ISSN: 2348 - 2117
International Journal of Engineering Technology and Computer Research (IJETCR)
Available Online at www.ijetcr.org
Volume 3; Issue 3; May-June 2015; Page No. 188-193

Symbiotic Organisms Search Framework for Distributed Database Optimization

[1]Atinderpal Singh, [2]Manreet Sohal,[3]Rajinder Singh Virk

Department of Computer Science and Engineering, Guru Nanak Dev University, Amritsar, Punjab.

[1]saini_amrit@live.com, [2]reetsohal@yahoo.com, [3]tovirk@yahoo.com.

Abstract

Optimization in database field is a challenging domain of research and it has attracted attention in few years. This paper draws attention to the query optimization in distributed database. Query Processing is sole concept incorporated in distributed database system. The research has revealed that the design of a query and its execution strategy plays an inevitable role in the optimization of a query. We have to employ the execution approach that produces optimal result. Data in distributed system is allocated at various sites. The query is answered by joining various relations in optimal manner. Communication cost plays a vital role in query processing in distributed environment. Various techniques has been be deployed to solve the above stated problem. In this paper we have proposed a new framework based on Symbiotic Organisms Search Algorithm to solve the optimization problem in distributed environment.

Keywords: Symbiotic Organisms Search, Distributed Database Design, Query Optimization, SOS Framework.

I.INTRODUCTION

Distributed Database consists of two or more data files located at different sites on a computer network [3,11].A distributed database is as a collection of multiple, logically interrelateddatabases distributed over several sites. A distributed database managementsystem (distributed DBMS) is defined as the software system that permits themanagement of the distributed database and makes the distribution transparent to theusers. Data may be stored in multiple computers located at same locations or it may be dispersed over a network of interconnected computers.Query Processing is defined as the activities involved in parsing, validating, optimizing and executing a query. The main aim of query processing is to transform a query written in high level language into efficient and correct strategy expressed in low level language [11].Query processing refers to the process of ensuring that the total cost or response time for query is minimized. The choice to be made by query optimizer include

- The order in which the various operations are executed.
- Data movement order between the sites.
- The algorithms for carrying out the various operations.

Finest cost disposition is the response that is expected for query optimization in Distributed system. Query optimizer plays this vital role to achieve the required goal. Optimization become complex when number of relations is increased at various distinct sites. The various operations in optimization processing are to be performed in such a manner that it gives optimal plan that leads to optimal result.

The query processing incorporates distributing and allocating various operations over multiple sites and response to it is received at a resultant site that gathers required data from distinct relations in distributed system. The whole process involves a cost associated with different operations and processing includes the cost criteria that incur minimum cost. Various concepts and techniques have been implemented for distributed queries. In this paper we have proposed a new framework for solving query optimization problem in distributed system using symbiotic organism search technique. We will implement it MATLAB in near future using simulator that we designed for allocation various operation over multiple sites and it includes all the performance criteria required for query processing. The rest of paper is discussed below in various sections.

II.RELATED WORK

Après et al proposed three styles of an optimization algorithm for random complex queries, one for minimizing the response time and two for minimizing total time cost of a distributed query under certain conditions for a distributed database. Perrizo extended this work to propose greedy approach for

Corresponding author: Atinderpal Singh

apair of joins instead of single join considered earlier. [1,2].

William Grosky et al.uses an adaptive selectivity estimation scheme for multidimensional queries which performs better than non-adaptive methods when the distribution of the data is not known. This research overcomes the disadvantages of previously formulated non-adaptive, static methods which are relatively inaccurate in a dynamic database [4].

J**orang H et al.** in their research presented the genetic approach considering apposite data structure to decrease the cost associated with query processing in scattered environment. Simulation was carried out for the disseminated queries and the result conveyed that genetic approach proved to be healthier in computing parameters and quality provided as equated to various other [5] .

