Statement of Present and Future Research Interests

Zhiyuan Chen

Current Work

My research interests lie in the area of database systems. In my dissertation research, I have explored new directions of improving performance of database systems given current trends of hardware development and the requirement of new applications. I have addressed the following three issues: database compression, result size estimation for queries on string and hierarchical data (e.g., XML and LDAP), and automatic physical database design.

Database Compression

Over the last decades, improvements in CPU speed have outpaced improvements in main memory and disk access rates by orders of magnitude, enabling the use of data compression techniques to trade reduced I/O and storage space against additional CPU overhead for compression and decompression of data. The goal of my research is to build *compressed database systems* that can store and query compressed data and achieve better query performance than conventional database systems. Considering today's storage prices, I mainly target improvement of query performance rather than saving of storage space, which is the goal of traditional data compression. Therefore, the main technical challenge here is to balance the benefits (I/O savings) and overheads of compression. I addressed the following two issues:

- Selection of appropriate compression methods. Since my goal is improvement of performance rather than saving of space, there are several novel issues in choosing suitable compression methods. First, compression and decompression are CPU intensive operations and their cost is important in choosing compression methods. Second, database applications often need random access to data in small pieces (tuples, attribute, etc.). This makes some popular compression methods such as Lempel-Ziv inappropriate because they have to decompress a large chunk of data. Third, data stored in databases is well structured and values of the same attribute usually share more similarities. Therefore, I have proposed strategies that intelligently select the most appropriate combination of compression methods to balance the benefits and overheads of compression.

- Query optimization in compressed database systems. Query optimization is crucial in compressed database systems because during query execution, data can be either eagerly decompressed when it is read into main memory, or lazily stay compressed in main memory and be decompressed on demand, and there are cases one strategy leads to orders of magnitude better performance than the other strategy. I have addressed the compression-aware query optimization issue and proposed two algorithms: one is provably optimal and the other is very fast and almost always generates optimal execution strategies. All the above work has been implemented in a full fledge compressed database system, which achieves up to an order performance speedup over conventional database systems.

Result Size Estimation for Queries on String and Hierarchical Data

String and hierarchical data such as XML and LDAP have become ubiquitous. Estimating the number of data items (tuples, documents, etc.) matching a query on string and hierarchical data is useful to optimize the execution of the query and to provide a quick feedback for query refinement. However, traditional estimation techniques such as histograms are not sufficient because of two reasons: (1) they only work for numerical data, and (2) they can only estimate result sizes of simple range queries, while we are interested in estimating result sizes of complex queries with multiple conditions (e.g., find web pages containing "CS Department" and "Opening"). I proposed a solution that uses a compact summary structure to store the result sizes for pieces of queries those are frequent in the data. This approach relies on two techniques: (1) a Set Hashing technique used in the summary structure to succinctly capture correlations in the data, and (2) algorithms based on conditional probability to reconstruct the result size of the original query from the result sizes of pieces of that query stored in the summary data structure. Experimental results demonstrate that our approach can achieve accurate and robust estimates (typically the estimation is within a factor of 0.5 to 2 of the real value) using limited space (typically less than 1% of the data size).

Automatic Physical Database Design

The automation of physical database design such as selecting indexes and materialized views has become of increasing importance because not only physical design is crucial for database performance, but also manual physical design is very costly (e.g., the salary of a database administrator can often buy a fleet of decent computers). Physical database design problems typically involve hard optimization problems and current approaches rely on often ad hoc heuristics to find good solutions. I proposed a new approach that starts with an empirical study of the structure of the search space for a particular physical design problem to identify the generic "shape" of the space. Then insights can be drawn from the shape of the search space and allow us to design much improved algorithms to find good solutions. I have applied this approach to two physical design problems: (1) the problem to select indexes, and (2) the problem to decide what attributes to compress in a compressed database system given a query workload. As a result, I have proposed index selection and compressed database design algorithms that achieve several orders speedup over existing algorithms while achieving results of the same good quality.

Future Work

After my graduation, I would like to continue working on three topics related to my current research: (1) efficient handling of new types of data in emerging applications, (2) building the next generation of automatic physical database design tools, (3) improving the performance of main memory database systems.

Efficient Handling of New Types of Data in Emerging Applications Conventional database systems are good at handling numerical data, but are not good at handling new types of data (e.g., documents, hierarchical data such as XML, and biology data) in emerging applications. Several challenging research issues need to be addressed. First, although redundant structures such as indexes and materialized views have been widely used in conventional database systems, little has been done on the redundant structures to speed up queries in new applications. Such redundant structures are not only crucial for performance, but also tinvolve novel research issues because of the complexity of new types of data over structured numerical data. Second, these new applications often require new types of operations and hence new processing techniques. For example, data cleaning and integration are two important operations in E-Commerce and it remains open on how to execute them efficiently and effectively. Third, given the vast volume of data available (typically generated by programs), it is often sufficient to get quick and approximate answers of a query. However, little is done on approximate query answering for new types of data. I believe progress on these three issues would have valuable contribution to research community and industry.

Building the Next Generation of Automatic Physical Database Design Tools Current automatic physical database design tools have several shortcomings. First, although query workloads typically change over time, these tools design physical databases in a static way. Hence, I would like to design tools that can monitor performance of database systems and dynamically suggest necessary changes of design. Second, although different design options often interact, current design tools usually decided them individually. For example, the decision on how to partition the data and how to index the data have strong correlations, however, current design tools decide them separately. Hence, I plan to design algorithms that consider physical design options in a joint and efficient way.

Improving the Performance of Main Memory Database Systems Over the last decades, the price of memory has dropped quickly and today many applications can fit their databases in main memory. However, the improvements on CPU speed have outpaced the improvements on memory access speed by orders of magnitude over the last decades. Hence, it is crucial to optimize the cache performance for main memory database systems and many research problems are still open. For example, most existing research deals with numerical data, and it is unclear how to improve cache performance for operations on string data. Also, compression is potentially useful to trade improved cache performance against additional CPU overhead to compress and decompress the data. However, the gap between memory access speed and CPU speed is much smaller than the gap between disk access speed and CPU speed and hence we may need compression methods and query optimization techniques different from what I proposed for disk based systems.

I am very excited about these three research directions because they both have potential impact and involve challenging technical and practical issues. For example, building new physical design tools both has immediate commercial impact and involves solving challenging optimization problems.