced advancements in processor and storage technology.	4. Shortest path algorithm which introduces and solves the problem of continuous k-NN monitoring over moving objects. Given a set of geographically distributed clients, the goal of the server is to constantly report the NN set of each client with the minimum Communication overhead.	4.This is based on space of execution alternatives, designed a cost model based on power laws to approximate the cost of each alternative, and provided pruning rules to speed up query optimization. It is based on single-column indices are rarely more efficient than simple sequential scans over the underlying table unless the data is extremely correlated.	4. Two genetic operators used in GA are as follows. Crossover Operation: In the crossover operation, sections of the parent chromosomes are exchanged to create new offspring chromosomes. Mutation Operation:New chromosomes are also generated by randomly modifying a few genes of an existing chromosome.
5. Parallel bucket algorithm suggested for large-scale aggregations relies on a novel data partitioning scheme, so that it can deal with a database substantially larger than the size of available memory.	5. It can deal with multiple, static, or moving queries freely of the underlyingdistance definitions. It confirms that are robust and efficient, when compared with almost infeasible.	5. It also works for given query workloads, the cost of building a useful set of indices (once) would be remunerated by a large number of queries that subsequently benefit from such indices.	5. Roulette Wheel: All chromosomes of a generation are placed on it, such that the area of the section of the wheel corresponding to an individual chromosome is proportional to its fitness value.
6. The balanced tree and merge-sort aggregation algorithms have improved the worst-case and average-case processing time significantly for small databases that fit in memory.	6. Threshold-based techniques that adapt based on the trade-off between accuracy and network overhead.	6. TA strategies to the set of algorithms that a relational optimizer can take advantage of to efficiently evaluate queries. One needs to be careful since each new index adds overhead for update statements in the workload, but this is the same trade-off that occurs in traditional query processing.	6. The various parameters 1) number of queries (Q); 2) number of tasks in tasks pool (T); 3) number of plans in plans pool (P); 4) minimum and maximum and numbers of plans per query (PQm and PQM); 5) minimum and maximum numbers of tasks per plan (TPm and TPM); 6) minimum and maximum values for execution time of tasks (ETm and ETM).

VI. CONCLUSION:

Balanced tree algorithm is used for count aggregation, merge sort algorithm for max aggregation, and bucket algorithm for large scale aggregation, parallel and parallel bucket algorithm, for large scale aggregation are recommended for query processing like a variety of temporal aggregation algorithms that overcome major drawbacks.

The role of Shortest Path Algorithm is introduces and solves the problem of continuous k-NN monitoring over moving objects. Given a set of geographically distributed clients, the goal of the server is to constantly report the NN set of each client with the minimumcommunicationoverhead.

Threshold Algorithmis used to answer top-k queries in relational database systems.This is based on space of execution alternatives, designed a cost model based on power laws to approximate the cost of each alternative, and provided pruning rules to speed up query optimization.

GA iteratively searches for an optimal solution by using evolutionary operations. Genetic operators are applied to the pool of chromosomes to generate new chromosomes for the next iteration.The most popular Meta heuristic techniques usedfor solving complex optimization problems.

Finally figure out that the query processing algorithms till date are significant but considering it from the future prospective there is need for making advancement in query processing algorithmsin mobile environment.

REFERENCES

[1] Soumi Chattopadhyay, Ansuman Banerjee,"QSCAS : QoS Aware Web Service Composition Algorithms With Stochastic Parameters", IEEE International Conference on Web Services,pp.1-8,2016.

[2] S.Venkata Lakshmi, Valli Kumari Vatsavayi,"Query Optimization Using Clustering And Genetic Algorithm For Distributed Databases", IEEE, pp.1-8,2016.

[3] Wenjiao Ban, Jiming Lin, Jichao Tong, Shiwen Li,"Query Optimization Of Distributed Database Based On Parallel Genetic Algorithm And Max-Min Ant System", IEEE 8th International Symposium on Computational Intelligence and Design,pp.1-5, 2015.

[4] Bin Cao, Jiaxing Wang, Jing Fan, Jianwei Yin, Tianyang Dong,"Querying Similar Process Models Based On The Hungarian Algorithm", IEEE transactions on services computing, Vol. 14, No. 8,pp.1-14 December 2015.