Rho Sangkyu et al.in their paper compared the various distributed database design models. In distributed data replication, join node selection, join order and reduction by semi join all makes a significant impact on the efficiency of a system. They have found that data replication was effective for retrieval and low selectivity concerns. The combination of join ordering, join node selection and reduction by semi join has shown marginal improvement and this combination was effective only for retrieval, update and low selectivity situations. Their results also conveys that there is some relation between total operating cost and response time design criteria [6]

Michael G et al. in their work revealed the latent of genetic technique to optimize the geologicacally disseminated queries at several network sites. The anticipated GA in their research employed tree organised data concept. It considers the use of local search to deliver real time answers. The lessening of dispensation cost of a query in distributed environment m is revealed. The delinquentframedruminate a distributed query tree concerning number of relations over several sites in a set-up and the outcomediscloses the minimal cost processed design to contract all relations in the form of solitary query.Local huntpart proves to be the preeminent part of execution plot to obtain ideal results.Trials were piloted on Pentium system and genetic approach evidenced to be superior than Random Search, Simulated Annealing and Multi Search techniques [7].

Ahmet C et al. anticipatednoveledgenetic algorithm incorporating new mutation and cross over operators and smeared with greedy procedure for gaining better concert by copies of replicated relations at adjoining site.Different Constraints for genetic approach were considered and result revealed that two-point truncate technique showed to producepreeminent results. In new framed GA cross-over point is dogged considering heuristic that has avaricious smanner to implement.When problem dimensionescalates with growth in number of relations than Novelette GA ascertain to be effective as equated to GA and Meticulous technique with improved solution value [8].

Areerat T et al. have proposed Exhaustive Greedy (EG) algorithm to optimize intermediate result sizes of join queries. Most intermediate result sizes of join queries estimated by the EG algorithm are comparable to the results estimated by the Exhaustive Search algorithm (ESU)that is modified to update join graphs[9].

NarasimhaiahGorla et al.has optimized the sub query allocation using genetic algorithm. Since the major concern in the query processing in distributed database system is to minimize the query processing. So based on the query type the different allocation of the sub query to various sites is to be optimized. It was found that GA produce better results as compared to other approaches[10].

Fan Y et al. have designed a new algorithm based on heuristic optimization that can effectively reduce the communication cost by reducing the amount of intermediate result data.The major concern in this algorithm is based on relational algebra equivalence transformation to form a query tree while performing selection and projection operations upto fragment definition[12].

Golshanara et al. In this paper, for the first time, a multi-colony ant algorithm is proposed for optimizing join queries in a distributed environment where relations can be replicated but not fragmented. In the proposed algorithm, four types of ants collaborate to create an execution plan. Hence, there are four ant colonies in each iteration[13].

Each type of ant makes an important decision to find the optimal plan. In order to evaluate the quality of the generated plan, two cost models are used—one based on the total time and the other on the response time. The proposed algorithm is compared with two previous genetic-based algorithms on chain, tree and cyclic queries. The experimental results show that the proposed algorithm saves up to about 80 % of optimization time with no significant difference in the quality of generated plans compared with the best existing genetic-based algorithm.

SOS based distributed database simulator will be deployed for optimization and it may provide positive results since it has been validated for optimization against various metaheuristic approaches considering several engineering problems.

III. SYMBIOTIC SEARCH ORGANISMS OPTIMIZATION

Symbiotic Organism Search optimization is stirred by usualcommunicationcampaigns of organisms that are assimilated by them for nourishing life in the biome. It is a dominantMeta heuristic procedure that can be deployed to several engineering tasks [14].

Resemblingother nature stirredproceduresSOSmimics the association between corresponding organisms of variousclasses for hunts those outcomes in acceptable and fittest organism. The SOS concepts evolve starting with initial ecosystem size. Every organism in the ecosystem has considering fitness value that conveys their chance to accomplish desired objective. The next age group of organisms is acquired by initializing the bioticcommunicationamid various organisms.The type of communication among organism outlines the main objective of each part.Each organism communicates with other in all three levels.The process lasts until the essential conditions are not fulfilled.

