Multi-Objective Optimization-Based Query Optimizer For Distributed Database Management Systems

Ravneet Kaur, Er. Ajay Sharma

Abstract: A rapid improvement in multimedia applications increases the amount of data stored in databases. Therefore, it becomes impossible to store this large amount of data on these standard database management systems. Therefore, database administrators have moved to distributed databases. However, optimizing the query on these databases is still a challenging task. Meta-heuristics-based solutions suffer from poor convergence speed and sometimes may get stuck in local optima. Therefore, in this paper, a novel multi-objective non-dominated sorting genetic algorithm is proposed. The effect of query cost and energy consumption will also be considered. Extensive experimental results show the effectiveness of the proposed technique over the competitive approaches.

Keyword: Optimization, Distributed database, Multi-objective.

1. INTRODUCTION

Distributed database can be considered as a set of many databases related logically that are dispersed over a network of computer. Distributed Database Management System (DDBMS) belongs to the class of application software that handles distributed database and offers transparent access mechanism to several users across multiple sites by incorporating parallelism as well as modularity. Due to increased network traffic along with decreased efficiency, partitioning of information at various locations has become significant which usually lead to the development of a group of related databases that are known as the distributed database, where each location has their own local processing and storage abilities. This enhances effectiveness, stability, availability, and also modularity compared to that in traditional centralized databases system. In a DDS, database applications running at any of the system's sites should be able to operate on any of the database fragments transparently i.e., as if the data come from a single database managed by one DBMS. The software that manages a distributed database in such a way is called DDBMS.

1.1 TYPES OF DISTRIBUTED DATABASE SYSTEMS

Based on the types of DDBMS software installed on the various sites of distributed databases, it can be categorized as Homogenous DDBMS or Heterogeneous DDBMS

- **Homogeneous distributed system**
  Data is dispersed but all the servers execute similar Database Management System software in homogeneous distributed database. It is also of two types:
  - Autonomous – Every database operates at its own without depending on other database across the network.
  - Non-autonomous – Data is dispersed over identical sites and master or central database management system relates all updates made on data over the sites.

- **Heterogeneous distributed system**
  Several sites work under the control of different database management systems in heterogeneous distributed database. It is also of two types:
  - Federated – Heterogeneous database systems operate as a single database system due to their independent nature and are highly incorporated.
Un-federated – Heterogeneous database systems incorporate a central module for coordination by means of which access is made for databases.

Client/Server Distributed System
In client server architecture there are many clients and few servers that are inter linked in a network. A particular query is sent to one of the servers from a particular client. That query is solved by the server that is available at that time and sends the response. This architecture is easy to be implemented and executed since the server here is centralized server. To a server in distributed system, client can be connected directly or indirectly. Client is connected directly to server only when information is accessed from database of that server to which client is connected.

1.2 APPLICATIONS
1. Maintain track of data – Main aim of distributed database is to keep track of data allocation, partitioning and duplication.
2. Recovery of distributed data – DDBMS is capable of healing from any site failure such as from failure of communication between sites.
3. Management of replicated data – DDBMS make decision of which fragment of data replicated must be accessed.
4. Distributed Query processing – DDBMS access particular site and transfer queries and information across several sites through some network.
5. Management of distributed directory – All information related to data contained in database is stored in a directory. DDBMS maintains distributed directory.
6. Managing distributed transaction - All strategies for executing queries as well as transaction accessing information from multiple sites are maintained by DDBMS.
7. Security - Another main task of DDBMS is executing a particular query or transaction taking into account security of data as well as genuine rights of users.

1.3 QUERY OPTIMIZATION IN DISTRIBUTED DATABASE SYSTEM
Query optimizer tries to find out most efficient way to run a given query by seeing the possible query plans. Main aim of optimizer can be to search for optimal join order of relations or tables that are demarcated for input query. Cost of executing query that is expressed as fragment statistics, calculating cardinalities on table operations is needed as input for selecting optimal strategy. Thus the final output of global query optimization phase is an optimized query execution plan that is defined in terms of algebraic query.

Basic task of this thesis is to create query optimizer for distributed database which make use of the positive characteristics of ACO Algorithm combined with another algorithm to optimize big queries in distributed systems. ACO is entirely combined with another algorithm, in order to overcome the insufficiencies of ACO and improving its processing time.

1.3.1 Components of Distributed Query Optimization
Optimizer can be considered as a software module which optimizes the query based on three basic components namely search strategies, search space and cost models.

