Query Processing

Vandana Jindal, Anil Kumar Verma, and Seema Bawa

***Abstract*—'Query' is retrieval of meaningful information. It may be done in one of the following ways – choosing parameters from a menu, Query by Example and Query language. While searching, it is essential that the user is able to understand what he is asking from the query. Energy efficiency is an important feature in designing and executing queries within the databases. Mistakes occur when the query processing stage in not well understood and attempts are made to execute too many calculations across a large result set during result processing, loading the system beyond its query rate capacity, generating a backlog of queries, which creates higher query latencies and possibly disruption of services.**

***Index Terms*—Query Processing, Query optimization, Contextual queries, Crosstab queries, Parameter queries.**

I. Introduction

The word derives from the Latin quaere i.e., to ask or seek. The word 'Query' is defined as retrieval of meaningful information from the database. Querying may be done in one of the following ways: **Choosing parameters from a menu, Query By Example and Query language**.

It furnishes various manipulations like retrieval of information (already available within the database), insertion of new information (into the database), deletion of information (from the database) and modification of data stored in the database. This is referred to as 'DML' (Data Manipulation Language). A subset of the DML for writing a query is referred to as a 'Query Language'. The means by which we can obtain the best plan, used in implementing the database request is 'Query Processing'. In search, true success comes from understanding what the user is asking from their query.

II. Query Types

The queries may be classified as depicted in Fig. 1.

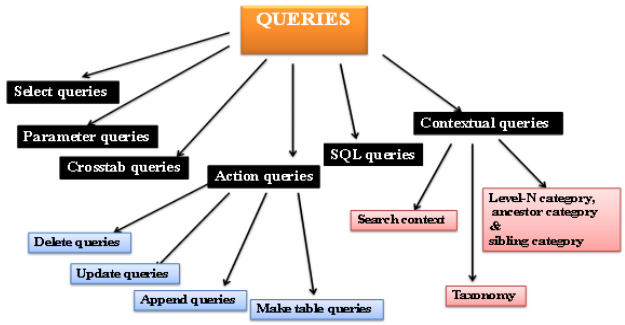

Fig. 1. The queries' classification.

Manuscript received October 29, 2010; revised February22,2011
The authors are with Thapar University, Patiala, India.

1) **Select queries:** These are used to retrieve data from one or more tables and display the result. It may also be used for providing various operations like sum, average, count etc.
2) **Parameter** queries**:** A type of interactive query which displays the results according to user specified criterion.
3) **Crosstab queries:** It is used for information analysis by summarizing the data in tabular form. Its use enables grouping of information in rows/ columns.
4) **Action queries:** Queries that perform some action/ changes to the records in a table. There are four types of action queries:

Delete queries – remove records from the tables.

Update queries – make global changes to a group of records in a table.

Append queries – add records from one or more tables to the end of one or more tables.

Make table queries – create a new table from all or part of the data in an existing table.

1) **SQL queries:** It is used for information retrieval from the database.
2) **Contextual queries:**

Search context – Contextual query language is a formal language used for representing queries to information retrieval systems. E.g. web indexes, bibliographic catalogs etc. The contextual query language has a single or multiple search clause(s) joined by Boolean operators. They are also associated with keywords, which may either be prefixed or may follow the clause.

Taxonomy – A taxonomy γ is a tree of categories where each node represents a predefined category. Each category is defined by the labels along the path from the root to the corresponding node.

Level-N category, ancestor category and sibling category: for a category c in taxonomy γ, c is called a level-n category if the node at c is located at nth level of γ.

III. General Strategy for Query Processing

Energy efficiency is an important feature in designing and executing databases. Query submitted by the user is not in a standard form that the system may understand. It has to be converted into computer understandable form. The query processor [1] transforms the query into a standard internal form like relational calculus, relational algebra, object graph, operator graph or tableau (see Fig. 2).

When a query is translated simply into its equivalent relational algebraic expression, it is form of non-procedural query language, but when the same is represented with a sequence of operations, it is said to be in a procedural form. This relational algebraic expression is represented as a Query Tree or a Query Graph. The query which is to be processed is passed through three stages:

1) Parsing and translation
2) Optimization
3) Evaluation

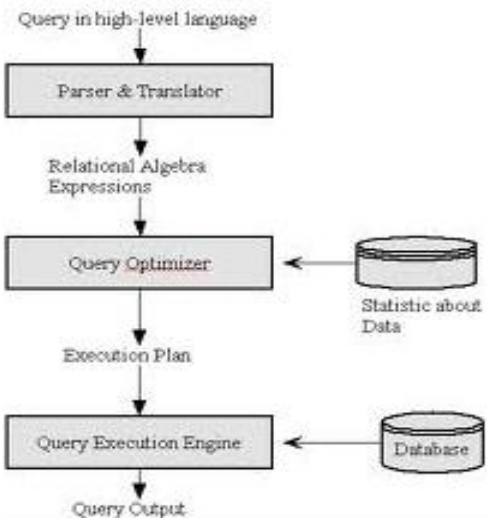

Fig. 2. Query processing [1].