The relationships that are common among distinct species organisms are as follows:

- ***Mutualism***: Relation in which organism of different species have advantage in living together.
- ***Commensalism***: Relation in which one of the two organisms have profit in living together at same place ,the other remains unchanged.
- ***Parasitism***: Relation in which one organism is at profit while other in loss living together.

SOS applies greedy strategy at the end of each interactive phase to have best organism in ecosystem. It considers the two basic parameters for its formulation.

- Population Size.
- Maximum Generations.

It delivers solution to extensive range of problems and is more vigorous than other computing algorithms and it avoid risk that may result in compromised performance.

The general outline of the algorithm is given below.

{
Initial population(Ecosystem);
While(The termination criteria is not met)
{
Mutualism Phase;
Commensalism Phase;
Parasitism Phase;
}
Output the best Organism;
}

.

Fig. 1: General Outline of SOS Algorithm.

IV. DATABASE DESIGN AND PROBLEM DESCRIPTION

The performance of strategy that is applied for distributed queries is dependent on the size of fragment that are produced when various operations of query are executed. The set of operations that are generated in answering a query is represented by an operator graph. Operations are represented by the nodes of graph and cost by edges or lines. The various assumptions are the allocation profiles, size of relations, transaction that occur at various sities and the total cost which involve the major communication cost[3,4,11,12].

1. Database Design

The Design of distributed database is simulated considering a set 'S' of network sites, a set 'R' of relations(tables) stored at various sites and a Set 'Q' as set of transactions. Atransaction query (*q)*for information retrieval, is broken into a set of sub queries on the 'R' set of relations[11].

2. Objective Formulation

Given a set of Relations or fragments

$R = \{r_1, r_2, \ldots, r_n\}$

A grid of sites $S = \{s_1, s_2, \ldots, s_m\}$

A set of sub queries $Q = \{q_1, q_2, \ldots, q_q\}$

A) Input Variables

DAV_{rs}: Data Allocation Scheme that have matrix representation for I/O cost and communication cost.

IF^{q}_{rK}: Matrix that represent the intermediate fragments ' *r*'used by a sub query 'K'of main query 'q'.

B) Output to be Generated

SDD^{q}_{Ks}**:** An Operation Allocation Scheme Matrix which optimizes objective function.

Distribution of relations to sites represented by cost function involving query processing cost and storage cost.

$$Total\ Cost = \sum_{\forall qi \in Q} QPi + \sum_{\forall s \in S} \sum_{\forall rj \in R} ST_{jp}$$

- Here $QP_i$is the cost for processing query q_i for any application.
- ST_{jp} is the cost for storing fragments $F_j$at site S_p .
- The total cost for the query is sum of local processing cost and Transmission cost i.e. according to OZSU model.
- Further we have modified it considering only the retrieval transaction according to our design.Query processing cost is given below.

$$QP_i = LP\_COST + COMM\_COST$$

- LP_COST represents the local processing cost.
- COMM_COST represents the communication cost involved in distributed query processing. Also processing cost include the access cost and does not include the integrity cost and concurrent update control cost for model with retrieval transactions.
- The best way to calculate cost of a query execution plan is to minimize the 'Total Cost of Query', which is represented in terms of time units and refers to use of resources such as CPU Cycles, Disk I/O & Communication Channels by a candidate allocation plan.

V. SOS FRAMEWORK FOR DISTRIBUTED DATABASE QUERIES

This section introduces the step-wise procedure for implementing SOS for distributed queries.

Step 1 : Data Distribution Scheme along with fragmentation schemes is given as input to the SOS_DDQ.

- A set of transactions on the given database is considered for processing along with transaction profiles.
- The Execution Order Query is given and size of intermediate fragments is estimated considering standard techniques using selectivity factors while determining the query execution order.
- Communication cost between each pair of Network Sites is given by Communication Coefficients and it is fed as input to SOS_DDQ.