– Search Space
In this transformational rules are applied to generate sets of alternatives and some query execution plans for the input query where order of execution is different for processing that search. Join trees represent the query execution plans which contain operators on different join or Cartesian product. It can be indicated by a graph for query that is represented as G = (N, A) where A represents set of edges and N denotes set of vertices.

– Search Strategy
It means applying algorithms to search the space as well as to calculate the best query execution plan based on join sites so as to reduce the cost required for optimizing given query. Normally two strategies are used for solving problems of join scheduling for optimizing query. One strategy is Deterministic strategy that generates plans initiating at base relations and then joining multiple relations at every step till one finds the complete plans. Those plans which don’t give optimal outputs are ignored in order to reduce the cost of optimization. Another strategy is Randomized strategy which finds the optimal strategy across some certain points only. These techniques may not guaranteed optimal solution but they reduce cost factors. Random means some selective points are only considered without taking the entire problem.

– Cost Model
The aim of optimizing query is basically to reduce total resources cost utilised in distributed systems. Cost model for optimization includes cost functions that find the operators cost gives the size of output. Cost function can be given by either response time or total time. Total time consists of communication cost as well as local processing cost. Increasing the utilisation of resources by increasing system throughput will minimize the total time.
1.4 QUERY PROCESSING IN DISTRIBUTED DATABASE

Processing query in distributed database means to transfer high level query to effective execution plan written in low level language on a distributed database, that is viewed as a one database by its users that to convert relational calculus query to relational algebraic query.

• Decomposition of query: Distributed query is decomposed in the form of algebraic query. Actual task is to re-write query in normalized form, analyse it semantically (based on meanings), simplify it by removing the redundant predicates and then restructure it into an algebraic query.

• Data Localization: Here the input is algebraic query on global relations where its data is localized according to the information available on distributed relations. Two steps are involved in creating fragmented query in this phase. In the first step a construction program is used to map the input query with the fragmented data and replaces each relation by its fragments to create a fragmented query. The second step involves simplifying this query and restructuring into another appropriate query.

• Global Query Optimization: The main task of this phase is to generate an optimal query execution plan with proper sequencing of relational operators in the fragment query which greatly minimizes the defined objective cost function.

• Local Query Optimization: Here the optimal Join Order received which is output of above phase is executed locally at multiple sites using the Local Conceptual Schema of the Distributed Database.

1.5 ANT COLONY OPTIMIZATION ALGORITHM

Amongst the Evolutionary Algorithms, Ant Colony optimization Algorithm (ACO) is one of the most recent techniques implemented for Approximate Optimization. ACO is defined as a population based stochastic algorithm that is best suited to solve Combinatorial Optimization Problems. From an Artificial Intelligence perspective, ACO is considered to be one of the most successful strands of Swarm Intelligence because of its powerful and accurate decision taking ability that is based on the collective behaviour of its social agents (ants). From an Operation Research perspective, ACO belongs to the class of metaheuristics with the ability to search for “solutions beyond an upper level”. The key factor responsible for the success of ACO for discrete optimization problems is cooperation and communication of computational resources between its artificial ants through an indirect message passing mechanism called Stigmergy. ACO has been successfully applied to query optimization problems of centralized and distributed database systems but the algorithm is bound with the limitation of random initialization at the initiation phase because of which it attains slow converges speed towards optimal solutions. The social agents of ACO, i.e. ants display an unorganized behaviour at an initial phase because the concentration of artificial pheromone values is almost negligible at the initial state. These ants wander in all directions to search for optimal paths which delays the convergence time. However, as the algorithm progresses, the concentration of pheromone increases because of Positive Feedback Mechanism and the algorithm converges to generate an optimal solution. The main aim of this thesis is to create a query optimizer for distributed database that uses the positive characteristics of Ant Colony Optimization Algorithm integrated with another algorithm to optimize large queries in distributed databases. This integration of ACO with another algorithm is purely done to overcome the drawbacks of ACO and to improve its processing time.

![Figure 2: Flowchart of Ant Colony Optimization Algorithm](image-url)
algorithm can be run continuously and adapt to changes in real time. This is of interest in network routing and urban transportation systems.

The first ACO algorithm was called the ant system and it was aimed to solve the travelling salesman problem, in which the goal is to find the shortest round-trip to link a series of cities. The general algorithm is relatively simple and based on a set of ants, each making one of the possible round-trips along the cities.

It suffers from few limitations such as theoretical analysis is difficult, sequences of random decisions (not independent), Probability distribution changes by iteration, Research is experimental rather than theoretical, Time to convergence uncertain (but convergence is guaranteed!), therefore it is integrated with genetic algorithm to overcome its insufficiencies.