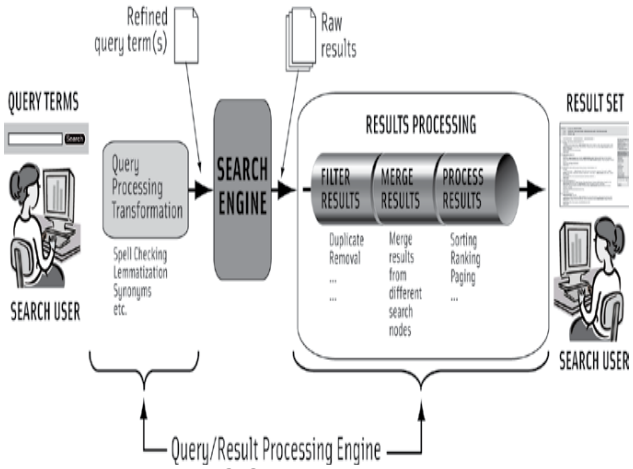

Fig. 3. Query processing strategy.

A. Parsing and Translating a Query

The main work of a query processor is to convert a query string into query objects i.e., converting the query submitted by the user, into a form understood by the query processing engine. It converts the search string into definite instructions. The query parser must analyze the query language i.e., recognizing and interpreting operators (AND, OR, NOT, +, - etc.), placing the operators into groups etc. The basic job of the parser is to extract the tokens (e.g. keywords, operators, operands, literal strings etc.) into their corresponding internal data elements (i.e., relational algebra operations and operands) and structures (i.e. query tree, query graph). Parser also verifies the validity and syntax of the query string.

B. Optimizing the Query

Query optimizer tries to find the most efficient way of executing a given query by considering the possible plans. It maximizes the performance of a query. It portrays the query plans as a "*tree*", results flowing from bottom to top. Query processor applies rules to the internal data structures of the query to transform these structures into their equivalent but more efficient representations. Rules may be based on various mathematical models and heuristics. Selection of application of rules – WHEN and HOW? are the basic functions of the query optimization engine.

C. Evaluating the Query

The last step in query processing is the evaluation phase (see Fig. 3). An evaluation plan tells precisely the algorithm for each operation along with the coordination among the operations. The best evaluation plan that a user generates by optimization engine is selected and then executed (There may exist various methods for executing the same query). The evaluation plan comprises of a relational algebra tree, providing information at each node (for each table) along with the implementation methods to be employed for each relational operator.

IV. Various Strategies for Query Processing

Various methods are present for query processing based on techniques like indexing etc. Selection has been made based on the recentness and references. By no means do we claim that the search has been exhaustive.

A. Indexing Methods

Indexing is a technique for improving the database performance. The significant property used is - that they eliminate the need to examine every entry while running a query. In vast databases this leads to reduction in time/cost. Indexes affect the performance and not the results. Given a particular query, the DBMS's query optimizer devises the most efficient strategy for finding matching data. The optimizer decides which index/indices to use.

Limitations:

- Speeds up the data access, but uses a lot of space in the database.
- Must be updated each time the data is altered.

Various indexing methods proposed are as follows:

1) Data Guide [2]

It is a brief and precise information pertaining to synopsis of structure of semi structured databases (i.e., not table-oriented as in a relational model or sorted-graph as in object databases). It may be employed for scanning, creating queries, accumulating information and enhancing query processing.

2) Forward and backward index [3]

This method is based on both inward and outward paths. It may be thought of as a **covering index i.e.,** Non-clustered index built upon all of the columns required to satisfy a SQL Server query, both in the selection criteria and in the WHERE clause.

3) APEX [4]

It is an Adaptive Path indEx for XML data. It is most suited where query follows a twig or a branch pattern. APEX while executing a query makes use of the most visited paths. It has to its credit that it can be updated incrementally in context to change in the query load.

4) BUS [5]

Bottom Up Scheme Index is used for updating documents which are structured in nature. Its essence lies in indexing

only the leaf nodes.

5) Ctree [6]

It is a compact tree indexing. The indexing structure of Ctree has both - the account of the path and the context relationships of elements.

Path summary is a tree that is distracted from the original data. In this new tree all equivalent paths are taken together like the DataGuide method does.

