Chapter 1

Introduction

This work has as academic framework the research group HADAS pertaining to the Grenoble Informatics Laboratory (LIG), and the Data and Knowledge Management group pertaining to the Group of information and communication technologies (G-TIC) at Universidad de las Americas, Puebla.

1.1 Context and motivation

We live in an increasingly digitalized and connected world. In order to carry out several tasks of our daily life, such as talk, shop, read and work, we use diverse computational applications and electronic devices integrated into the environment in almost any type of object imaginable, including cars, tools, appliances, clothing and various consumer goods. All of them communicated through interconnected networks.

Most of these activities involve the generation, exchange and sharing of information. It is possible to load and share our music and photo libraries using the services provided by different websites, our answering machines are virtual mailboxes, our reference libraries are online and what is more, businesses and sciences today use computational and electronic devices to support their activities. We spend hours every day in this

digital world, so it is not surprising that the amount of data we access is growing in exponential and unstoppable way. The data generated end up in databases, log files, among others.

In many ways, this description introduces an ubiquitous computing environment, this paradigm is also described as pervasive computing, ambient intelligence, or more recently, everywhere [12]. Different kinds of flavors of ubiquitous systems and applications can be described.At their core, all ubiquitous computing environments share a vision of small, expensive, robust mobile or stationary computational (software applications, web services, calculation tool, database systems, etc.) and electronic devises (sensor networks, PDAs, cell phones, etc.), distributed at all scales throughout everyday life and generally turned to distinctly common-place ends.As special characteristic, users are not aware of the devices to which they interact [27].

We can find this type of environments everywhere e.g. transport, enterprises, mobile environments, services, at home, among others. The main functionality of ubiquitous environments is to provide any information, at any time from any place using any device. In order to provide this functionality, query evaluation techniques are required. Query optimization is essential to improve query evaluation efficiency. Ubiquitous environments present challenges across computer science. Request large volumes of data located everywhere, and which in addition are highly dynamic and heterogeneous remains a huge challenge. This implies the improvement of query evaluation and query optimization techniques. This thesis project is focused on query optimization [5][27].

Query optimization is a widely studied problem, a wide variety of query optimization techniques have been suggested e.g. semantic, parametric and query optimization via probing queries [29]. Even though these approaches allow the efficient query processing they are presented in the framework of classical query evaluation procedures that rely upon cost models heavily dependent of metadata (e.g. statistics and cardi-

nality estimates) and that typically are restricted to execution time estimation. These characteristics hamper the efficient information access [17].

There exist different computational environments where no metadata are available; in addition, execution time is not the only optimization objective. An ubiquitous computing environment is a good example of this type of environments, where classical query optimization techniques are not useful any more [13]. Since devices are autonomous, dynamic and have limited physical characteristics (e.g. energy or CPU power), the evaluation time of queries is no longer the main optimization objective. Moreover metadata required for a priori estimating the evaluation cost of query plans are not always available [18] [16].

1.2 Problem definition

Properties that characterize resources in ubiquitous environment represent new challenges for the evaluation of queries, and of course, also for their optimization. Some of the challenges that query optimization in ubiquitous environments involves are related to the variable execution context as a result of their dynamicity and autonomy properties. Others refer to the absence of statistics and metadata used to determine the cost of an query plan (possible solution for given query) by means of an adaptive cost function that optimize a query in terms of different parameters according to the user requirements. Physical constrains represent another significant challenge. Classical query optimization techniques are not capable to deal with the challenges that resource characteristics in ubiquitous environments imply.

In general, classical query optimization declines in two main aspects, the exploration of a search space (set of alternative query plans that solve a query) and the evaluation of each query plan in the search space in order to select the optimal [22]. The exploration

of the search space involves the aspects related to query plan generation [17]. The evaluation of the query plan involves the definition of a cost model in order to estimate the cost of each query plan in the search space according to an specific optimization objective. This cost model is tightly tied to the use of metadata.

In ubiquitous computing environments, where devices are autonomous and have limited physical characteristics (e.g. energy), the evaluation time of queries is no longer the main objective of optimization. Furthermore, classical query optimization techniques based the execution plan evaluation on metadata that is not available [25]. This thesis project addresses the query optimization problem in execution environments, such as ubiquitous computing environments, that lack of metadata and where is necessary to optimize a query according to a wide variety of parameters in order to satisfy user requirements.

