Developing Xquery Model for Distributed Query Processing in Mobile Networks

T.P.Andamuthu
Professor
Maharaja Engg college for Women, Perundurai, Tamilnadu, India

P.Balasubramanie
Professor
Kongu Engg College, Perundurai, Tamilnadu, India

ABSTRACT
The emergence and popularity of mobile computing environment, so get a variety of semi-structured data to follow some common XML model. The Extensible Markup Language (XML) model has recently gained huge popularity because of its ability to represent a wide variety of structured and semi-structured data. Several Query languages have been proposed for the XML data model, the most-widely known is XQuery. Traditional query processing to a database focused on structured data retrieval and structures to support them. In this paper we present a model and an algorithm for querying structured and semi - structured data for mobile computing environments based on the model of XQuery. We employ a variety of servers to handle different jobs. Buffer is maintained in the mobile node and the cache is stored in the query server, and mobile server. Priority is given to requests based on various parameters such as priority by the user, the required bandwidth, etc. parameters considered for performance measure of the effectiveness of the request, delivery ratio and average power consumption and the results show that the proposed algorithm works better than existing systems.

General Terms
Mobile Computing, Distributed Computing, Databases

Keywords
Query Processing, Mobile computing, cache, query efficiency etc.

1. INTRODUCTION
Today, small notebooks, PDAs and smart phones are getting smaller, portable solution possible to access digital information in an easy manner. Mobile terminals communicate and interact with other terminals via wireless networks such as Wi-Fi or Bluetooth. They may spontaneously network with each other through their proximity and perform proximity applications [1-10]. These devices will become both sources and consumers of data and can communicate with other devices to meet individual or collective computing. In such a mobile environment the special elements of the network is very dynamic and can be extremely volatile. Traditional query processing techniques[1-12], the global data and statistics collected according to the chart and the histogram is no longer such a very mobile environment, sufficient for many reasons, such as: limited resources, and powerful servers, the data types of terminals ranging from PDAs to communicate the type of energy to a higher bandwidth or limited this may be the heterogeneity bandwidth to a wired network with wireless network. In addition, the query optimization and execution of queries in order to evaluate effectively the account must take into account available resources. The number of small devices such as PDAs, smart phones, and sensors of different types of data moving rapidly grows with them. access to computing and communications is not only required a local, but also the way to the user from one location to another.

The database is organized by a set of logically related data. Traditional structured database using the relational model [1] Relational database access or manipulate with a very high level language called SQL. Statement of the kind referred to in the request specifies the data obtained from the database. For query processing, system software, database management systems (DBMS), generates a number of implementation plans, choose one with the lowest cost, and then executes it to retrieve the requested data. The relational model has many advantages, including simplicity, rigorous mathematical justification and ease of programming and use. However, it is also a lot of limitations, including poor modeling capabilities. To overcome these drawbacks, new and powerful data model with semantics is richer than the relational one, were introduced.

Among those models are the deductive [2], object-oriented [3] and XML [4]. The XML (Extensible Markup Language) model organizes data into documents. The XML data model has gained huge popularity because of its ability to represent a wide variety of unstructured and semi-structured data as well as its use in integrating different data sources (traditional / relational databases, data files, email messages, web pages …etc.) for display on a variety of devices, including personal computers, personal digital assistants (PDAs) and smart mobile phones. Several Query languages have been proposed for the XML data model, the most widely used one is XQuery [5]. XQuery is based on XPath in which requested data is represented in some order of requirements of the Xquery language. Recursive queries are quite important in the context of XML databases.

Facilitating ubiquitous data access at any point of time to the user is the thrust area of all such researches[22-25]. Critical constraints and limitations of connectivity, unreliable communication, low battery and memory, slow processing, storage and moderate limited screen size presented as challenges in data management for mobile computing, despite their privileges and advantages. These factors require a condition of adaptability of mobile devices to the network environment available today. Location dependent data access by the user when the devices are on the way somewhere is a prominent feature of mobile computing applications. The distributed query processing systems assume an 'always-on' situation in the network, which can seriously compromise and amended by intermittent connectivity. An error in the current system is the lack of network connectivity. At times, the instantaneous transmission of accumulated data available since the last upload, through the connection may not be feasible. The possibility of unwanted irrelevant information that has priority over vital information can not be ruled out. Priority data for transmission is essential for the mobile nodes for the beneficial use of connectivity in the binding context of bandwidth-limited and uneven. Node connectivity (server) to Manet, it is not forever. The minimum requirement to stay connected to the node network, it must have sufficient power to operate and the ability to hear the broadcast of another node on the network at least. Network nodes (servers) can operate in one of three modes of power reduction that is designed, 'forward', 'receive' and 'standby'. In 'send' more energy is used for sending and receiving messages. 'Receive' mode allows the processing of data and receive messages while no processing, transmission or reception of CPU happen in 'standby' mode.

