Adaptive Query Processing in Mobile Environment

Hocine Grine
University of Valenciennes
59313 Valenciennes cedex 9, France
hocine.grine@univ-valenciennes.fr

Thierry Delot
University of Valenciennes
59313 Valenciennes cedex 9, France
thierry.delot@univ-valenciennes.fr

Sylvain Lecomte
University of Valenciennes
59313 Valenciennes cedex 9, France
sylvain.lecomte@univ-valenciennes.fr

ABSTRACT
These last years, the evolution of nomadic terminals and mobile networks has yield to the development of the ubiquitous computing. In this context, actual query evaluation and optimization techniques in distributed databases based on the use of a global schema and a cost model are no more relevant. Furthermore, a query processor deployed in this type of environment must face a significant heterogeneity, in particular mobile terminals, communication networks and different data sources. In order to cope with this heterogeneity, such service should have the capability to adapt itself dynamically. In this paper, we describe the problems related to query processing in mobile environment and the needs for adaptability.

Categories and Subject Descriptors
C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*distributed applications*; H.2.4 [**Database Management**]: Systems—*query processing*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*distributed systems*

General Terms
Query Processing, Mobility, Adaptability

Keywords
Adaptive Query Processing, Mobile Computing

1. INTRODUCTION
Today small notebook PCs, PDAs (Personal Digital Assistants) and smart phones are becoming more and more small and portable making possible the access of digital information any time, any where [22]. Mobile terminals are communicating and interacting with other terminals through wireless networks like WiFi or Bluetooth.

Mobile devices can spontaneously network with one another within their proximity and execute proximity applications. These devices will become both sources and consumers of data, and can interact with other devices in order to perform individual or collective computing. In such mobile ad hoc environment, elements of the network are very dynamic and can be extremely volatile. For example, a mobile user looking for a restaurant will obtain different results based on the time and the place he issued the query. As the location of other devices changes with respect to other entities and data sources are constantly in movement it may not be possible to collect information about available data sources at any given point of time. Traditional query processing techniques based on global data schema and collected statistics and histograms are no more adequate in such highly mobile environment for many reasons such as: terminals heterogeneity ranging from a PDA with limited resources and energy to puissant servers, data types (time, place, IDs of physical entities), communication type that can be a wired network with a higher bandwidth or a wireless network with a limited bandwidth. Furthermore, query optimization and execution must take into account available resources in order to effectively evaluate queries. For querying in such environment, query processors shall have the capacity to face these constraints and adapt themselves in order to ensure a QoS that conforms user preferences terminal constraints and network characteristics.

The remainder of the paper is organized as follows. In Section 2, we introduce mobile computing systems. In Section 3 data management challenges in mobile environments are given. Section 4 defines the current work on query processing. Section 5 defines adaptive query processing in mobile environment. Finally, a conclusion is given in Section 6.

2. MOBILE COMPUTING SYSTEMS
2.1 Mobile Computing Environments
The number of small devices like PDAs, smart phones and sensors grows rapidly carrying with them different sort of data. The access to computing and communications is necessary not only from the local one, but also while the user is moving from one place to another.

Mobile computing deals with the mobility of hardware, data and software in computer applications. It is a specialized class of distributed systems where some nodes can disengage from joint distributed operations, move freely in the physical space and reconnect to a possibly different segment of a computer network at a later stage in order to resume suspended activities.

MPAC '05, November 28- December 2, 2005 Grenoble , France

The goal of mobile computing is to permit users and programs to be as effective as possible in this environment that is characterized by uncertain connectivity and heterogeneity, without changes to the manner in which they operate.

Figure 1 shows the existing and widely architectural model of a system that supports mobile computing. This architecture consists of stationary and mobile components. The only mobile component is the mobile unit. A *Mobile Unit* is a mobile computer which is capable of connecting to the fixed network via a wireless link. Stationed hosts are connected together via a high-speed network (Mbps to Gbps). Components in the fixed network are either *Fixed Hosts* that are not capable of connecting to a mobile unit, or *Base Stations* which are computers capable of connecting with a mobile unit and are equipped with a wireless interface (they are also known as *Mobile Support Stations*) .

