Research Statement

Jennifer L. Beckmann
jbeckmann@cs.wisc.edu

My research focuses on data management for next-generation applications, with an emphasis on performance, query processing, and manageability of database systems. I see many challenges in information management as more applications use databases to organize personal information, to manage scientific data, and to manage file systems. In my dissertation, I focus on applications that need efficient support for heterogeneous data integration, specifically data that is heterogeneous and sparse. I address the shortcomings of current systems by developing solutions with theoretical performance models, by validating my solutions in real systems, and by running experiments with real world data.

Heterogeneous data integration is one area that databases need to provide more efficient and seamless solutions for applications. In data integration, schema mapping tools provide ways to combine data from different sources; however, those mappings can result in performance issues for relational systems. Sparse data sets, which are common in emerging applications such as e-commerce product catalogs, arise from integrated data sources each of which has a slightly different way to describe its data. My dissertation focuses on the performance of relational systems for sparse data sets. In applying relational systems to sparse data, I address issues of storage efficiency, index creation, and statistics collection.

Application of Database Systems to Sparse Data

Data management for e-commerce product catalogs is one such next-generation application. E-commerce product catalogs gather product specifications, or lists of properties, from multiple vendors into a single website. Consumers use the specifications to comparison shop. For example, a product specification for the Apple iPod Nano lists its *weight* as *1.5 oz* along with 55 other attributes and values describing the player. The catalogs have limited uses, however, because the typical query interface is only keyword-based and keyword queries are not expressive enough to select products based on a predicate. For instance, it is impossible to specify a keyword query looking for an mp3 player that weighs less than 2 oz. Relational databases are a natural fit for storing and querying product specifications because they associate attributes and values, and provide an intuitive way to query data. However, the data is highly unstructured and storing it in relations results in extremely *sparse tables* where each object (product) only defines a handful of the hundreds of possible attributes. Relational systems are ineffective at handling sparse data in three ways: table storage is inefficient for null values; limited indexing restricts query optimization possibilities; and finally, the number of attributes in the data makes it difficult for optimizers to identify important statistics.

In the first part of my thesis, I identify that current systems are inefficient at handling sparse data because record storage formats waste space and the added space creates high I/O overheads. Systems that do not store sparse data efficiently make it hard to fully integrate sparse data with classic relational data. Typical relational tables use *horizontal* schema, with the columns of the table comprising all the available attributes. A problem with horizontal schema for sparse data is that conventional record storage formats are inefficient for storing null values and, because of the inefficiency, users are forced to use an ad-hoc *vertical* schema that splits a single row in a horizontal table into multiple rows. Vertical schema, nevertheless, can also be inefficient, especially when a query includes many attributes, and vertical schema are a mismatch when queried with horizontal schema.

My work proposes and evaluates a novel technique for storing sparse data in a tuple format that does not allocate space to null values. Our techniques show a significant speedup over traditional relational storage for sparse data stored in either a horizontal schema or vertical schema [ICDE 2006]. The work is significant because it enables relational systems to support an important and growing class of data in a seamless manner. We developed our solution by constructing a theoretical framework for comparing queries over horizontal and vertical schema. Through a performance model, we noticed that the traditional RDBMS storage format slowed the performance of queries over horizontal schema and that our storage format would significantly improve performance for data stored horizontally or vertically. We showed the benefits of the technique by implementing the ideas in the open source database PostgreSQL and evaluating them with synthetic and real word e-commerce data that we collected from the Internet.

The second part of my thesis addresses the problem of providing index access paths to all attributes in sparse data. The availability of indexes over a data set is important for query efficiency, because indexes allow direct access to the rows based of a relation on column values. In the absence of an index, the only way to execute a selection query is to scan the entire table. One seldom builds hundreds of indexes on a table for dense data, because the maintenance of the indexes slows the time to insert, delete and update rows of the table. In sparse data, however, where a table may have hundreds of attributes, indexing only a few attributes in a table means that a table scan is the only evaluation plan for almost all selection queries on that table.