The whole geographical area is divided into cells. These cells are roughly circular in shape. At the center of each cell is a Base Station (BS), also referred to as a server, which communicates with the mobile devices in its cell area through the wireless channel. Each BS serves an area and those areas are connected via a wired network as referred to earlier [12]. When a mobile device moves from one cell to another it begins communicating with the new BS in the new cell. In a centralized mobile database the database resides in the central server or BS. A mobile user can get data from the server in two methods: pull-based and push-based. In a pull-based method there are two channels namely an uplink channel and a downlink channel, which is also called the pull channel. A mobile device sends the query to the server via the uplink channel and the query results come to the device via the downlink channel. The downlink channels are private to each mobile device. In a push-based method the server broadcasts the data on a broadcast channel and the mobile devices tune to that channel to retrieve their necessary information [12-15]. In this case the server has to periodically broadcast information to the clients. In a hybrid model, the push-based method is extended by using an uplink channel via which clients can send explicit requests [12-15].

In this paper, we propose the system with different servers for different job processing like one server for processing location based queries and another server for processing data queries. The queries are prioritized based on different parameters like the preference given by the user, size of the result to be transmitted, bandwidth requirement, time to transmit and finally time of request. The more important issue is the bandwidth is reused based on the reusability concept. The nodes in the system are distinguished as mobile node, mobile server and the query server. The parameters considered for performance evaluation are query efficiency, delivery ratio and average power consumption.

2. BACKGROUND

A method for secure query processing in mobile databases has been presented in [12]. Queries are sent to the server through point-to-point channels by the mobile devices. Computation of the superset of results of many such queries is made by the mobile database server for broadcasting through a channel of larger bandwidth. The mobile devices can retrieve the required information by tuning in to that channel. Senders of the queries can view the results of both their queries and those of others in the same group. The undesirable aspect in this is, malicious users have access to the results f the queries sent by some other users. In (1) proposal of a method is made that denies any user doing so even when the results of multiple queries are broadcasted through the same channel and are accessible to all users. In [13], the authors discussed the various challenges in distributed processing of location dependent continuous queries in a mobile environment by studying the different scenarios in which both the querying unit and the object being queried are in motion. The authors discussed a classification of different s categories of location dependent queries and various solutions to such complex queries. Caching techniques enable faster access to data, minimizing the huge network traffic created because of the processing of location dependent queries in a wireless environment. The query results are cached for a further reduction of the data transfer and reusability of the results. The thrust area in this is the methods of processing and application of such location dependent queries (2). The study presents and classifies various methods of processing of different location dependent queries and data management problems in evaluation of such queries.

A continuous query processing system for intermittently connected mobile networks that comprises of a delay-tolerant continuous query processor distributed across the mobile hosts has been proposed in [14]. In addition, a mechanism for prioritizing query results has been designed that guarantees enhanced accuracy and reduced delay. The conventional continuous query processors send the results of continuous queries instantaneously over the network, whereas the query processor in [14] stores them in an output buffer.

Dorian C. Arnold Barton P. Miller [17] have proposed a scalable failure recovery model for data aggregations in large scale tree-based overlay networks (TBONs). They have formalized the TBON model and its fundamental properties to prove that our state compensation model properly preserves computational semantics across TBON process failures.

Dragan Stojanovic et al. [18] address the problem of processing continuous range queries over mobile objects, whose motion is constrained by a spatial network. The query range represents the user-selected area, the map window, the polygonal feature, or the area specified by the distance from a reference point of interest. In contrast to regular queries that are evaluated only once, a continuous query remains active over a period of time. A major challenge for this problem is how to provide efficient processing of continuous queries with respect of CPU time, I/O time and main memory utilization.

Wei-Shinn Ku et al. [19] proposed two novel algorithms for processing k nearest neighbor queries and range queries on spatial networks with privacy protection. The main idea is to hide the exact mobile user location with a cloaked region. The cloaked region covers the query requester and at least K − 1 other users based on the K-anonymity concept. The spatial queries are executed based on both the cloaked region and the underlying networks.