In relation with the two main aspects involved by classical query optimization techniques, this work is mainly focused in the problem that the evaluation of query plans imply when metadata are not available. Nevertheless, the search space exploration techniques must be analyzed in order to determine if is possible to suggest a more appropriate query plan generation technique according to the computational problem addressed by this thesis. This query optimization problem declines in three main subproblems: the acquisition or generation of new knowledge rather than metadata, the management of this knowledge, and finally, its exploitation.

Since the cost of execution query plans cannot be evaluated a priori due to the absence of metadata to perform the appropriate calculus operations. It is needed to search alternative sources of knowledge (knowledge acquisition or generation) in order have some measures as points of reference about the query performance and the computational resources consumed during its execution. The knowledge associated to already solved queries that are similar to the new one can be a convenient source. A model for

the representation of this knowledge must be proposed.

Manage this knowledge involves the specification of data structures for storing the knowledge and techniques for insert new knowledge, update the existent and eliminate the obsolete, this means, not useful to solve new queries. Finally, knowledge exploitation involves search strategies to find the useful knowledge for solving a new query; this knowledge is related to similar queries that were previously evaluated. Define a search strategy to knowledge recovery as well as the identification of similarity between queries is an important computational problem.

Query optimization is still an open research problem. New optimization techniques must be developed. These techniques must allow to the user customize the optimization objective according to his requirements. It is necessary to deal with metadata absence; useful knowledge must be obtained from previously executed queries and be managed and exploited by means of automatic learning techniques, their goal is to improve or acquire new capabilities from experience related to some specific activities. We propose an approach for query optimization in ubiquitous computing environments based on machine learning in order to address the problem related to lacking of metadata [Fra01].

1.3 Objectives

The general objective of this thesis project is to propose a query optimization technique for ubiquitous computing environments based on knowledge obtained from previously executed queries. Also this technique must not be restricted to a single optimization objective; it must be personalized according to user requirements. This objective declines in the following specific objectives:

1. **Definition of a query optimization technique based on learning.** A learning process must be adapted to query optimization in order to take advantage of

all the knowledge that is acquired from the queries that are similar to the new one and that have been already solved. This implies the creation of methods for storing, managing and exploiting this knowledge. Knowledge exploitation involves the specification of useful knowledge by applying similarity functions between queries.

2. **Definition of a customizable query optimization technique.** Typically, classical query optimization techniques propose as optimization objective the execution time of a query. However, exist other optimization techniques that propose another optimization parameters e.g. memory, CPU or energy consumption, but this is constant, is to say, any query is optimized according to the same optimization objective. This thesis has as objective, to propose a query optimization technique with an optimization objective that can be personalized according to user requirements.

3. **Definition of a query plan generation strategy.** The search spaces can be enormous, in order to explore them, search strategies based on enumerative, dynamic and genetic algorithms have been proposed. The first two strategies consider all the execution plans in a search space, the last one not offer any guaranty about the optimality of the result. It is convenient to propose a new query plan generation technique appropriate to the computational problem addressed in this thesis.

4. **Prototype implementation.** Finally, is needed to construc a query optimizer that implements the query optimization techniques based on machine learning approaches, as well as the specified query plan generation techniques must be carried out. ?

1.4 Approach and main results

In this thesis project we propose a new query optimization technique based on learning approaches, specifically case-based reasoning. This technique deals with the challenges that context like ubiquitous computing environments represent for existent optimization techniques, particularly the lack of metadata. Different from most classical query optimization techniques, our approach suggest a posteriori estimation cost based on measures (e.g. consumed memory and energy, execution time, etc.) performed during previous executions of queries. Moreover, the optimization technique that we propose is not another cost function-based query optimization technique; it allows the cost model customization for better adapting to the client application requirements.

1.4.1 Machine learning

Machine learning is an artificial intelligence subdiscipline concerned whit design and development of methods that allow computers to learn in an automatic fashion in order to improve or create specific capabilities [2].