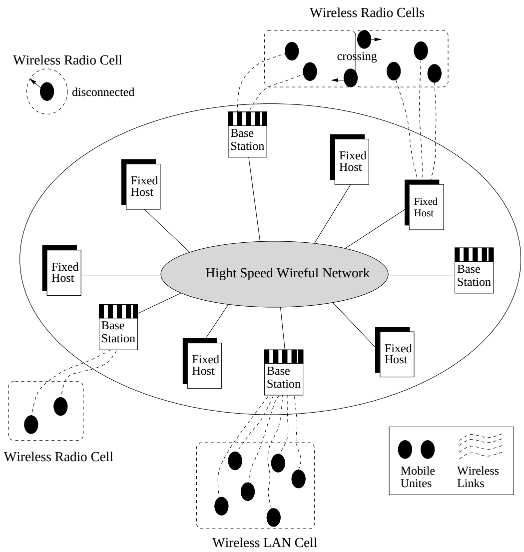

Figure 1: Mobile Computing Architecture

2.2 Mobile Computing Applications

Mobile applications can be categorized depending on whether they use fixed or radio communication services. Nomadic systems are typically based on wired dial-up, or local area network communication facilities. Mobility is not transparent, requiring a new connection to be explicitly established when the user moves to a new location.

Mobile systems use wireless technology for transparent communications while travelling in a train, car, plane or even while walking. During the course of a connection the radio reception is likely to vary considerably.

Location-aware services and applications require information on a user's geographical location in order to display a position on a map or provide information. This requires a generalized positioning solution, to track the current position of the user. This can be accomplished using GPS, cellular telephone base stations, active badges, or determining which fixed computer is being used [7].

Mobile computing devices may also need access to local servers supporting electronic mail, printing, file service or databases. This could imply the need to migrate resources from the user's home servers to local ones, rather than just maintaining network connections to the home servers, in order to reduce communication costs.

2.3 Mobile Computing Characteristics

Mobile computing environment has some characteristics that make the system unique and fertile area of research. The main issues in mobile computing introduced by [5] are mobility, wireless communication and portability:

2.3.1 Mobility

The location of mobile units is an important parameter when locating a mobile station that may hold the required data and when selecting information especially for location dependent information services. In this case the search cost, to locate mobile units, is added to the cost of each communication involving them.

2.3.2 Wireless Medium

Wireless networks offer a smaller bandwidth than a wired networks; the first ones offer a bandwidth that varies between 9 and 14 Kbps. while any Ethernet offers a bandwidth of 10 Mbps. However, there is an asymmetry in the communication because the bandwidth for the downlink communication (from servers to clients) is much greater than the bandwidth for the uplink communication (from clients to servers).

Mobile environment is subjected to frequent disconnections. These disconnections can be classified in two types: (i) forced disconnections, which are usually accidental and unavoidable, like for example those disconnections that take place when entering an out-of-coverage area. (ii) Voluntary disconnections, that take place when the user decides to disconnect his unit with the goal of saving energy.

2.3.3 Portability of Mobile Elements

The design of portable computers implies that they must be small, light, consume little energy, etc. This causes these computers to have generally more limited functionalities than fixed hosts, mainly in aspects such as computation power, storage capacity, screen size and graphic resolution, autonomy, etc.

2.4 Adaptability in Mobile Computing

The variable supply of resources, as well as the differing demands on them, suggest that the client must adapt to these changes. However, this broad notion of adaptation requires to be defined. In what sense does a mobile system adapt? Which parties in the system are responsible for adaptation decisions?

Three models of adaptation are being discussed below:

1. **Application-Transparent Adaptation**: In the first model of adaptation, the system is wholly responsible for adapting to changes in the supply of and demand for resources. The system automatically handles

changes in connectivity between hosts, and transparently decides when to propagate updates or invalidate and re-fetch stale data. Individual applications have no say in how to make use of available bandwidth, though applications in either system can provide specific functionality, such as conflict resolution.

2. **Laissez-Faire Adaptation**: At the opposite end of the spectrum, applications are solely responsible for coping with the consequences of mobility. This approach, referred to as laissez-faire adaptation, has been taken by commercial software such as Eudora. In such systems, applications monitor the availability of resources, and make their own adaptation decisions in isolation of other applications or the system.