Giuseppe Amato et al. [20] proposed approach the WSN can be programmed using a query language (MW-SQL), which offers constructs, specialized for sensor networks. The query language is offered by a JDBC driver which is encapsulated within the OSGi framework.

3. DATA MODELS

3.1 Relational Model

The relational model [1] organizes data into relations stored as two-dimensional tables. The column headers of a table represent the attributes of the relation, whereas, the rows are the relation's tuples. A table in the relational model can also be viewed, as shown in Figure 2, as a three-level tree where the root node represents the table name, the 1st level nodes represent its rows and the 2nd level nodes represent the individual values contained within each one of these rows.

Relational databases are accessed using a very high-level language called Structured Query Language (SQL). A statement of this type, referred to as a query, specifies the data to be retrieved from the database. To process a query, a software system, the database management system (DBMS), generates a number of execution plans, selects the one with the least cost and then executes it. As a result the data that satisfy the query is returned to the user. In relational databases, a given query plan consists of a sequence of operations called the relational algebra operations (select, project, join, transitive closure ...etc.). The execution of these operations, especially join and transitive closure, against very large collection of data is generally slow. Many serial and parallel algorithms have been proposed to efficiently execute these operations. Some of these algorithms and their performance evaluation can be found in [6] – [12].

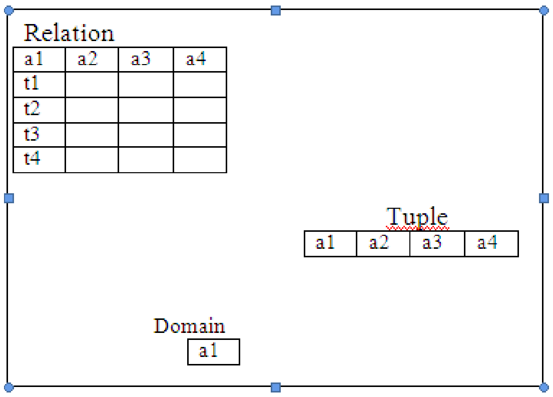

Fig.1: Relational Model Representation

3.2 XML Model

Many solutions for XML data management use relational database technology as the underlying storage medium. Supporting the ordered nature of the XML data in the relational model context is an issue since order information is lost while converting from XML to the relational data representation. Many solutions for semi-structured data have been extended to support XML data. These solutions tend not to support XQuery, and more importantly, do not support order requirements of XQuery expressions. Concurrently with these efforts to exploit existing database technologies, native XML storage manager systems have also been proposed. An advantage of such native storage is that XML documents may be clustered in physical XML document order, thus facilitating efficient children/descendant access.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
<book category="COMPUTER">
  <title lang="en">Computer Programming</title>
  <author>Andamuthu</author>
  <year>2011</year>
  <price>300.00</price>
</book>
<book category="GENERAL">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>500</price>
</book>
</bookstore>
```

Fig.2: XML model Representation

3.3 Query Processing For Structured Data

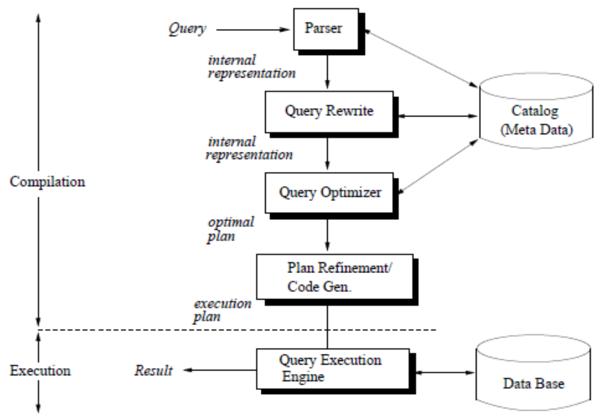

Fig.3: Phases of Query Processing

