2010 2nd International Conference on Computer Technology and Development (ICCTD 2010)

Bio-inspired Algorithms for Query Optimization in Biological Databases

L. Melita
Department of Computer Science,
Jimma University,
Jimma, Ethiopia
pmmelita@gmail.com

Gopinath Ganapathy
Department of Computer Science,
Bharathidasan University,
Tiruchirapalli, India
gganapathy@gmail.com

P.Prakash
R & D Department,
OMNE Agate Systems Pvt. Ltd,
Chennai, India
prakash.paraman@gmail.com

Abstract — **Despite of the numerous research efforts on distributed query processing, the complexity of the problem has been little solved which paves way for the exploration on the solution of the problem. However, due to its inherent difficulty, the complexity of the majority of problems on distributed query optimization remains unknown. In this paper, we analyze and present the problems identified and the possible algorithms used for solving them. The higher probability of problem solving nature of the Bio-inspired Algorithms has been projected with a Comparative Study on literature proof and as a result a proposal for Query Optimization on Biological Databases is introduced.**

Keywords: Bio-Computing; Memetic Algorithms; Query Optimization.

I. INTRODUCTION

Due to new distributed database applications such as huge deductive database systems, the search complexity is constantly increasing and we need better algorithms to speedup traditional relational database queries.

Query processing is much more difficult in distributed environment than in centralized environment because a large number of parameters affect the performance of distributed queries, relations may be fragmented and/or replicated, and considering many sites to access, query response time may become very high [1].

It is quite evident that the performance of a Distributed Database System is critically dependant upon the ability of the query optimization algorithm to derive efficient query processing strategies. The distributed query optimization has several problems related to the cost model, larger set of queries, optimization cost, and optimization interval [2].

Since data are geographically distributed in such a system, the processing of a distributed query, as pointed out in [3], is composed of the following three phases:

- ✓ Local processing phase which involves all local processing such as selections and projections,
- ✓ Reduction phase where a sequence of reducers (i.e, semijoins and joins) is used to reduce the size of relations, and
- ✓ Final processing phase, in which all resulting relations are sent to the assembly site where the final query processing is performed.

In Specific, Biological databases taken into concern being the most important among the classes of scientific databases, exhibit many special characteristics that make management of biological information a particularly challenging problem.

The high amount and diversity of data available in the large number of different databases to the scientific user's advantage, creates the problem of knowing exactly where to get specific information. Another problem is that different databases with different contents also use different formats adding further technical difficulties to the already complex task of accessing and exploiting the data.

This paper describes the different classes of problems in querying and query processing and also discusses the techniques available to overcome them.

II. QUERY OPTIMIZATION ALGORITHMS

A. Classification of Optimization Algorithms

The most common types of algorithms for optimization are deterministic and non-deterministic algorithms [4].

The classical dynamic programming is also a Deterministic algorithm (also known as exhaustive search dynamic programming algorithm), produces optimal left-deep processing trees with the big disadvantage of having an exponential running time. This means that for queries with more than 10-15 joins, the running time and space complexity explodes [4].

The difficulties associated with using mathematical optimization on large-scale optimization problems have contributed to the development of alternative solutions.

978-1-4244-8845-2/10/ $ 26.00 © 2010 IEEE

Linear programming and dynamic programming techniques, for example, often fail (or reach local optimum) in solving NP-hard problems with large number of variables and non-linear objective functions [5]. To overcome these problems, researchers have proposed evolutionary-based algorithms for searching near-optimum solutions to problems.

The non-deterministic algorithms or cost based optimization algorithms say evolutionary algorithms [6]-[7] on the other hand do not generally produce an optimal access plan. But in exchange they are superior to dynamic programming in terms of running time.

B. Problems in Query Optimization

Traditionally the optimization goal has been minimization of the total cost of execution, but in many applications, other factors such as response time, staleness of the data used in answering the query [8], or accuracy of the data [9] may also be critical.

It has been found that the following five classes of distributed query optimization problems cover the majority of distributed query optimization problems studied in the literature [10].

- ✓ *Local optimization of semijoins:* Determine the optimal set of semijoins to reduce a single relation.
- ✓ *Join sequence optimization:* Determine the optimal join sequence to transfer relations to the assembly site.
- ✓ *Relation semijoins on broadcasting networks:* Solve the semijoin optimization problem on broadcasting networks.
- ✓ *Single-reducer tree queries:* Determine the minimal cost semijoin sequence to fully reduce the root relation of a tree query.
- ✓ *Full-reducer tree queries:* Determine the minimal cost semijoin sequence to fully reduce all relations in a tree query.

For each class of problems, it has been described that the corresponding problem is to be reduced first, and then it has been formally proved that the reduction derived is valid and can be done in polynomial time [10], which is yet to be effectively implemented in a selected algorithm.