[5] Mi young Jang,Sune-Je Jang,Jae-Woo Chang,"A New K-NN Query Processing Algorithm Enhancing Privacy Protection In Location Based Services", IEEE First International Conference on Mobile Services,pp.1-8,2012.

[6] Ing Ray Chen, Anh Phan Speer, Mohamed Eltoweissy,"Adaptive Fault-Tolerant Qos Control Algorithms For Maximizing System Lifetime Of Query-Based Wireless Sensor Networks", IEEE transactions on dependable and secure computing, Vol. 8, No. 2, pp.161-176, March-April 2011.

[7] Jiaheng Lu,Tok Wang Ling,Zhifeng Bao,Chen Wang,"Extended XML Tree Pattern Matching: Theories And Algorithms",IEEE transactions on knowledge and data engineering, Vol. 23, No. 3,pp.402-416, March 2011.

[8] Fan Yuanyuan Mi Xifeng "Distributed Database System QueryOptimization Algorithm Research", IEEE,pp.1-4,2010.

[9] T.V. Vijay Kumar, Vikram Singh,Ajay Kumar Verma,"Generating Distributed Query Processing Plans Using Genetic Algorithm", IEEE, International Conference on Data Storage and Data Engineering, pp. 173-177,2010.

[10] Ke Yi,Feifei Li,George Kollios,Divesh Srivastava,"Efficient Processing Of Top-K Queries In Uncertain Databases With X-Relations", IEEE transactions on knowledge and data engineering, Vol. 20, No. 12,pp.1669-1682, December 2008.

[11] Ken C.K. Lee, Baihua Zheng,Wang-Chien Lee,"Ranked Reverse Nearest Neighbor Search", IEEE transactions on knowledge and data engineering, Vol. 20, No. 7,pp.894-910, July 2008.

[12] Sudipto Guha,Kyuseok Shim,"A Note on Linear Time Algorithms for Maximum Error Histograms",IEEE transactions on knowledge and data engineering, Vol. 19, No. 7, pp.993-997, July 2007.

[13] Nicolas Bruno, Hui(Wendy) Wang,"The Threshold Algorithm: From Middleware Systems To The Relational Engine",IEEE transactions on knowledge and data engineering, Vol. 19, No. 4, pp.523-537, April 2007.

[14] Ahmed Elmagarmid, Jin Jing, Abdelsalam (Sumi) Helal, and Choonhwa Lee, "Scalable Cache Invalidation Algorithms For Mobile Data Access", IEEE transactions on knowledge and data engineering, Vol. 15, No. 6, pp. 1498-1511, November/December 2003.

[15] Murat ali bayir,ismail h. toroslu,ahmetcosar,"Genetic Algorithm For The Multiple-Query Optimization Problem",IEEE transactions on systems, man, and cybernetics part c: applications and reviews, Vol..37, No.1, pp.147-153, January 2007.

[16] Kyriakos Mouratidis,Dimitris Papadias,Spiridon Bakiras,Yufei Tao,"A Threshold-Based Algorithm For Continuous Monitoring Of K Nearest Neighbors", IEEE transactions on knowledge and data engineering, Vol. 17, No. 11, pp.1451-1464, November 2005.

[17] Manish Gupta,Manghui Tu,Latifur Khan,Farokh Bastani,I-Ling Yen,"A Study Of The Model And Algorithms For Handling Location-Dependent Continuous Queries",Springer-Verlag 2005.

[18] Wang Lian,David Wai-lok Cheung,Nikos Mamoulis,Siu-Ming Yiu,"An Efficient And Scalable Algorithm For Clustering XML Documents By Structure", IEEE transactions on knowledge and data engineering, Vol. 16, No. 1, pp.82-96,January 2004.

[19] Sineenard Pinyapong, Toshikazu Kato, "Query Processing Algorithms For Time, Place, Purpose And Personal Profile Sensitive Mobile Recommendation", IEEE, International Conference on Cyberworlds,2004.

[20] Bongki Moon,Ines Fernando Vega Lopez,Vijaykumar Immanuel, "Efficient Algorithms For Large-Scale Temporal Aggregation", IEEE transactions on knowledge and data engineering, Vol. 15, No. 3, pp.744-759, May/June 2003.