3. **Application-Aware Adaptation**: The middle ground between these two extremes is a collaborative effort between the system and the application [14]. The nature of this partnership is a consequence of end-to-end considerations. The system is best positioned to know what is available to the mobile client. Thus, it is responsible for monitoring resource availability, enforcing resource allocation decisions, and optimizing the use of client-wide resources. An individual application, however, is the only party which can know fully what its own needs are. Hence, an application must be informed by the system of significant changes in the availability of resources, and react to those changes in whatever way it sees fit. Application-aware adaptation is the only adaptation model that can support the sort of mobile computing.

3. DATA MANAGEMENT IN MOBILE COMPUTING SYSTEMS

A mobile computing system can be viewed as a dynamic type of distributed system where links between nodes in the network change dynamically. Figure 2 (adapted from Figure 3 in [4]) shows the classification of database systems in terms of four orthogonal axes, i.e., *autonomy* which refers to the distribution of control, not of data. It indicates the degree to which individual DBMSs can operate independently. Whereas autonomy refers to the distribution of control, the *distribution* dimension deals with data, more precisely the physical distribution of data over multiple sites. *Heterogeneity* may occur in various form in distributed systems, ranging from hardware heterogeneity and differences in networking protocols to variations in data managers. *Mobility* deals with clients that are issuing queries, servers that receive queries, or data targeted by a query.

3.1 Data Management Challenges in Mobile Environments

Mobile environment entities are treated as information repositories, we can describe this model as a type of mobile distributed databases. The system is highly autonomous since there is no centralized control on the individual databases that client maintain. It is also heterogeneous since it may include different types of hardware and different data types. Distribution is also another aspect of ad-hoc networks, where parts of data reside on different computers. Mobility is of course given, every entity can change its location and there is no requirement that some nodes must be fixed in a network.

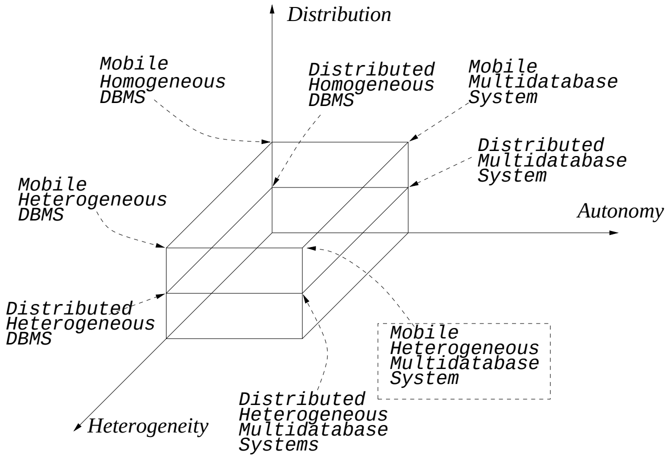

Figure 2: A Classification of Database Systems

There exists a considerable number of paper discussing general issues and research challenges related to mobility [8, 9]. Challenges engendered by mobility concern mainly : transactions, query processing, replication, caching and security. Mobile environment imposes the following issues that are primarily related to the randomness of every device's neighborhood at any instance of time. The neighborhoods consist of all reachable devices that a particular device can communicate with all available data that is accessible at that time. The main problems are [13]:

3.1.1 *Spatio-temporal variation in data source availability*

Entities in a mobile environment are moving and so their neighborhood. Hence, depending on the specific location and time a particular query is issued, the originator may obtain different answers or none at all. Furthermore, data sources availability may vary with location and time.

3.1.2 *Absence of a global catalog and schema*

As the neighborhood changes dynamically, a mobile device has no prior knowledge of the current set of available data. Some entities in the mobile environment have limited capability, and can not perform schema translation.

3.1.3 *Reconnection is not guaranteed*

When a device moves away from a current neighborhood it may affect any ongoing interaction among other devices of that neighborhood. An inconsistent global state may be caused by the uncertainty that the mobile devices will ever again be able to communicate among themselves.