III. Biological Databases

A variety of biological databases has been developed that provide database support for research activities conducted in different biological disciplines and practical applications in the pharmaceutical industry. Protein or DNA sequence data are the primary data that reside in these databases while various related data such as annotations, mutant information and physico-chemical characteristics are often added as well.

A. Complexity of Biological Data

Despite of the higher usability of the biological databases, the specific features of the biological data to be incorporated in the databases are complexive. Some of such features are listed below:

- Representing Biological data is intricate when the data is complexive
- Representation and Understanding of every Biologists differ for the same data, influencing the quality of data
- Attributes may increase/change with time and usability
- Retention of modified and prior values is required for every copy of data
- Voluminous data with variations

These factors in turn influence the method of querying by the Biologists. Biologists being quite unaware of the logical structure of the database may not know how to retrieve the required information or understand the representation of the data either. This affects querying and hence its performance. The following techniques may help improving the methods of querying.

- The tool to self-describe the representation and the context of the data
- Coherence of the different formats of information, whether they blend, when retrieving from different databases
- Query builder to build simple/ complex queries
- Interactive interfaces to clarify the statement/ requirement of the query
- Sample output (intelligently retrieved from earlier transactions) to confirm from the user, prior to request passing to different remote servers.
- Biologists have to represent right queries
- Interpreter to state the built query to the database

IV. Query Optimization on Biological Databases using Bio-inspired Algorithms

The integration of numerical algebraic calculations in database systems enables to perform automatic optimization of entire computations, with the resulting benefits of query optimization, algorithm selection and data independence becoming available to computations on scientific databases. This removes the barrier between the database system and the computation [11] allowing the database optimizer to manipulate a larger portion of the application.

In the optimization of scientific computations, the identification of suitable transformation rules is of central importance. Those rules are applied to generate equivalent logical expressions, and the optimizer must find a set of physical algorithms that can implement or execute each expression. For instance, a join operator can be implemented as either a merge- or hash-based algorithm, while an interpolation can be implemented by any of variety of curve fitting algorithms. Other query optimizer issues include limiting the search space, detecting common sub-expressions and improving cost estimation. Moreover, the user should be able to enable or disable certain logical transformations to control the accuracy of the results of the equation.

The task of retrieving data of different formats from different databases adding further technical difficulties to the already complex task of accessing and exploiting the data must be addressed.

A. Bio-inspired Algorithms

Taking the Bio-inspired Algorithms say Evolutionary algorithms (EAs) into concern, they are the stochastic search methods that mimic the natural biological evolution and/or the social behavior of species. Such algorithms have been developed to arrive at near-optimum solutions to large-scale optimization problems, for which traditional mathematical techniques may fail [12].

The first evolutionary-based technique introduced in the literature was the genetic algorithms (GAs) [13]. GA's were developed based on the Darwinian principle of the 'survival of the fittest' and the natural process of evolution through reproduction. Despite of its demonstrated ability to reach near-optimum solutions to large problems, GA's may require long processing time for a near optimum solution to evolve.

In an attempt to reduce processing time and improve the quality of solutions, particularly to avoid being trapped in local optima, other EAs have been introduced during the past 10 years. In addition to various GA improvements, recent developments in EAs include four other techniques inspired by different natural processes: memetic algorithms (MAs) [14], particle swarm optimization (PSO) [15], ant colony systems [16], and shuffled frog leaping (SFL) [17].

B. Comparison of Bio-inspired Algorithms

Based on the comparative study and reports given by Emad Elbeltagi et al [12], Sidhartha Panda et al [18], Poonam Garg [19], Ju´lius S et al [20] and Michael Wetter et al [21], the following statements are generalized.

Surprisingly, the GA performed more poorly than all the other four algorithms. It was noticed that as the number of variables increased, the processing time by GA to reach the target also increased.

Upon applying the MA, the results improved significantly compared to those obtained using the GA, in terms of both the success rate and the processing time. That is to say, the local search of the MA improved upon the performance of the GA.

The PSO algorithm outperformed the GA and the MA in solving simple functions in terms of the success rate and the processing time, while it was less successful than the MA in solving the complex functions with increase in number of variables.

The ACO algorithm could be applied only to the discrete optimization problems, which gives the same success rate, as that of GA whereas the processing time comparatively increases.

While the success rate for the SFL was zero for complex/simple functions, it was found to have a better performance than GA.

C. Bio-Computing for Query Optimization.

It is quite evident that the performance of a database is critically dependant upon the ability of the query optimization algorithm to derive efficient query processing strategies.

The goal is to execute such queries as efficiently as possible in order to minimize the response time that users must wait for answers or the time application programs are delayed. And to minimize the total communication costs associated with a query, to improved throughput via parallel processing, sharing of data and equipment, and modular expansion of data management capacity. In addition, when redundant data is maintained, one also achieves increased data reliability and improved response time.

