IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 12, December 2014

Query Processing in Global Information Systems

Lijo John

Lecturer In Computer Engineering, Thiagarajar Polytechnic College, Alagappanagar, Thrissur, Kerala.

Abstract: The retrieval of meaningful information is referred to as a query. It can be done in one of three ways: by selecting parameters from a menu, by Query by Example, or by Query language. It is critical that the user understands what he is asking from the query while searching. Energy efficiency is an important consideration when designing and running queries in databases. When the query processing stage is not well understood, 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, resulting in higher query latencies and possibly service disruption.

Keyword: Query Processing, Global information, DBMS , DML, Query translator, Query processing strategy.

I. INTRODUCTION

Query can be defined as the retrieval of meaningful information. The language used to provide various manipulations such as retrieval of already available information within the database, insertion of new information into the database, deletion of information from the database, and modification of data stored in the database is known as 'DML' (Data Manipulation Language). A Query Language is a subset of the DML that is used to write a query. 'Query Processing' is the method for obtaining the best plan for implementing the database request. [1] True success in search comes from understanding what the user is asking from their query.

The success of a database is entirely dependent on query handling. Query handling means that it completes the process of taking query input, processing it, and producing output related to it. It is not as simple as described in a single sentence. It entails a variety of resources and a complete mechanism to ensure that this process runs smoothly.

Queries are the primary functional unit of a database, and the database's functionality is entirely dependent on the output generated by queries. A query is any task or function that must be performed on a database. [2] Queries can perform tasks in the form of specific data gathered by various filtering criteria applied through conditions.

Queries can also be used to summarise data or manage it using various automated tasks. It can also generate new information based on existing data. Overall queries are the building blocks that manage the database's entire functionality. Query languages are used to implement queries. There are numerous languages available for writing queries in databases. [3] Queries may be written in either procedural or non-procedural languages.

A complete procedure to make the query work is expressed in procedural languages in the form of steps, such as what data should be imported, from where it should be imported, and in what order it should be executed; a complete explanation is provided explicitly. The main idea behind procedural languages is to know what to do and how to do it. [4]

Query types

1. Select queries: These are used to retrieve and display data from one or more tables.
2. Parameter queries: Used to create on-the-fly queries that prompt the user for criteria when the query is run.
3. Crosstab queries: These are used to summarise data from a single field and group it in a tabular format based on two criteria.
4. Action queries: Queries that modify the records in a table. Action queries are classified into four types:
    a. Delete queries: delete records from tables.
    b. Update queries - make global changes to a subset of a table's records.
    c. Append queries - append records from one or more tables to one or more tables.
    d. Run table queries to generate a new table from all or a portion of the data in an existing table.
5. SQL queries: A query written in SQL, a highly advanced query language.
6. Contextual questions:
    a. Search context - A formal language used to represent queries to information retrieval systems is contextual query language. For example, web indexes, bibliographic catalogues, and so on. A single or multiple search clause(s) are joined by Boolean operators in the contextual query language. They are also linked to keywords, which can be prefixed or followed by the clause. [5]
    b. Taxonomy A taxonomy is a tree of categories, with each node representing a distinct category. The labels that run from the root to the corresponding node define each category.
    c. 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

IJIREEICE
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526
INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 12, December 2014

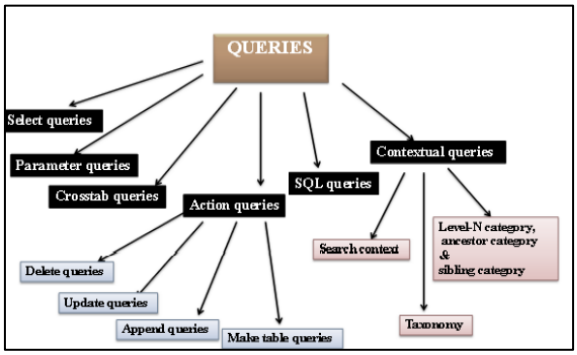

Fig. 1 The queries‟ classification.

Challenges during query processing

The following are the difficulties encountered during information retrieval:

1) Creating more effective questions.
2) Providing "easy-to-judge" results in order to minimise results (what the user has to read through)

Measuring query processing

The term QPS (query per second) refers to the measurement of processing a query and noting the significance of the results. The following factors have an impact on these measurements: [6]

1) The hardware used in the search matrix.
2) Restrictions imposed by licencing, i.e., the number of search needs.
3) The index's availability of linguistic features.
4) How difficult is the query?
5) Several query-related features were invoked.

Steps for improving search

1) A phased approach assists in identifying the appropriate characteristics as well as pinpointing areas for further improvement. [7-9]
2) When it comes to search applications, more caution should be exercised. Because processing speed is directly proportional to the number of search rows, scalability would eventually be promoted.
3) The majority of the outcome must be considered. There is a trade-off between speed and achieving the desired result. The more data returned, the longer it takes for the data to flow back to the client.
4) The results' applicability is an important consideration. Users are usually only interested in a small portion of the total result set. It is critical that they are fully aware of the cost factors involved. [10-11]
5) Indexing can be used to improve database performance. Users must be aware of searches such as mixed relevance.

Objectives

- investigate processed query results
- investigate query processing stages
- investigate query processing strategy
- investigate challenges encountered during query processing

II. RESEARCH METHODOLOGY

In common parlance, research refers to a search for knowledge; however, research can also be defined as a scientific and systematic search for relevant information on a scientific topic. Clifford Woody defines research as "defining and redefining problems, formulating hypotheses or suggested solutions, collecting, organising, and evaluating data, marking deductions and reaching conclusions, and finally carefully testing the conclusions to determine whether they fit the formulating hypothesis."