3.1.4 *The query may be explicit or implicit*

In a ubiquitous environment, some interactions occur without explicit human intervention. But, in an ad-hoc environment, some entities are able to accept queries from humans and propagate them in the ad-hoc network.

3.1.5 Collaboration

In mobile environment entities interact in random manner which implies that privacy and trust issues should be taken into account. There are three main issues that must be considered. First, there may be an entity that has reliable information but refuses to make it available to others. Second, there may be an entity that is willing to share its information; however, that information may be unreliable. Lastly, when an entity makes information available to another entity, questions regarding protection of future changes and changing of that information arise.

4. CURRENT WORK ON 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). Figure 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.

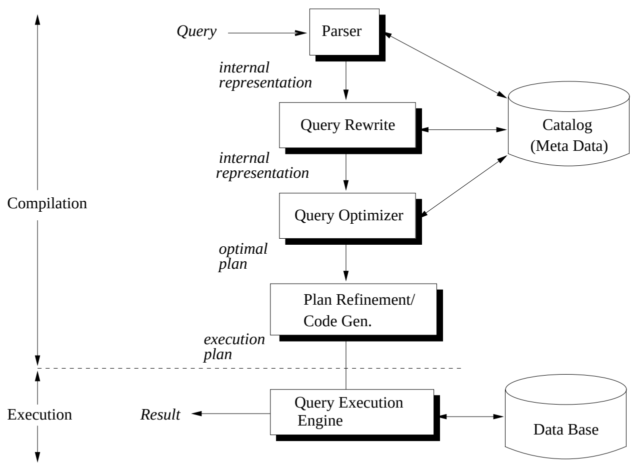

Figure 3: Phases of Query Processing

4.1 QoS Aware Distributed Query Processing

In a dynamic environment, several problems arise. Among them we can quote: congestion failures, unpredictable communication network, lack of knowledge about the load and potential delays at remote end systems, varying user's requirements and expectations from service provider.

One feasible way to capture these changes is to resort to QoS management [26]. Because QoS management aims at deciding and controlling if and how data streams can be delivered to the user within the given delay, cost or quality constraints, then the following two quality criteria are considered: (i) the cost of a service request , (ii) the total delay for obtaining the response.

To keep track of the current dynamic performance information about the underlying network, it is important for the optimizer to be customized to various environments and application requirements. Furthermore, revised or new cost models (particular communication cost) should be built. Two modules are added to the conventional query processor : one that provides different optimization criteria, the other provides the dynamic information for the underlying network. The architecture is based on the adoption of "user profile" generated from QoS requirements, QoS "profile mapping" module is used for this conversion. Finally, a QoS monitoring function is invoked during the optimization and the execution phase.

4.2 Adaptive Query Processing

In a distributed environment, data may be distributed over several site with different workloads. Thus, statistical information about the available data sources may be minimal and the query optimizer will not have at compile time necessary statistics, good selectivity estimates to produce an optimal query plan. Therefore, traditional optimizers cannot predict the future availability of resources. To face this problems, *Adaptive Query Processing* is introduced. In that case the query processor adapts itself to changing environmental conditions at runtime.

A query processor is considered to be adaptive if it receives information from its environment and determines its behavior according to that information. This action is repeated in an iterative manner, i.e., there is a feedback loop between the environment and the behavior of the query processor system [6]. Adaptive Query Processing (AQP) makes query processing more robust to optimizer mistakes, unknown statistics, and to changes in conditions over the running time of a query.

There are mainly six areas of interests where the adaptive query processing tries to adapt to:

1. Fluctuation in Memory: systems in this category try to adapt to memory shortages and to the availability of excess memory. Adapting to memory fluctuation can be done by (i) choosing operators like the hybrid hash join algorithm or memory adaptive sorting, (ii) scheduling the execution of a query, (iii) query re-optimization by changing the query plan.

2. User Preferences: adapting to user preferences includes cases where users are interested in obtaining some partial results of the query quickly. The user may also classify the elements of output in term of importance.