1.6 GENETIC ALGORITHM

The Genetic Algorithm (GA) is used to execute large queries with lesser joins. Initially Chromosomes population is created by making use of Genetic Algorithm, where each of the chromosomes is used to present a query plan. Every chromosome is made up of genes where the site of relation is represented by every chromosome’s gene. Gene’s value depicts the location in which the particular relation is situated is depicted by a gene’s value.

2. LITERATURE REVIEW

Matysiak et al. (2011) [1] a combinatorial optimization technique is proposed that is based on tabulated search to optimize large join queries in distributed system. To combine relations, optimizer chooses a particular join order. To compare the performance of tabulated search with different other search methods, several plan for executing query, several query sizes and different types of query are important.

Azarbad et al. (2011) [2] Image thresholding is a crucial procedure for processing an image and recognition of pattern. Multiple techniques of thresholding are planned in the paper. Hierarchical evolutionary algorithms can be seen as an alternative for conventional genetic algorithms. The technique proposed is dependent on PSO (particle swarm optimization) and this technique is unverified clustering method that is related to independent multiple level thresholding technique. Simulation results are then used to confirm and calculate the performance of the proposed technique.

Chen et al. (2011) [3] Proposed a technique in which research is done arbitrarily that exploits an similarity between the way in which metal first cools down and then freeze into energy crystalline structure and the exploration for a minimum is searched in a more general system required for computation. In order to optimise huge query, Chen developed Simulated Annealing method that is based On Graph-Based Approach.

Butey et al. (2012) [4] this paper proposed Peer-to-Peer Database Management Systems (PDBMS) for evolution. It brings up P2P technique for exploiting the power of existing distributed database management technologies. PDBMS proposed in the paper, are entirely independent and also any term of domination as central server and generating cost related global schema will be missing here. This paper also identify large number of appropriate issues in the overlap of database and P2P systems and also presents view of constructing peer-to-peer database management systems.

Dokeroglu et al. (2012) [5] proposed a technique focussed on collaborative behaviour of agents which results when corresponding individuals interact locally with each other. This technique is used for optimization due to appropriate or correct decision making capability of the social agents based on their cooperative behaviour. The main reason for the success of this technique is communication as well as cooperation of resources of computation among social agents through the process of indirect message passing.

Xiang H et al. (2013) [6] describes about the DDMS and processing of query in distributed system. Wide range of analysis along with progress is carried out to attain higher level accessibility to huge amount of data which is stored in DDBMS. It also mention requirement distributing huge scientific datasets distribution such as the Sloan Digital Sky Survey. Basic aim of the paper is to completely examine cross-joins query in distributed database through a scientific database that is distributed heterogeneously.

Mishra et al (2014) [7] query execution plans are evaluated by making use of particle swarm optimization and join operation. In order to arrange a swarm that is migrating across the search space for finding the best answer, a set of particles or social agents are used. Particles are denoted by the appropriate query plans based on relation schemes. The entire query processing plan may be created by optimizing entire query at the time of compilation. By assuming the arbitrary population of random speed and variables, function of evaluating particles that is indicated by the query plans in the relation, can be estimated.
proposed multi-colony ant algorithm, for the first time, so as to optimise join queries in distributed systems in which tables can be duplicated but they cannot be partitioned or fragmented. In this planned algorithm, 4 kinds of ants cooperate to generate an execution plan. Therefore each of the iteration has four ant colonies. In order to find the optimal plan, each ant performs decision-making. Two types of cost prototypes centred on total time as well as response time are utilised for the evaluation of the quality of the generated plan.

Adimi et al. (2014) [9] projected Particle Swarm Optimization (PSO) fuzzy multiple objective approach to generate optimum localizing and value arrangement of Unified Power Flow Controller (UPFC) within a control system for a large time.

Kumar et al. (2015) [10] proposed ACO algorithm reliable on technique named stochastic search. Movements of ant are modified from one relation to other. The above procedure repeats for a predefined quantity of duplications until the best query plans are produced as outcome. For relations in large numbers, ACO algorithm creates best query plan. But it has weaknesses such as high respond time and also optimization overheads are higher.

Ren K et al. (2015) [11] proposed an alternative technique named VVL (very lightweight locking) in order to cynical the effect of concurrency control for main memory systems that aims to eliminate all the overheads related to traditional operations of lock manager. Paper also introduces a protocol named SCA (selective contention analysis) that allows systems that implement very lightweight locking technique to attain maximum transactional output under large contention loads.