Step 2: Ecosystem Initialization

Optimization parameters for Distributed Query Processing:

- Number of Organisms:50
- Maximum Iterations/Generations: 50
- Reaching the maximum iteration is the stopping criteria.
- Initial ecosystem is randomly generated along with corresponding fitness value based on objective function.

Step 3: Consider the best (organism) solution, Z_{best}.

- The organism with that has least minimum fitness value out of the entire ecosystem is chosen as Z_{best}.

Step 4: Mutualism Phase

- Starting with initial the first organism Z_1 is matched with Z_i and another organism Z_2 is selected randomly from the pool of ecosystem. Here Z_2 is considered as Z_j.
- Mutual vector is determined that represents the interaction among organisms and also benefit factors are determined.
- New organisms are compared with the previous one and fitter organisms are considered for next generation.

Step 5: Commensalism Phase

- An organism is selected randomly from ecosystem. New candidate solution Z_1 is calculated and compared with previous Z_1.
- The fitter organism is considered as solution for next generation and updated accordingly.

Step 6: Parasitism Phase

- An organism is selected randomly from ecosystem. Then parasite vector is created by mutuating with Z_i in random dimensions.
- Parasite vector is compared with the randomly selected organism.
- The fitter organism is considered as solution for next generation and the other organism is discarded.

Step 7: Move to step 3 if the current Z_i is not the last organism in pool of ecosystem, other wise move to next step.

Step 8: Stop if the maximum number of generations is reached and print the optimal solution, else move to step 3 and repeat the whole process.

The Flow of SOS_DDQ design is below and it explains the working.