3. Data Arrival Rates: systems in this category adapt to data arrival rates in parallel and distributed systems, where the response times of remote data sources are less predictable. Parachute queries [3] can be used to adapt to data source failures. Query scrambling [20] and adaptive operators like Xjoin [19] and double pipelined hash join [24] are used to adapt to slow or bursty transfer rates. Eddies [1] is an example of an operator that adapts itself to continuous queries in data stream systems.

4. Actual Statistics: in some cases it is not possible to gather accurate statistics about the data sources. Thus, statistical information are collected at runtime in order to change the query plan and to adapt the query execution.

5. Fluctuation in Performance: performance can be affected when a node faces problems like high memory and CPU load, poor data layout on disks, or competing data streams.

6. Any changes in the Environment: some techniques can adapt to many kinds of changes in their environment by combining elements of other categories, i.e., to computer resources, like memory and processor availability, and to data characteristics, like operator costs, selectivities and data arrival rates.

4.3 Levels of Adaptability

The techniques of query evaluation can be classified into three categories: static query evaluation, personalized query evaluation and dynamic query evaluation. Thus, according to this classification , three levels of adaptability can be observed, namely : static, personalized and dynamic [21].

4.3.1 Static Adaptability

The first level of adaptability is observed during the design and the development phases of the query processor. Static adaptability mainly concerns the generation of query optimizers or extensible DBMS. This tool is used to construct new query optimizer using two methods: (i) by extending the search space where the optimization is based on rules that represent all possible manipulations of query plan, (ii) by extending search strategies, where a certain number of search strategies is proposed in order to develop a query processor.

4.3.2 Personalized Adaptability

After the development of the query processor, this one is used by different users with different needs and preferences. Personalization in DBMS appears principally at the level of data manipulation language. The aim of personalization is to adapt the choice of query execution strategy to the user's needs and preferences e.g., users may send a top N query or wants to have results as fast as possible.

4.3.3 Dynamic Adaptability

Query optimizer uses information collected from the environment in order to decide which query plan to be executed. However, these information may change at runtime and thus imply query optimizer errors. To face this problem, it is important that the query execution strategy may have the ability to adapt to environment changes. Dynamic adaptability aims to rectify incorrect estimations established at compilation time due to incorrect statistics or simplified cost metrics; it has been applied to long-running continuous queries.

5. ADAPTIVE QUERY PROCESSING IN MOBILE ENVIRONMENT

Consider an urban area with several vehicles where drivers and passengers in these vehicles are interested in information relevant to their trip [25]. For example, a driver would like his vehicle to continuously display on a map, at any time, the available parking spaces around the current location of the vehicle. The query processor uses GPS coordinates to send queries like "find the closest parking space". When the driver arrives to a parking space, GPS signals are less dependable in indoor environments such as brick and mortar retail spaces, our driver will look for "a closest parking place". The application looks for another positioning system like ISLANDS [18], APS [12] or RADAR [2] in order to fulfill the user's query. We assume that two vehicles can communicate with each other when their distance is smaller than a threshold. This communication can be enabled by a local area wireless protocol such as IEEE 802.11, Ultra Wide Band (UWB) or Bluetooth. With inter-vehicle communication, a mobile user discovers the desired information from the vehicles it encounters, or from distant vehicles by multihop transmission relayed by intermediate moving vehicles. When a vehicle leaves a parking slot, it informs other vehicles that its occupied place is henceforth free but only for a certain period of time. Drivers can ask queries like "find the closest free slot within 20 meters" or "find a nearby place where two slots are free" or "find a slot that will be free in the next 5 minutes".

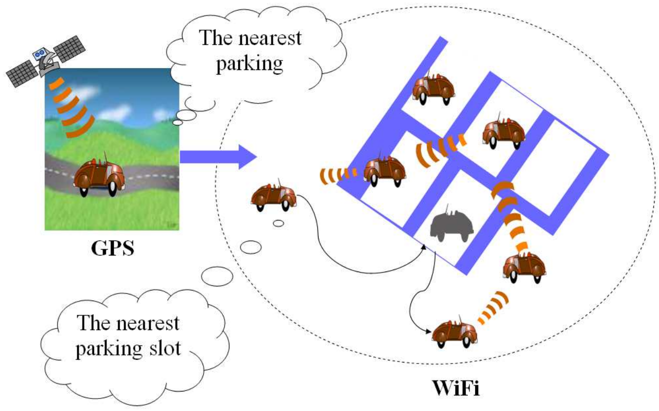

Figure 4: Park*Me* application

From the previous scenario, we identify several factors that affect query processing in a mobile environment, and are divided into three classes [5]: portability, mobility, and wireless communication.