III. RESULT AND DISCUSSION

Table I shows a description of the various clauses as well as their order.

Table 1. CLAUSES IN SQL

Order	Clause	Function
1	From	Choose from the table or joins if more than one table
2	Where	Applies filtering criteria
3	Group be	Aggregates the data of tables
4	Having	Filters the aggregated data
5	Select	Return the result
6	Order by	Sorts the result
7	limit	Limits the restaurant data

The query translator then translates the query and sends it to the unit. Figure 2[12-13] shows an example.

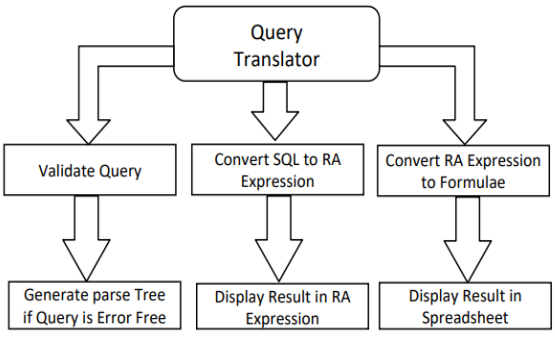

Fig. 2 Query translator

The evaluation phase is the final step in query processing (see Fig. 3). An evaluation plan specifies the algorithm for each operation as well as the coordination between them. The best evaluation plan generated by the optimization engine is chosen and then executed (There may exist various methods

for executing the same query). The evaluation strategy consists of a relational algebra tree with information at each node (for each table) and implementation methods for each relational operator. [14-15]

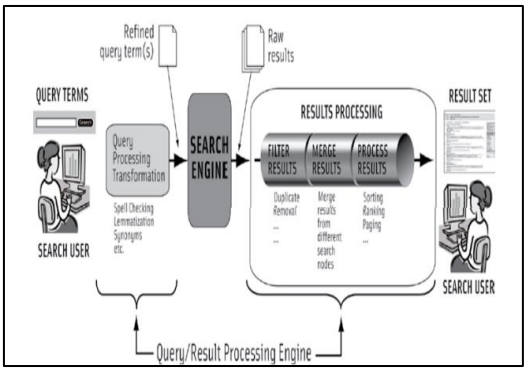

Fig. 3 Query processing strategy

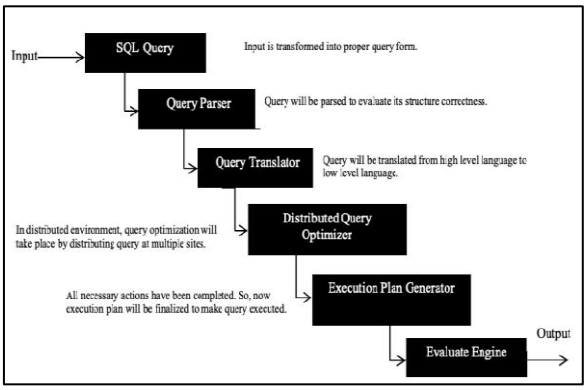

Fig. 4 Stages of Query Processing

Query processing involves several steps, including a query parser, translator, optimizer, execution engine, and evaluate engine. The query processing is depicted above.

Table 2. Results of proceeded queries

Size of query (no. of targeted attributes)	Execution time (In second)	Total Returned records	Expected number of Records	Efficiency	Pay load
1	0.4	35	49	71.4%	35
1	0.3	1	49	2.04%	4
3	0.5	17	252	6.7%	17
4	1	22	510	4.3%	22

The query processing time, number of records fetched, and processing efficiency are all shown in the table above.

IV. CONCLUSION

In emerging application domains, there is an increasing demand for processing advanced types of queries. Not only is optimised query processing required in DBMS, but it is also required in areas such as wireless networks, wireless sensor networks, and so on, where energy consumption is a critical factor affecting the application and effectiveness of a (wireless sensor) network. If the execution time is short and the number of records returned is large, the query performance is evaluated. Furthermore, for the same time interval, simple queries returned more records due to a single attribute selection. When more than one attribute participates in query formulation, the time required increases. Map reduction is also important.

REFERENCES

[1] W. Wary et al.," Efficient processing of XML Path queries using the Disk-based F&B index", Proc. Int'l Conf. Very large Database (VLDB), Morgan Kaufmann 2005.
[2] C. W. Chung et al.,"APEX: An Adaptive path index for XML data", Proc. Int'l Conf. Management od data (ACM Sigmod), ACM Press, 2002, pp. 121-132.
[3] D.W. Shin, H.C. Jang and H.L. Jin, "BUS: An Effective Indexing and Retrieval Scheme in", In Proc. of digital libraries, pp 235-243, Pittsburgh, 1998
[4] H. Wang et al., "ViST: A Dynamic Index Method for Quering XML Data by Tree Structures", Proc. Int'l Conf. Management of Data (ACM Sigmod), ACM Press, 2003, pp. 110–121.
[5] P.R. Raw and B. Moon, "PRIX: Indexing and Querying XML Using Prüfer Sequences", Proc. Int'l Conf. Data Eng. (ICDE), IEEE CS Press, 2004, pp. 288–300.
[6] F. Weigel, Content Aware Data Guides for Indexing Semi-Structured Data, University of Munich, Germany 2003.
[7] 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.
[8] 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.
[9] 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.
[10] 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.
[11] 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.
[12] 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.
[13] 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.
[14] 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.
[15] 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