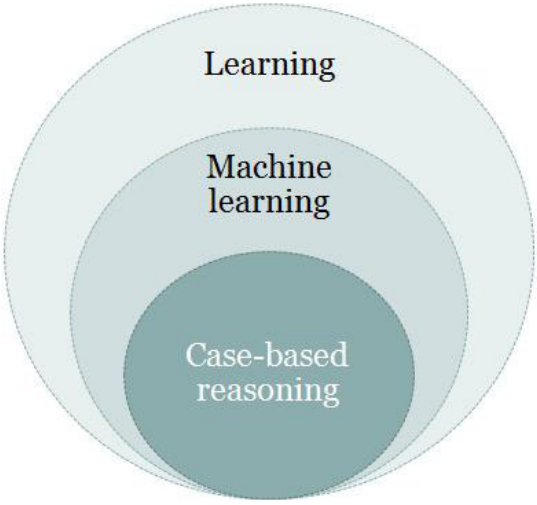

Figure 1.1: Machine learning: Case-based reasoning approach

Typically, these methods are based in information obtained as a result of experiences related to problem solving and execution environment behavior. Machine learning has a wide range of applications, including search engines, medical diagnosis, fraud detection, speech recognition, games, robotic among others. There are different automatic learning methods, we centered our interest in case based reasoning [6].

Case based reasoning is a machine learning approach (Figure 1.1) that consists of a process to solve new problems based on experience of similar problems already solved. A case is the minimum unit of reasoning and consist of a problem, its solution, and, typically, annotations about how the solution was derived. Is storage in a case base. Case based reasoning has been formalized for purposes of computer reasoning as a four-step process [23] [6]:

1. **Retrieve.** Given a new problem, cases relevant to solving it must to be retrieved.

2. **Reuse.** Map the solution from the previous case to the new problem. This may involve adapting the solution as needed to fit the new situation.

3. **Revise.** Having mapped the previous solution to the target situation, test the new solution in the real world (or a simulation) and, if necessary, revise.

4. **Retain.** After the solution has been successfully adapted to the target problem, store the resulting experience as a new case in memory.

1.4.2 Query optimization using learning approaches

The optimization technique based on learning considers a query case base where cases represent experiences of queries that have been previously executed, optimized and evaluated. It address the exploitation of the case base for generating execution plans and learning on existing ones according to measures inserted in the case base.

Given a query, the knowledge acquired from previous experiences is exploited in order to propose reasonable solutions. It is possible to learn from each new experience in order to suggest better solutions to solve future queries. On the other hand, there may be no useful knowledge to solve the query, in this case we suggest generate a pseudo-random query plan to solve the query. We said that is a pseudo-random generation since it applies classical and well known heuristics for query plan construction but also takes some random decisions.

It is important to mention that we based our study and our solution in the optimization of queries specified in a declarative language, as is well known, by means of this type of languages is possible to specify what must be done but not how is done. The internal execution order of a sentence can seriously affect its evaluation. An optimization process must be carried out in order to improve the evaluation of a query.

The reasoning process that must be accomplished given a new query is an adaptation of the four-step case-based reasoning process to query optimization. These steps are retrieving, readapting, reviewing and retaining. Figure 1.2 illustrates this reasoning process.

1. **Retrieving**, this step is based on a similarity function in order to perform a smart search to retrieve the most relevant cases to solve the query of the problem. Among these relevant cases, the one that minimizes the cost function of the problem is selected.

2. **Reusing**, this step is related to the adaptation process of the execution plan included in the relevant case to the new query.

3. **Reviewing**, it consists in verifying the query by its execution during which resource consumption measures are done.