6) XISS [7]

It is XML Indexing and Storage System used for storing, indexing and querying XML data. The documents in XML are either path indexed, node indexed or sequence-based indexed. XISS is based on node indexing. It employs extended preorder numbering scheme for establishing relationship between the structured relational data and the semi-structured XML data.

7) ViST [8]

Virtual Suffix Tree provides indexing on both the content and structure of XML documents. It executes both dynamic insertion and deletion. Unlike other indexing methods where a complex query is sub-divided into smaller parts for the ease of execution and then the answers joined, ViST employs the tree structure to circumvent the problem of join operation.

8) Prufer Sequences [9]

It is a unique sequence associated with the tree, implemented using B^+ trees. Trees and sequences establish a one-to-one correspondence employing the Prufer Sequence.

B. Natural Language Processing Technique

Natural Language (NL) refers to the language spoken by people like Hindi, English etc. As is shown in Fig. 4, Natural Language Processing (NLP) refers to the applications that deal with NL in one way or the other. The processing is required because of the existence of vast amount of data on the internet and intranet. There are two NLP systems namely:

1) *Natural Language Understanding System* whose main task is to understand and find answers to the input (NL).
2) Natural *Language Generation System* is also referred to as text generation.

Fig. 4. Natural language processing.

As is shown in Fig. 5, the criteria followed are exact string matches for the query in a document. The documents are then ranked according to relevance. Using NLP increases precision of results by utilizing observations about discourse structure, noun phrases and proximity of matches of query terms. NLP steps performed were executed within EAGLE framework system [10]. The words were disambiguated based on their part of speech (noun/ verb). Semantically, irrelevant words were eliminated that added clutter (prepositions/conjunctions/pronouns). Though the above mentioned words are vital for proper analysis of complete sentence, these can be ignored for models using single-word or noun phrase size tokens.

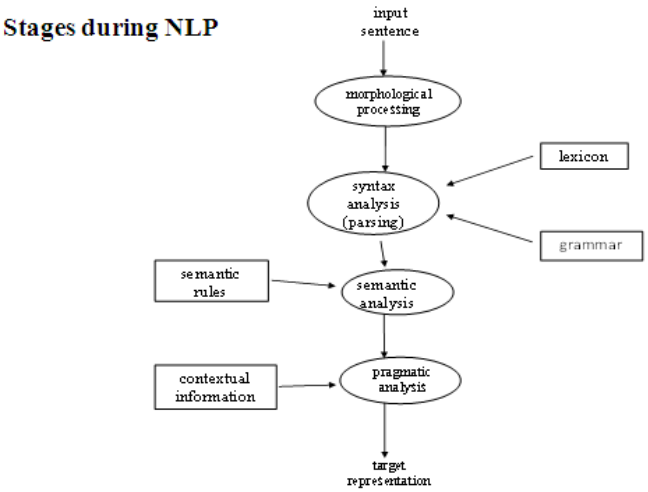

Fig. 5. Steps involved in processing of natural language.

Step 1: Morphological Processing

Strings are broken into tokens.

Step 2: Syntax Analysis/ Parsing

Checks strings of words, with the help of dictionary of word definitions (lexicon) and set of syntax rules (grammar)

Step 3: Semantic Analysis

Interprets the meaning of the sentences

Step 4: Pragmatic Analysis

Interprets the results of semantic analysis and derives knowledge from external commonsense information

Various models used are:

1) **Vector Space model:** A simple model based on linear algebra. It is a model used for representing text documents as vectors of identifiers. It finds its application in filtering of information, retrieving information, indexing etc. Allows ranking documents according to their possible relevance. The constraints associated with it is that the lengthy documents are poorly matched due lack of rich similarity value. Various models have come up based on this model's extendibility i.e., generalized vector space model, latent semantic analysis, term discrimination and Rocchio classification.
2) **Proximity model:** "Proximity" as the name suggests is the nearness in space, time or relationship. It is a method used in retrieving information to find documents that have words appearing "near" each other. It is a ranking system based on the following:

a) For any 2 words from the query, the closer they are to the document, the more pertinent the document is.
b) Greater the frequency of appearance of the pair in the document, greater the relevancy of the document.
c) If in the document the pairs appear in the same sequence as in the query they should find higher ranking.

Applications of NLP:

- Information retrieval/ detection.
- Searching and retrieving parts of a document.
- Information extraction fitting pre-defined templates.
- Answering questions i.e., fact retrieval.
- Translating documents from one language to another.
- Text summarizing.

- Automated customer service over the telephone.
- Tutoring systems (machine student interaction).
- Spoken language control of a machine.

Difficulties of NLP:

- Lack of robustness and efficiency.
- Ambiguity in how the complex structures could be compared for relevance.