5.1 The Impact of Mobility On Query Processing

The effects of mobility on query processing require that algorithms employed must be capable of managing frequent loss and appearance of mobile device in the network, and that overhead should be minimized during periods of low connectivity. In this environment we can distinguish many characteristics that are listed in the next sub sections.

5.1.1 Different Location Models

Location-based applications require a well-formed representation of spatial knowledge. Current location models can be classified into symbolic or geometric models. In geometric model locations are specified as an n-dimensional coordinate (longitude-latitude pair) or a set of coordinates defining an area's bounding geometric shape (such as a polygon). Symbolic models refer to a location by some abstract symbols. Such a representation allows a reference to a place simply by abstract symbol or name, which makes it very convenient for human interaction.

5.1.2 Different Query Types

The mobility in a mobile environment introduces three types of entities: (i) mobile client that submits a query, (ii) mobile server that processes a query or a part of it, and (iii) moving object which represents the data targeted by the query. According to these entities queries can be classified into four categories [10]:

1. ***Location Dependent Queries:*** in this category three types of queries are present [15]: firstly, if all the predicates and attributes used in a query are non location related then it is called a *Non Location Related Query* (NLRQ). For example: "select all restaurants with Italian specialty". Secondly, if a query has at least one Location Related simple predicate or one Location Related attribute then it is called *Location Aware Query* (LAQ). For example: "How is the weather in Lille?". Finally, if a query results depend on the location of the query issuer then the query is called *Location Dependent Query* (LDQ). For example: "Find me the closest hotel within 5 miles of my current position".

2. ***Moving Object Database Queries***: this type of queries includes all queries issued by mobile or fixed terminals and querying objects which are themselves moving. For example a query of this type could be : "Find all the cars within 50 feet of my car" [17, 16].

3. ***Spatio-Temporal Queries***: in a mobile environment, answers to user queries can vary with location [11]. That is, query results depend on a query's spatial properties. For a location-bound query, the query result must be both relevant to the query and valid for the bound location. The spatio-temporal type includes all queries combining space dimension with time dimension which generally associated to moving objects. The time notion introduces two sort of spatio-temporal queries: the first one considers trajectories describing a time history of the object movement. The second one focuses on the current position of the moving object and possibly its future position.

4. ***Continuous Queries (CQ)***: continuous queries are another type of queries that allow users to receive new results when they become available. For example a driver is asking for gas stations within 10 miles from its position. The result of the query is a set of gas stations that varies *continuously* with the movement of the driver [17].

5.1.3 Query Optimization

Query optimization methods try in general to obtain execution plans which minimize CPU, input/output and communication costs. In centralized environments the cost that affects most is the input/output whereas in distributed environments, communication cost is the most important. In a mobile distributed environment, the communication costs are much more difficult to estimate because the mobile host may be situated in different locations. The best site from which to access data depends on where the mobile computer is located. In general, it is not worth calculating plans and their associated costs statically, but rather, dynamic optimization strategies are required in this mobile distributed context.

5.1.4 Query Execution

In static systems, query processing execution sites are determined in advance, i.e., which steps are performed on the client and which one on the server. In a mobile environment, where users are moving, such assumption is inadequate. Thus, mobile database systems must be able to choose an execution site for the different phases of query processing depending on their current environment and should be able to revise that decision as flexible as possible.

5.2 The Impact of Portable Devices Limitations

If we reference dynamic location information in a query, we have to use a location management component to get this information. Thus, depending on the offered localization strategy, we have different possibilities to use this information. The cost evaluation of a query execution plan is guided by required resources of the plan. The main factors that are used in stationary systems are CPU-usage and the number of hard disk access. In mobile systems, additional varying factors like energy consumption, available memory and CPU-speed may be included.