There is no doubt that dynamic programming method always gives us optimal solution, however, since the time and space complexity of the GA-base optimization is much less, it is the wise choice to go for Evolutionary (Bio-inspired) Algorithms.

According to the experimental results we can conclude that for the problem of query optimization, including a high quality heuristic solution can help the GA to improve its performance by reducing the likelihood of its premature convergence. It can be seen that Memetic Algorithm seems to be more sensitive to the amount of complex data and outperform all other Evolutionary Algorithms, including GA in this case. Hence an improvised memetic algorithm can be a better solution for the query optimization problem.

V. CONCLUSION

Considering the Bio-inspired algorithms for the Query Optimization, the conclusion can be made that the Memetic Algorithms shall out perform the rest of the algorithms, as Biological databases handle complex data and the query request may take any number of variables with convoluted statistical and aggregate/ non-aggregate functions. A Cellular Memetic algorithm can be implemented in order to improve the performance of the algorithm, there by restricting the interactions of the individuals of the population, which is under the author's research.

REFERENCES

[1] Li, Victor O. K. "Query processing in distributed data bases", MIT. Lab. for Information and Decision Systems Series/Report no.: LIDS-P; 1107, 1981.

[2] Reza Ghaemi, Amin Milani Fard, Hamid Tabatabaee, and Mahdi Sadeghizadeh, Evolutionary Query Optimization for Heterogeneous Distributed Database Systems", World Academy of Science, Engineering and Technology 43, 2008.

[3] C. Yu, Z.M. Ozsoyoglu, and K. Kam, "Optimization of Distributed Tree Queries," Comput. Syst. Sci., vol. 29, no. 3, pp. 409-445, Dec. 1984.

[4] Kristina Zelenay,"Query Optimization", ETH Zürich, Seminar Algorithmen für Datenbanksysteme, June 2005.

[5] Lovbjerg M. Improving particle swarm optimization by hybridization of stochastic search heuristics and self-organized criticality. Masters Thesis, Aarhus Universitet, Denmark; 2002.

[6] Yannis E. Ioannidis and Youngkyung Cha Kang, "Randomized Algorithms for Optimizing Large Join Queries".
[7] Michael Steinbrunn, Guido Moerkotte, Alfons Kemper, "Heuristic and Randomized Optimization for the Join Ordering Problem", The VLDB Journal - The International Journal on Very Large Data Bases, Volume 6, Issue 3 (August 1997), Pages: 191-208, ISSN:1066-8888.
[8] C. Olston and J. Widom. Offering a precision-performance tradeoff for aggregation queries over replicated data. In VLDB, 2000.
[9] R. Avnur, J. M. Hellerstein, B. Lo, C. Olston, B. Raman, V. Raman, T. Roth, and K. Wylie. Control: Continuous output and navigation technology with refinement on-line. In SIGMOD, 1998.
[10] Chihping Wang, Member, IEEE, and Ming-Syan Chen, Senior Member, IEEE, "On the Complexity of Distributed Query Optimization", IEEE Transactions on Knowledge and Data Engineering, Vol. 8, **No.** 4, August 1996.
[11] Aleksejs Kontijevskis, "Scientific databases - Biological data management", Uppsala University, April, 2007.
[12] Emad Elbeltagi, Tarek Hegazy, Donald Grierson, Comparison among five evolutionary-based optimization algorithms, Advanced Engineering Informatics 19 (2005) 43–53.
[13] Holland J. Adaptation in natural and artificial systems. Ann Arbor, MI: University of Michigan Press; 1975.
[14] Moscato P. On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Technical Report Caltech Concurrent Computation Program, Report 826, California Institute of Technology, Pasadena, CA; 1989.
[15] Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of the IEEE international conference on neural networks (Perth, Australia), 1942–1948. Piscataway, NJ: IEEE Service Center; 1995.
[16] Dorigo M,Maniezzo.V, Colorni A. Ant system: optimization by a colony of cooperating agents. IEEE Trans SystMan Cybern 1996;26(1):29–41.
[17] Eusuff MM, Lansey KE. Optimization of water distribution network design using the shuffled frog leaping algorithm. J Water Resour Plan Manage 2003;129(3):210–25.
[18] Sidhartha Panda, N. P. Padhy, Comparison of Particle Swarm Optimization and Genetic Algorithm for TCSC-based Controller Design, International Journal of Computer Science and Engineering 1;1 2007.
[19] Poonam Garg, A Comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of Simplified Data Encryption Standard algorithm, International Journal of Network Security & Its Applications (IJNSA), Vol.1, No 1, April 2009.
[20] Ju´ lius Sˇ troffek, Tom´aˇs Kovaˇr´ık, Execution Plan Optimization Techniques, Sun Microsystems PostgreSQL Conference, 2007.
[21] Michael Wetter, Jonathan Wright, A comparison of deterministic and probabilistic optimization algorithms for non-smooth simulation-based optimization, Building and Environment 39 (2004) 989 – 999.