C. Backing off Technique

Based on the assumption that NLP algorithms are imperfect and some information is not important for determining relevance of a document. It is a 5 stage process:

1) Match query terms exactly.
2) Use only root and part of speech information.
3) Account for NLP mistakes in part of speech assignments (software confuses adjective with proper nouns).
4) Match word roots only, ignoring morphology and POS Tags.
5) Match to a synonym or hyponym.

V. Measuring Query Processing

QPS (query per second) is the term used to define the measurement of processing a query and by noticing the significance of the results. These measurements are influenced by the following:

1) Hardware that is employed in the search matrix.
2) Restrictions imposed by licensing i.e., which determines the number of search needs.
3) The availability of the linguistic features in the index.
4) How complex is the query?
5) Various features invoked pertaining to the query.

VI. Challenges during Query Processing

The challenges faced during information retrieval are:

1) Comparatively improving the formulation of questions.
2) Depicting results in the easily judge able form so that results are closely related to the search.

VII. Query Processing - A word of caution

- The metrics used to evaluate query processing include QPS (query per second), the number of results returned per query; and the relevancy of these results as judged by the real users.
- Query load performance is measured as the maximum QPS number that the system can process with acceptable response times.
- Algorithms applied at query and response time need to be measured for speed and efficiency
- Mistakes most commonly made – is the failure to understand the query processing stage and the attempt to perform too many calculations across a large set during result processing loading the system beyond its query rate capacity, generating a backlog of queries, which creates higher query latencies and possibly disruption of services. The backlog results in timeouts and retransmits, generating higher loads on the search service, resulting into degrading service.

It is recommended that you plan for increasing query loads and upgrade the system accordingly.

VIII. Steps for Improving Search

1) A phased approach helps in recognizing the suitable characteristics which will be apt and will also enable in pinpointing the areas for further enhancements.
2) More caution should be taken in case of the search applications. As speed of processing is directly proportional to the number of search rows it would eventually promote scalability.
3) The bulk of the result must be taken into consideration. A trade off exists between speed and the fulfillment of the required result. More the data returned, more time is required for the data to flow back to the client.
4) An important factor is the applicability of the results. Users are usually concerned with only a small amount of the whole result set. It is very essential that they be fully versed with the cost factors involved.
5) Indexing may be employed for improving the database performance. Users must be aware of the searches like mixed relevance.

IX. Conclusion

There is an increasing demand for processing advanced types of queries in emerging application domains. Not only optimized query processing is required in DBMS but also in areas like - wireless networks, wireless sensor networks etc. where energy consumption is a crucial factor affecting the application and effectiveness of a (wireless sensor) network.

References

[1] A. Silbershatz, H. Korth, and S. Sudarshan, *Database System Concepts*, 4th ed. McGraw-Hill, 2002.

[2] F. Weigel, *Content Aware Data Guides for Indexing Semi-Structured Data*, University of Munich, Germany 2003.

[3] W. Wary *et al.*, "Efficient processing of XML Path queries using the Disk-based F&B index," in *Proc. Int'l Conf. Very large Database (VLDB)*, Morgan Kaufmann, 2005.

[4] C. W. Chung *et al*., "APEX: An Adaptive path index for XML data," in *Proc. Int'l Conf. Management Od Data (ACM Sigmod)*, ACM Press, 2002, pp. 121-132.

[5] D. W. Shin, H. C. Jang, and H. L. Jin, "BUS: An Effective Indexing and Retrieval Scheme in," in *Proc. Digital Libraries*, Pittsburgh, 1998, pp. 235-243.

[6] Q. Zou, S. Liu, and W. W. Chu, "Ctree: A Compact Tree for Indexing XML Data," in *Proc. the 6th Annual ACM International Workshop on Web Information and Data Management*, ACM, New York, NY, USA, 2004, pp. 39-46.

[7] Q.-Z. Li and B.-K. Moon, "Indexing and querying XML data for regular path expressions," in *Proc. the 27th Int'l Conference on Very Large Data Bases (VLDB)*, Rome, Italy, September 2001, pp. 361-370.

[8] H. Wang *et al.*, "ViST: A dynamic index method for quering XML data by tree structures," in *Proc. Int'l Conf. Management of Data (ACM Sigmod)*, ACM Press, 2003, pp. 110–121.

[9] P. R. Raw and B. Moon, "PRIX: Indexing and querying XML using prüfer sequences," in *Proc. Int'l Conf. Data Eng. (ICDE),* IEEE CS Press, 2004, pp. 288–300.

[10] C. Doran, J. C. Reynar *et al.*, 1997. "EAGLE: An extensible architecture for general linguistic engineering," RIAO'97, Montreal, Quebec, June 1997.