5.3 The Impact of Wireless Communication

The new networking technologies allow spontaneous connectivity among mobile devices, including hand helds, computers in vehicles, computers embedded in the physical infrastructure, and (nano)sensors. Mobile devices can suddenly become both sources and consumers of information. There is no longer a clean distinction between clients and servers, instead devices are now peers. Furthermore, there is no longer a guarantee of infrastructure support. Consequently, for obtaining data, devices cannot simply depend on a help of some fixed, centralized server. Instead, the devices must be able to cooperate with others in their proximity in order to pursue individual and collective tasks.

5.4 The Need for Adaptability

We listed above the main obstacles that a query processor can face in a mobile environment. More specifically, in an ad hoc environment mobile devices are highly volatile, meaning that the time of a connection in the network is not known. Furthermore, as mobile devices are moving and as new data can arrive at any moment, there is no guarantee about the type of information available at any given time and space and the answer to a query issued by a mobile user could be sent to a location that is already left. To support distributed query processing in mobile network, disconnection, bandwidth and reliability must be considered and query optimization must take these characteristics into account and adopt appropriate optimization strategies based on the status of the network.

Resource limitations create asymmetry between mobile elements. In [23] three feature of asymmetry were identified, namely: asymmetry feature of computing capability between the server and mobile computer, asymmetry feature of energy consumption between message sending and receiving, and asymmetry feature of energy consumption between activeness and idleness of mobile computer. [23] has examined three different join methods and developed three query processing schemas. These features can be easily identified in a mobile ad hoc network, where mobile devices can play the role of client and server. Thus, the query processor should be able to choose the adequate schema in order to cope with energy problems.

Mobility of users affects mostly the network topology and the location of mobile devices that can be seen as a frequently changing data. In this case query processor may have to treat continuous data (in our case location) and continuous queries that may include a temporal dimension. For example a stationary driver may issue a query and wants to locate the closest free parking slot, then in the course of his driving he may issue a query and asks for the closest free parking slot. Thus, the query processor will have to treat first a snapshot location dependent query (in stationary mode) then a continuous location dependent query (in mobile mode). In order to cope with the frequently changing location, query processor should have a monitoring function that detects the mobility of the user and then informs the query processor to dynamically change the query evaluation mode.

Location aware applications use location sensing systems like GPS to locate objects in mobile computing. These location sensing systems have some limitations e.g., GPS can not be used indoors, Active badges are affected by sunlight and fluorescent light which interfere with infrared (for more details we refer readers to [7]). Thus, the query processor should have the ability to detect other location sensing systems and then toggle to the one that best meets query evaluation needs.

Traditional query evaluation techniques generally depend on the application. They are optimized to deliver a complete answer and do not hold take into account the user preferences. However, these preferences are important in a constrained environment. For example, in an ad hoc application, a user may wish to receive quickly the results of his request even if they are incomplete. On the other hand, in a B2B environment, the user may wish to receive complete results and as soon as possible.

6. CONCLUSION

Mobile computing is the future vision of the world. In such an environment, many challenges have arisen but generally, the proposed solutions only try to treat a particular problem. For example, [13] presented a system framework for query processing in ad hoc wireless networks which only supports simple queries, and does not take query optimization problems into consideration. As information will be accessed from anywhere, at any time, and from any terminal, the need of an adaptable query processing, allowing a more effective evaluation, has appeared. Furthermore, this query processor should have a flexible architecture that can adapt to device capabilities and heterogeneity. In this article we defined the main constraints that a query processor can face in a mobile environment and presented the needs for adaptability when deployed in a mobile constrained and distributed environment.

7. ACKNOWLEDGEMENT

This work is done as part of the MOSAIQUES project supported by FEDER and the region of Nord-Pas-de-Calais.

8. REFERENCES

[1] R. Avnur and J. M. Hellerstein. Eddies: continuously adaptive query processing. *SIGMOD Rec.*, 29(2):261–272, 2000.

[2] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, pages 775–784, 2000.

[3] P. Bonnet and A. Tomasic. Parachute queries in the presence of unavailable data sources. Technical Report RR-3429, INRIA Rocquencourt, France, 1998.