Mishra et al. (2015) [12] Today distributed queries are input on database scattered across the entire globe, which can make effective processing of queries to be challenging and also strategy is needed to produce optimal QEPs (query execution plans). This paper tries to search such optimal query execution plans by making use of parameter named TLBO (Teaching-Learner based Optimization). Teaching-learning based algorithm for optimization is considered to be the best one.

3. GAPS IN LITERATURE
It is noticed that many issues have been neglected in existing researches by conducting the review:
1. The effect of query cost and communication overheads are ignored in most of existing research on distributed databases.
2. The use of multi-objective optimization is ignored by most of existing researchers.
3. The use of multi-objective non-dominated sorting genetic algorithm to reduce query cost is also neglected in existing literature.

Problem Definition
A set of interrelated databases, known as distributed databases, is placed on several servers or sites, with fast advancements in data in current systems. It increases availability, dependability and modularity as compared to that of existing database system. In today's era, every user needs access to the databases. According to a distributed allocation plan, the data may be repeated at several sites in distributed databases. Today, each and every user wants access response to the query of a user. Here significant role is played by query optimization to handle the distributed queries. Thus optimising query has been a crucial subject for the Distributed Database Management System (DBMS). The progress in computer hardware, software, networks, storage and protocols have altered the view of business demands by making the processing of distributed database, feasible and useful option.

Consequently, the impact of efficient query processing is increasing in a large number of applications. The query optimization is a critical subject for database management systems, whose purpose is to define an appropriate execution plan for the user's query. The cost of each plan should be produced due to more than one plan for such a query, which is significantly dependent on the amount of participation and data transfer between the sites. Due to the increasing number of joins and the number of query execution plans, the query optimization problem is an NP-hard problem. Therefore, many heuristic and meta heuristic approaches are proposed to solve this problem. To overcome the issues associated with the existing techniques, a new multi-objective non-dominated sorting genetic algorithm based query optimization technique is proposed. The effect of query cost and communication overheads will also be considered. The use of non-dominated sorting genetic algorithm can find optimistic query in order to reduce the query cost and the proposed technique will be compared to existing techniques based upon certain performance metrics.

4. PROPOSED ALGORITHM

Step 1: Start
This step indicates the start of our proposed technique.

Step 2: Initialize Database System
To start the system we use appropriate command or directory. This step initializes database system based on isolate or replicate database.
**Step 3: Generate query optimization**

This step will generate optimizer. Query will fetch data from the sites and we tune the optimiser considering several factors.

**Step 4: Apply multi-objective non-dominated sorting genetic algorithm**

In this step, we apply all the three operators of genetic algorithm. Then multi-objective non-dominated sorting is done.

**Step 5: Evaluate parameters**

This step calculates all the parameters and selects the best parameters.

**Step 6: End**

This will terminate the algorithm.

---

5. **RESULTS AND DISCUSSION**

Existing and proposed techniques are compared based on following factors or metrics.

1. **QUERY COST** is what optimizer thinks of how long your query will take (relative to total batch time). The optimizer tries to choose the optimal query plan by looking at your query and statistics of your data, trying several execution plans and selecting the least costly of them.

![Query cost analysis](image1)

**Figure 5.1: Query cost analysis**

Figure compares the first parameter namely query cost for the given queries of both existing and proposed techniques. The figure shows that results of proposed technique are better than the existing technique.

2. **ENERGY CONSUMPTION** is the amount of energy or power used. It refers to all the energy used to perform an action, manufacture something or simply inhibit a building. Energy consumption is measured in terms of joules.

![Box plot of energy consumption](image2)

**Figure 5.2: Box plot of energy consumption**

Figure shows the box plot of energy consumption for both the existing and proposed techniques, thereby comparing energy consumed by both the techniques. It can be viewed that results of propose technique are better than that of existing technique. Energy consumed by proposed technique is much less than that of existing technique.

6. **CONCLUSION**

To optimize the distributed queries in an efficient manner, a novel multi-objective non-dominated sorting genetic algorithm is proposed. A novel multi-objective fitness function is also designed to handle this issue. The proposed technique is designed and implemented on MATLAB tool.

---
Extensive experimental results show the effectiveness of the proposed technique over the competitive approaches. It has been found that the proposed technique not only reduces the query cost, but also reduces communication overheads and energy consumption at the same time. In near future, we will consider the effect of security attacks on the distributed attacks to design an attack resistive proposed technique.

REFERENCES