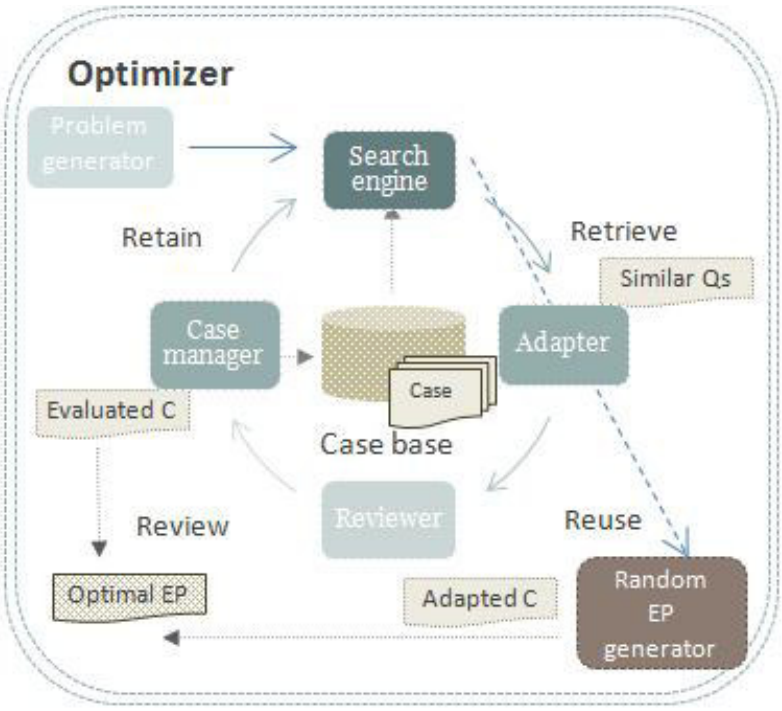

Figure 1.2: Case-based reasoning process

4. **Retaining** step is in charge of the storing of the problem and in the case base in form of a new case.

Similarity notion is a very important concept because is the basis for the retrieving step, also is significant for readapting and retaining steps. The similarity function determines which query related to cases stored in the case base are similar and useful to optimize a new query, which is the problem to be solved. The adaptation depends on the similarity level that is determined between the two queries. Basically, the adaptation process consists in evaluating the operations of the query with variations in their conditions. Finally, cases in the case base classified according to their similarity [21] [4].

In order to validate this optimization technique we construct QuOS, a query optimizer that implements query optimization techniques based on machine learning ap-

proaches, particularly case-based reasoning. It is composed by two main modules, the case-based reasoner and the execution-plan generator. The case-base reasoner is in charge to adapt the solutions of similar queries to the new situation. The execution plan generator is in charge to generate new execution plans in a pseudo-aleatory way. The case-base reasoner is the most complex of the two modules but the smartest, on the other hand, the execution plan generator is simpler and probably faster but it does not apply case-based reasoning techniques.

1.5 Document organization

This thesis is composed of seven chapters including this introduction. Chapters two and three define the theoretical framework of this thesis that is divided in two main computational disciplines, the first one addresses query evaluation and optimization issues, the second one is related to machine learning approaches, particularly, case based reasoning. Chapter four describes the query optimization technique using case-based reasoning that is the core of this thesis. Chapter five detail the characteristics of SQuO, the query optimizer that implements the query optimization technique proposed in this thesis. Chapter six presents the validation of the prototype SQuO. Finally, Chapter seven presents the conclusions and perspectives of our work. An overview of each chapter is detailed below.

Chapter 2. In principle, this chapter describes some query evaluation process details in order to remark the importance of query optimization within this process. Then, describes the main query optimization approaches, as well as different optimizer architectures and execution environments.

Chapter 3. This chapter presents an overview about machine learning approaches, particularly, case-based reasoning. It address the general knowledge representation by mans of cases in a case based, as well as the entire reasoning process that must be accomplish according to this approach.

Chapter 4. This chapter describes the query optimization technique proposed in this thesis project. It is a query optimization technique that uses case-based reasoning and aims the query optimization in execution environments where no metadata is available. In addition, it allows the optimization objective personalization according to the user application requirements. This includes the adaptation of the case-based reasoning approach to query optimization; an important aspect of this adaptation is the proposition of a similarity function between queries. Also, it includes a pseudo-random query plan generation technique in order to feed the case-based with some first knowledge.

Chapter 5. This chapter presents the system SQuO which is a query optimizer based in the query optimization technique using case-based reasoning. First describes its general architecture. Immediately, describes the data structures for knowledge representation employed by the system. Finally, details its main functions.

Chapter 6. This chapter describes the experimental validation of our proposal. In general, the experimental validation consisted in the evaluation of several queries given different knowledge that are pre-loaded to the case-based. The principal objective is to study the behavior of our prototype, as well as to compare the learning process when different knowledge is pre-loaded to the case base.

Chapter 7. This chapter presents our conclusions and the perspectives of the im-

provements and extensions that could be realized over the presented prototype. Also, present interesting alternative applications of the proposed query optimization technique.