[4] M. H. Dunham and A. Hellal. Mobile computing and databases: Anything new? *SIGMOD Record*, 24(4), December 1995.

[5] G. H. Forman and J. Zahorjan. The challenges of mobile computing. *Computer*, 27(4):38–47, 1994.

[6] J. M. Hellerstein, M. J. Franklin, S. Chandrasekaran, A. Deshpande, K. Hildrum, S. Madden, V. Raman, and M. A. Shah. Adaptive query processing: Technology in evolution. *IEEE Data Engineering Bulletin*, pages 23(2):7–18, 2000.

[7] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.

[8] T. Imielinski and B. R. Badrinath. Data management for mobile computing. *SIGMOD Rec.*, 22(1):34–39, 1993.

[9] T. Imielinski and B. R. Badrinath. Mobile wireless computing: challenges in data management. *Commun. ACM*, 37(10):18–28, 1994.

[10] N. Marsit, A. Hameurlain, and F. M. Z. Mammeri. Query processing in mobile environments: A survey and open problems. *dfma*, pages 150–157, 2005.

[11] M. F. Mokbel, W. G. Aref, S. E. Hambrusch, and S. Prabhakar. Sina: Scalable incremental processing of continuous queries in spatio-temporal databases. *SIGMOD*, 2004.

[12] D. Niculescu and B. Nath. Ad hoc Positioning System (APS). In *GLOBECOM, San Antonio*, November 2001.

[13] F. Perich, S. Avancha, A. Josh, Y. Yesha, and K. Joshi. Query routing and processing in mobile ad-hoc environments. Technical report, UMBC, November 2001.

[14] M. Satyanarayanan, B. Noble, P. Kumar, and M. Price. Application-aware adaptation for mobile computing. In *EW 6: Proceedings of the 6th workshop on ACM SIGOPS European workshop*, pages 1–4, New York, NY, USA, 1994. ACM Press.

[15] A. Y. Seydim, M. H. Dunham, and V. Kumar. Location dependent query processing. In *MobiDE*, pages 47–53, 2001.

[16] A. P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao. Modeling and querying moving objects. In *ICDE*, pages 422–432, 1997.

[17] A. P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao. Querying the uncertain position of moving objects. *Lecture Notes in Computer Science*, 1399:310–337, 1998.

[18] M. Thilliez and T. Delot. A localization service for mobile users in peer-to-peer environments. In F. Crestani, M. D. Dunlop, and S. Mizzaro, editors, *Mobile HCI Workshop on Mobile and Ubiquitous Information Access*, volume 2954 of *Lecture Notes in Computer Science*, pages 271–282. Springer, 2003.

[19] T. Urhan and M. J. Franklin. XJoin: Getting fast answers from slow and bursty networks. Technical Report CS-TR-3994, University of Maryland, February 1999.

[20] T. Urhan, M. J. Franklin, and L. Amsaleg. Cost-based query scrambling for initial delays. *SIGMOD Rec.*, 27(2):130–141, 1998.

[21] T.-T. VU. *Une approche pour la construction d'valuateurs adaptables de requtes*. PhD thesis, Institut National Polythechnique de Grenoble, 2005.

[22] M. Weiser. The computer for the 21st century. *Scientific American*, pages 94–104, 1991.

[23] M.-S. C. Wen-Chih Peng. Query processing in a mobile computing environment: Exploiting the features of asymmetry. *IEEE Transactions on Knowledge and Data Engineering*, 17:982–996, July 2005.

[24] A. N. Wilschut and P. M. G. Apers. Dataflow query execution in a parallel main-memory environment. In *PDIS '91: Proceedings of the first international conference on Parallel and distributed information systems*, pages 68–77, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.

[25] B. Xu, A. Ouksel, and O. Wolfson. Opportunistic resource exchange in inter-vehicle ad-hoc networks. In *IEEE Int. Conf. Mobile Data Management*, Berkeley, CA, Jan. 2004.

[26] H. Ye, B. Kerherve, and G. von Bochmann. Qos aware distributed query processing. In *DEXA Workshop*, pages 923–927, 1999.