My work uses indexes that do not maintain information about the null values in the data, which we call sparse indexes, to provide index access to all attributes while keeping index maintenance costs low. In suggesting the use of sparse indexes, we also recognize that sparse indexes have limited capability for queries that search for rows with undefined values (is-null queries) and for queries that search for present values (is-not-null queries) over text data types. We propose “is-present” indexes that identify the rows that are non-null for text attributes and that optimizers include support for index differencing plans within query optimizers. Results from our work show that query plans using our “is-present” indexes and plans that use set difference offer speedups of over 12 times that of a table scan. We developed key ideas and implemented a test-bed for evaluating our ideas using Microsoft SQLServer.

In the final aspect of my thesis, I consider statistics over sparse data sets. By using real-world e-commerce product catalogs, I characterize the structure of the data and, how knowledge of this structure can help query optimizers. Sparse data results from storing many different, but somewhat related, objects. For example, a catalog provider probably suggests a set of attributes for a group of products, such as mp3 players, and vendors should use those attributes. However, vendors may elect not to define certain attributes, or choose some attributes from other product groups to extend a product schema (e.g., cell phones that play mp3s). The flexibility of vendors to add or remove attributes results in strong correlations among attributes in the same product grouping. By knowing the distributions in a sparse data set, query optimizers can better estimate the selectivity of conjunctive predicates and make better query optimization decisions. Because keeping multi-column statistics on all pairs of hundreds of attributes is unreasonable, the challenges are in defining necessary statistics, discovering those statistics, and storing statistics for optimizers. As work in progress, I have defined single-column statistics for measuring sparsity and am considering techniques from machine learning co-clustering to discover more intricate attribute dependencies.

Industrial Research Lab Internships

My summer internships have also given me plenty of opportunity to experiment with issues in data management. At Microsoft, I evaluated efficiency of algorithms for discovering approximate functional dependencies. I evaluated the integration of relational database technology with artificial intelligence techniques for tagging unstructured text at IBM.

I also enjoy cross-discipline research because it is an excellent source of new applications for data management. At AT&T in 2000, I collaborated with programming language and speech recognition researchers to develop a new programming language for the development of interactive speech and visual interfaces for applications [Eurospeech 2001]. I contributed to the language constructs and runtime semantics of the new language. At IBM in 2004, I worked with researchers in data management and machine-learning text-analytics to bridge database technology with text understanding. I contributed to early design and system architecture evaluation, and considered different storage representations for the system. Collaborations on these two projects showed me that cross-discipline research requires a talented team that is ready to learn about research challenges outside of their own expertise.

Challenges in the Future of Data Management

In the future, my research will continue to focus on improving data management techniques for emerging applications. My path in this area largely depends on the individual needs of my future employer. I see many challenges for systems as more applications, such as file systems, personal information management, and scientific computing, employ data management solutions. One area that all of these applications will demand from data management is more accurate, efficient, and seamless heterogeneous data integration. Possible work in these areas includes:

- An extension of my thesis work would provide query interfaces to sparse data. Consider being faced with a product catalog of electronics where you want to search for products that record mp3s and have an equalizer with at least three settings. Out of the hundreds of attributes, you must guess which ones would contain product recording formats and information about equalizers. I am interested in the interface for data sets where the user knows very little about the attribute names, or in general, meta-data.
- The integration of heterogeneous data sets often involves combining information from several sources into one single view of the data. There are several difficulties in combining heterogeneous sources, but a major hurdle is in negotiating the diverse and incompatible models and schemas of disparate sources. Tools for suggesting mappings across data sources exist, but those tools do not consider the performance aspects of their mapping suggestions. For instance, a tool may suggest a mapping that requires a complex and long-running query over one data source. Such a mapping will affect the performance of all queries over that source. In such a case, would it be possible to suggest another mapping? Or if the data source allows, what kind of optimizations can be made at the source to make the query run faster?

My long term career goal is to be a technical leader in building more seamless, efficient, and manageable data management systems. My approach to research uses my systems knowledge to develop theoretical performance models and to demonstrate my solutions in real systems with real world data. In my dissertation work, I took a specific application, sparse data, and developed theoretical model for performance in relational database systems, identified solutions in storage and indexing, and used example e-commerce data to identify characteristics of sparse data and evaluate the performance of my techniques. In being a research leader, I look ahead to identifying more next-generation applications of data management technology and fostering collaborations necessary to enable those applications.