Query processing deals with designing algorithms that analyze queries and convert them into a series of data manipulation operations. The main function of a query processor is to transform high-level query (typically, in relational calculus) into an equivalent lower-level query (typically, in some variation of relational algebra). Fig. 3 shows the classic architecture for query processing. This architecture can be used for any kind of database system including centralized, distributed, or parallel systems. The query processor receives a query as input, translates and optimizes this query in several phases into an executable query plan, and executes the plan in order to obtain the results of the query. We use this architecture for mobile computing environments [22-25].

4. XQUERY MODEL BASED QUERY PROCESSING

4.1 System Model

In the proposed system, we have mobile nodes, mobile servers and centralized XQuery servers. The whole geographical area is divided into cells and the cell shape is assumed to be hexagonal. Mobile nodes are the nodes that belong to a particular cell of the network. Mobile server is similar to the Base station of the cellular network. Mobile users directly communicate with the mobile servers. The XQuery servers are similar to the MSC in the cellular network. Mobile servers directly communicate with query servers. Buffer is maintained in the mobile node. Mobile servers and query servers maintain the cache. The queries are classified as location based queries and data queries. Location based queries are the queries related to the location of the mobile node, about the traffic or about the information in the specified location to get the particular object etc. Data queries are the queries to retrieve some information from the mobile nodes.

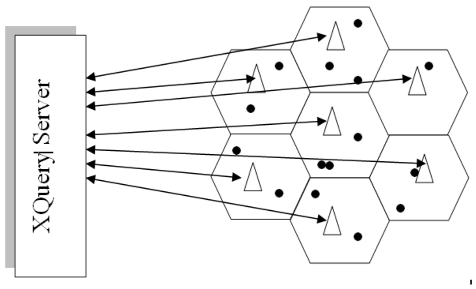

Fig. 4 System Model

When the mobile nodes require a piece of information, the query is sent to the mobile server which is in its same cell. The queries at that instant of time are prioritized based on the user specification. If some queries are given the same priority by the users, then they are sorted according to the time of request. Even then if some queries are equally prioritized, they are sorted based on the estimated bandwidth requirement which depends on the size of the result to be transmitted. Finally if any collision occurs in prioritization random method is used. After the queries are prioritized, queries are optimized and processed in the mobile server if possible. The queries can be processed in the mobile server if the required information is fully available in the server. Otherwise the query is forwarded to the Xquery server based on the type of the query. If the query is location based, it is sent to location query server otherwise it is sent to the data query server. The query is processed by the corresponding query server and cached. The output is sent to the mobile server. The mobile server also caches this information and sends the output to the required mobile node. The importance of the cache here, data will be ready if similar queries occur. This improves the performance of the system to maximize the delivery rate. The queries also need not be sent to the query server one at a time. More number of queries can be sent at a time to the query server by the mobile server by maintaining the buffering system. This is again done based on the priority. If the query is highly prioritized, the processing takes place immediately otherwise it is buffered for a bit of time period. This process reduces the power consumption as the number of times to transmit and receive data will be reduced.