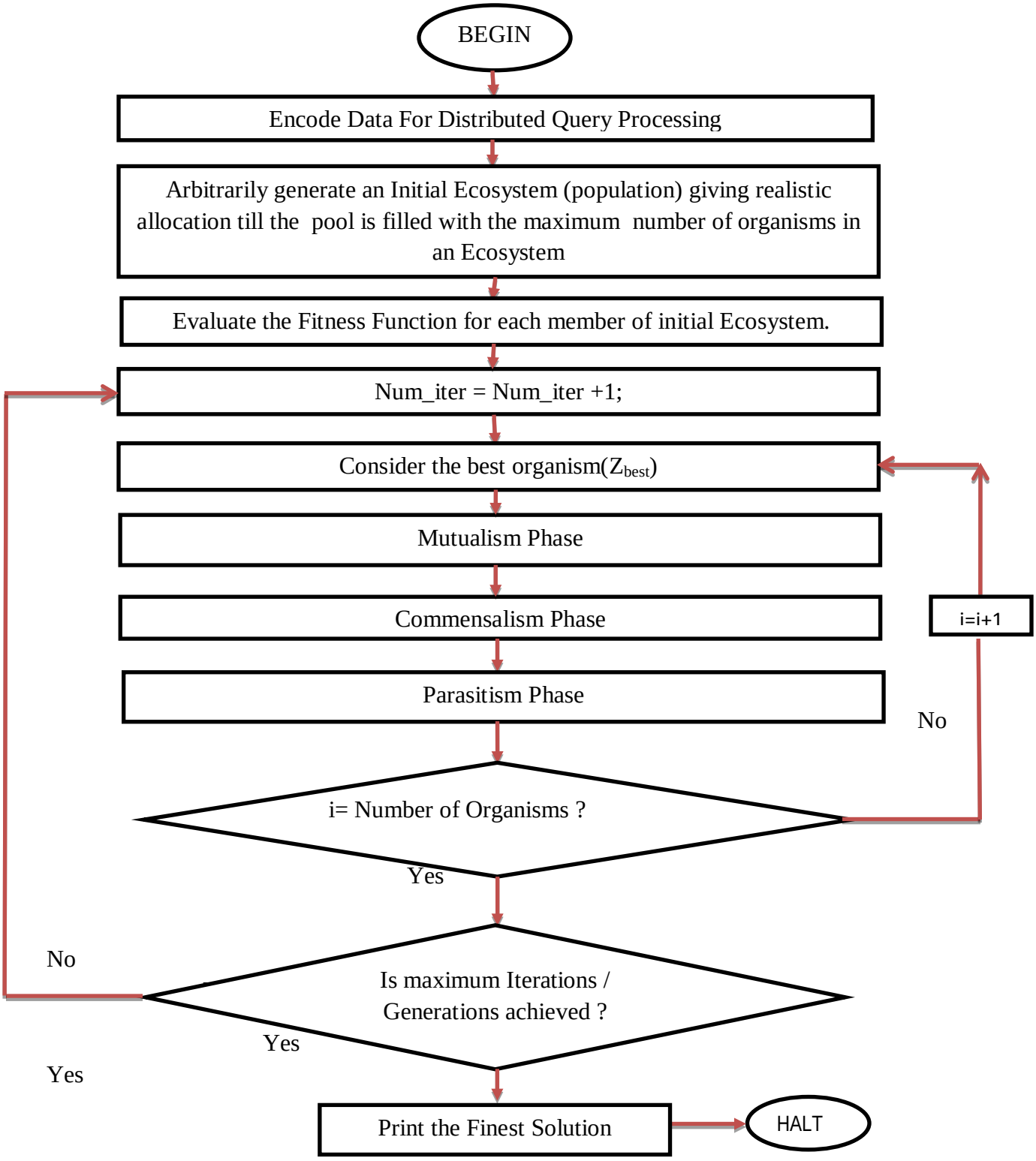

Fig. 2: SOS_DDQ Design for Distributed Queries

VI. CONCLUSION

We have proposed a new framework based on meta heuristic approach Symbiotic organisms Search for Distributed Query Optimization. A simulator has been designed SOS_DDQ in MATLAB that will work for distributed queries. In near future we will come up with its results and compare its working with other Evolutionary techniques in relevance to Distributed Environment.

REFERENCES

1. Apers P., HevenerR.,Yao S., "*Optimization Algorithms for distributed queries*",Transactions on Software Engg.SE(9):57-68, IEEE 1983.
2. Perrizo W., "*A Method for Processing Distributed Database Queries*", *IEEE Trans. on Soft. Engg.*,(se-10.).no.4:466-471,1984
3. Ceri, Pelagatti,"*Distributed Databases Principles & Systems*" McGraw Hill International Editions, 1985
4. William Grosky, Junping Sun, FarshadFotouhi "*Dynamic selectivity estimation for multidimensional queries*",springer, 1993.
5. Jorng H.,Jiunm W., Yiming H., Baw L., "A genetic Algorithm for set Set query Optimization in distributed System", IEEE,1996.
6. Rho Sangkyu, March Salvatore, "Optimizing *distributed join queries:Agenetic algorithm approach*", 1997.
7. Michael Gregory,"*Genetic Algorithm Optimization of Distributed Database Queries."IEEE,1998.*
8. Ahmet C,Ender S, **"***An Evolutionary genetic algorithm for optimization of distributed database queries*",IEEE,2009**.**
9. AreeratTrongratsameethong, JarernsriMitrpanont, "*Exhaustive Greedy Algorithm for Optimizing Intermediate Result Sizes of JoinQueries*", IEEE, 2009
10. NarasimhaiahGorla, Suk-Kyu Song, "*Subquery allocation in Distributed Database using GA*", JCS & T, Vol. 10, No.1.2010
11. M.Tamerozsu, PatricValduriez "*Principles of Distributed Database Systems*", springer, 2010
12. Fan Yuanyuan, MiXifeng,"*Distributed database System Query Optimization Algorithm Research*", IEEE, 2010.
13. Golshanara, Ladan, Seyed Mohammad, and Hamed Shah-Hosseini. "*A multi-colony ant algorithm for optimizing join queries in distributed database systems*", Knowledge and Information Systems (2013): 1-32.
14. Min Y., Doddy P., "*Symbiotic Organisms Search: A new Metaheuristic Optimization Algorithm*",Computer and Structures 139,98-112,Esevier,2014.