```
1. mobile node sends a query to mobile server
2. mobile server categorize the query
3. Buffer the query based on the category
4. Is data (location based) buffer?
  i. prioritize the queries in the buffer
  ii. If (can be processed in the mobile server?) then
    1. queries are processed
    2. cache the results
    3. broad cast the results
  iii. else
    1. queries are forwarded to the data (location) query
       server
    2. queries are processed
    3. cache the results
    4. sent to the mobile server
    5. Mobile server broadcasts the results.
```

Our proposed approach supports different types of XQuery order, namely document order and order imposed by the query itself in a variety of ways. A key point in our solution is that caching the queries and its updates. Caching popular queries and reusing results of previously computed queries are one important query optimization technique, especially in modern distributed environments such as mobile and ubiquitous environments. Based on the recent proliferation of XML data and the emergence of the XQuery language, developing a system based on a query caching for XQuery and rewriting strategies to respond to the user entering based query caching XQueries, whenever possible, instead of accessing remote XML data sources provides effective information retrieval.

5. RESULTS AND DISCUSSIONS

This section presents experimental evaluation of our DQPC architecture when XQuery model is used. In order to test our protocol, The NS2 simulation software [10] is used. NS2 is a general-purpose simulation tool that provides discrete event simulation of user defined networks. The network nodes were placed uniformly at random within a square of 1000 meters. We varied the average speed of the mobiles from 10, 20, …, 50 in order to study the impact of server mobility. The data broadcast size is varied from 1000, 2000… 5000. In all the experiments, we used the following evaluation criteria:

5.1. Simulation Parameters

Average Power Consumption The average power consumed by clients and the average power consumed by servers are calculated.

Query Efficiency The data pull section will rely on the measurement of query efficiency. This is a measure of the percentage of data queries that get served during an entire simulation.

Delivery Ratio which is the ratio of results received by the clients and the no. of queries sent.

5.2 Simulation Results

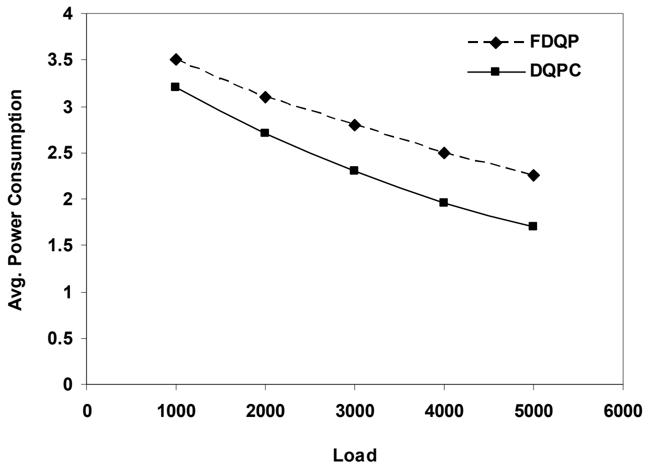

Fig. 5 Load vs Average power consumption

In Fig 5, when the server load increases, the average power consumption for server decreases as less time is spent for transmitting. The results in Fig. 2 shows that the proposed algorithm DQPC performs better than the FDQP (Fault-tolerant distributed query processor) because of caching. It can be observed that the performance increases as the load increases. This is because of the caching property. As the number of accesses increases, the cache hits occur then the performance increases. So, the results are mostly dependent on the cache performance.

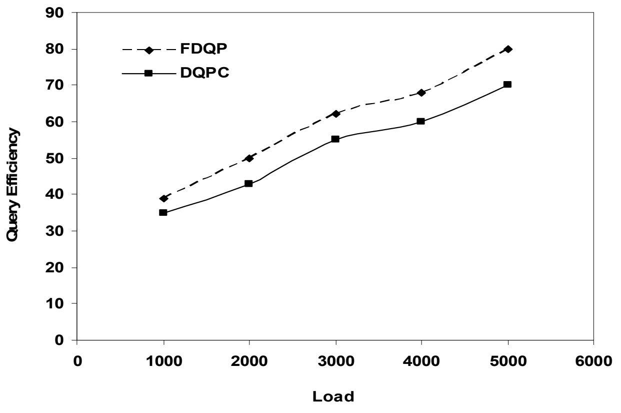

Fig. 6 Load vs Query Efficiency

Fig. 6 shows that the performance of FDQP and DQPC increases as the system load or broadcast size increases and it also shows that the proposed algorithm DQPC increases the efficiency of the query processing when compared to the FDQP.

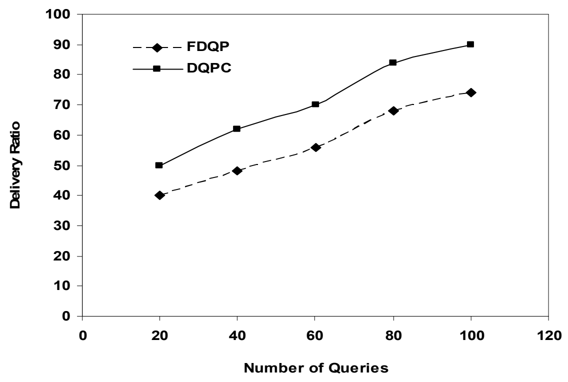

Fig. 7 Number of Queries Vs Delivery Ratio

The delivery ratio of the client in FDQP is compared with delivery ratio of the mobile server as the role of client and mobile host server in FDQP is played by the mobile server in DQPC. The results in Fig. 7 show that the delivery ratio is increased with the proposed algorithm.

6. CONCLUSIONS

X-Query based distributed query processing algorithm based on caching technique is proposed in this paper. The different servers are maintained for query processing based on the type of the query. Cache is maintained at both mobile server and query server. At the same time buffering technique is implemented at the mobile server. The queries are prioritized to improve the performance. The main advantage of the proposed algorithm is cache technique. The advantage of using X-Query model is its easy to implement and extend to mobile computing platforms because of its interoperable and portable properties. The simulation is carried out and compared with the fault-tolerant distributed query processing algorithm and is shown the performance of the system is improved.

7. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of this paper.

8. REFERENCES

[1] Agrawal R., S. Dat and H. Jagadish,(1990)“Direct Transitive ClosureAlgorithms:DesignandPerformanceEvaluation,” ACM Trans. Database Sys, Vol. 15, No. 3,pp. 427-458,.

[2] Bancilhon.F,(1998) “Object-oriented Database Systems,” Proc. ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, Austin, Texas.

[3] Bray.T,(2004) “Extensible Markup Language (XML) 1.0(Third Edition),” W3C Recommendation.

[4] Cod, E.F (1970) “A Relational Model of Data for Large Shared Data-banks,” Comm. Of ACM, Vol. 13, No. 1, pp. 377-387.

[5] Dewitt D and D. Schneider,(1989)“A Performance Evaluation of Four Parallel Join Algorithms in a Shared-nothing Multiprocessor Environment,” Proceedings of ACM Conference on Management of Data (SIGMOD).

[6] Gallaire.H, J. Minker, and J. M. Nicolas,(1984) “Logic and Databases: A Deductive Approach,” ACM Computing Surveys, Vol. 16, No. 2.

[7] Jae-Gil Lee , Kyu-Young Whang, (2006) “Secure query processing against encrypted XML data using Query-Aware Decryption”.

[8] Maged El-Sayed , Katica Dimitrova , Elke A. Rundensteiner, (2005) “Efficiently supporting order in XML query processing”. Elsevier- Data & Knowledge Engineering, pp. 355–390.

[9] Qadah G. Z. and K. B. Irani,(1988) “The Join Algorithms on a Shared – memory Multiprocessor Database Machine,”IEEE Transactions on Software Engineering, Vol. 14, No. 11, pp. 1668-1683.

[10] Qadah G. Z, L. J. Henschen and J. Kim,(1991) “EfficientAlgorithms for the Instantiated Transitive ClosureQueries,” IEEE Transaction on Software Engineering, Vol. 17, No. 3, pp. 296 – 309.

[11] Qadah G. Z. and J. Kim, (1992) “The processing of a class oftransitive closure queries on uniprocessor and sharednothing multiprocessor systems,” Data & KnowledgeEngineering, pp. 57 – 89.

[12] D.Saha and N. Chowdhury, “A Method for Secure Query Processing in Mobile Databases”, International Association of Engineers, Volume 14, Issue 1, Engineering Letters, 14:1, EL_14_1_20, February 2007.

[13] Tabassum K, Hijab M, Damodaram A, “Location Dependent Query Processing – Issues, Challenges and Applications”, International Conference on Computer and Network and Technology (ICCNT), April 2010, pp: 239 – 243. DOI: 10.1109/ICCNT.2010.39

[14] T. P. Andamuthu and Dr. P. Balasubramine, “A Delay-Tolerant Distributed Query Processing Architecture for Mobile Environment”, International Journal of Computer and Information Engineering, volume 4, issue 2, 2010, pp: 86 – 90.

[15] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik. “Broadcast Disks: Data Management for Asymmetric Communication Environments.” Proceedings of ACM SIGMOD Conference, San Jose, CA, May 1995.

[16] S. Acharya, M. Franklin, and S. Zdonik. “Balancing Push and Pull for Data Broadcast.” Proceedings of ACM SIGMOD Conference, Phoenix, AZ, May 1997.

[17] Dorian C. Arnold Barton P. Miller “A Scalable Failure Recovery Model for Tree-based Overlay Networks”, 2007 ACM.

[18] D. Stojanović et al. "Continuous Range Query Processing for Network Constrained Mobile Objects", Proceedings of the 8th International Conference on Enterprise Information Systems (ICEIS 2006), May 24- 27, Paphos, Cyprus, 2006, pp. 63-70.

[19] Wei-Shinn Ku “Privacy Protected Query Processing on Spatial Networks”, 2007 IEEE 23rd International Conference on Data Engineering Workshop, April 2007.

[20] Giuseppe Amato et al. “Enabling Context Awareness through Distributed Query Processing in Wireless Sensor Networks”, 2nd International Workshop on Requirements and Solutions for Pervasive Software infrastructures, September 16, 2007, Innsbruck, Austria.

[21] Sandeep Prakash , Sourav S. Bhowmick , Sanjay Madria b, (2006) “Efficient recursive XML query processing using relational database systems”. Elsevier-Data & Knowledge Engineering ,pp.207–242.

[22] Shapiro D. I.,(1986) “Join Processing in Database Systems with Large Main Memories,” ACM Trans. Database Sys, Vol. 11, No. 3, pp. 239-264.

[23] Toroslu, I. H. and Qadah, G. Z.,(1996) “The Strong Partial Transitive-Closure Problem: Algorithms and Performance Evaluation,” IEEE Transaction on Knowledge and Data Engineering, VOL. 8, NO. 4,pp. 617 – 629.

[24] Guilherme Figueiredo, Vanessa Braganholo and Marta Mattoso, “A Methodology for Query Processing over Distributed XML Databases”, www.dcc.ufrj.br/~braganholo/artigos/RT-guilherme.pdf