HEURISTIC APPROACH FOR OPTIMIZATION OF LOAD BALANCING IN DISTRIBUTED SYSTEMS

***Sheo Kumar, #Sugandha Chakraverti, $Dr.S.C. Agarwal, @Ashish Kumar Chakraverti**

*Ph.D. Scholar, MBU, Solan, HP

#Ph.D. Scholar, MBU, Solan, HP

$Director, KCC, Greater Noida

@Assistant Professor, IEC-CET, Greater Noida

ABSTRACT

A distributed system is one in which components located at networked computers communicate and coordinate their actions only by passing messages. Now days Distributed system face different type of problems. One of those problems is load balancing in distributed system. A lot of improvement in hardware is being done. But in this paper we are representing a software technique in which a heuristic is designed to optimize the query processing with the help of cost optimization. Cost of query processing is depending on different parameter. With the help of all parameter we have design a heuristic (A mathematical equation) which helps to have best query sequencing and designing to have optimum load balancing in distributed environment. For the representation of query operation sequencing we use the concept of operator tree.

Keywords:-*Distributed system, Heuristic, Operator tree, load balancing*

INTRODUCTION

Distributed System:

A distributed system is one in which components located at networked computers communicate and coordinate their actions only by passing messages. It has following characteristics:

- **Concurrency of components**
  The capacity of the system to handle shared resources can be increased by adding more resources to the network.
- **Lack of global clock**
  Communication between remote nodes is possible only with the help of messages through a network.
- **Independent failures ofcomponents**
  Faults in the network result in the isolation of the computers that are connected to it, but it does not mean that they stop running. In fact the programs on them may not be able to detect whether the network has failed or has become unusuallyslow.

The sharing of resources is a main motivation for constructing distributed systems. Resources may be managed by servers and accessed by clients or they may be encapsulated as objects and accessed by other objects.

The challenges arising from the construction of distributed system are the heterogeneity of its components, openness which allows components to be added or replaced, security, scalability.

MOTIVATION

The mindset behind the development of distributed systems was the availability of powerful microprocessors at low cost as well as significant advances in communication technology.

When a group of people work together, there is generally a need to communicate with each other, to share data, and to share expensive resources. This requires interconnecting computers and resources. Total computing power available in such system can be enormous. Other significant advantages of distributed system over traditional time-sharing systems are asfollows:-

- Resources sharing
- Enhancedperformance
- Improved reliability and availability
- Modular Expandability

Distributed database :

A distributed database is a collection of data which belong logically to the same system but spread over the sites of computer network. It emphasizes two important aspects:

1. Distribution: Data are not resident at the same site so that we can distinguish a distributed database from a single, centralizeddatabase.

2. Logical correlation: Data have some properties which tie them together, so that we can distinguish a distributed database from a set of local database or files which are resident at different sites of a computernetwork.

Problem Definition:

In a distributed transaction processing system it is generally found that when number of sites tries to fetch their query from their remote sites then load on the system increases. It is felt that the query executing on the server facing problem to execute it faster, it happens due to capacity of the query fetching algorithms. The problem arises due to heterogeneity in the components of queries because of addition, replacement, security and scalability among queriescomponents.

As the user sitting at different location and accessing the network, a strategy is required to find out the location of data and fast processing algorithm is needed. The query processing consists of decomposing the queries and data localization which involves identification of fragments and generation of the query operator tree. File allocation with respect to multiple files on a multiprocessor system requires global optimization model to minimize overall processing cost under the constraints of response time and storage capacity with a fixed number of copies of each file.

For executing the queries faster, data transfer cost also should be minimum. This can be achieved by optimally allocating the data to the sites of the distributed database system. Optimal allocation of fragments is a complex problem because of mutual interdependency between allocation scheme and query optimization strategy.

In distributed database environment the communication cost is a major cost in query processing. Thus it requires some strategy to minimize the amount of data transmitted over the network.

REVIEW OF EXISTING LITERATURE

A distributed database is a collection of multiple logically interrelated databases distributed over a computer network, and a distributed database management system is a software that manages a distributed database while making the distribution transparent to the user.

In DDBS, the processing of a query involves data transmission among different sites via computer network. The DDBMS are important because they provide certain performance, reliability and availability advantages over a centralized system.

In DDBS a relation is decomposed into fragments, each being treated as a unit, permits a number of transactions to execute concurrently. The fragmentation of relation permits the parallel execution of a single query by dividing it into set of sub queries that operate on fragments.

The development of a distributed database system requires effective solutions to many complex and interrelated design problems. The cost dependencies between query optimization and data allocation on distributed systems are well recognized but little understood. The problem of fragment allocation to various sites on network is a fundamental aspect DDBS. It arises whenever the database relation are decomposed into fragment and distributed over various sites of the network. The distribution of the fragment may be fully replicated (i.e. database exists in its entirety at each site), non-replicated (i.e. each fragment is stored at exactly one site), and partial replicated (i.e. some fragments of the database may be replicated whereas other may not).The allocation problem deals with the questions of allocating fragments to sites so as to minimize the data transfer cost and query executioncost.

RELATED PREVIOUS WORK

In DDBS, a query requires data to be accessed from one or more sites. The cost of executing the query depends on the location of the query as well as the data. Specifically, the data locality of a query determines the amount of data transfer incurred in processing the query; the higher the data locality the lower the data transfer cost. This gives the rise to the requirement of and efficient data allocation policy. The aim of which is to increase the data locality [AHMA02].

The Query optimization in DDBS has multiple dimensions. Comprehensive surveys of state-of –the- art research on distributed query optimization [ONO 90, IOAN 90, GRAE 93, SHEK 93, KEMP 93, LANZ 93, GALI 94, VANC 96, STEI 97,VANC 98, DONA 00,CYRU00,MOUR04]anddistributeddataallocation [CHIU90,REVE90,RAM91, LIN94, KARL95,KARL97,TAMH98, HUAN01,AHMA02]exist.Themajorityof research in one area has assumed a given solution for the other. Early research provides guidelines for data replication and allocation based on optimizing query strategies. The author analyzes the interdependencies among data partitioning, data allocation, query optimization, concurrency control, and network design in local multi access distributed systems.

A methodology for distributed database design includes an iterative integration of complex problems such as relation partitioning, data allocation, query optimization, and load balancing.

The combined distributed query optimization/data allocation problem has an immense search space for optimal solution. Both optimization problems individually have been proven NP-hard.

Various aspects of the Query optimization in DDBS have been covered by researchers. The problem has been tackled with various approaches with different schemes considering different aspect such as communication cost, throughput, response time, storage capacity, link capacity, data transfer cost, reliability, availability, processing cost etc. Overall objective(s) of the study has been to optimize the one or more measure(s) of effectiveness of queries inDDBS.

A thorough related literature reveals that the previous work can be broken down along following major lines:

Commonly Adopted Approaches

(1) Mathematical programming Approach

(2)Genetic algorithmApproach

(3) Petri net modelingApproach

Consideration of differentAspects

Design of DDBS is extremely complex. There are four aspects that must be considered:

(i) *Communication network consideration:* Deals with the location of nodes, allocation of computer resources, network topology and selection of link capacities.

(ii) *Data allocation consideration:* Deals with the determination units of data to allocate i.e. file fragmentation

(iii) *Operating strategies consideration:* Deals with query optimization and concurrency control mechanism e.g. determining which copy or copies of data to access, where to process the data, how to route the data, locking and commit protocol .

(iv) *Local database design consideration:* Deals with record structures, record placement, secondary indexes andprocessing.

Query Optimization Method in DDBS

The following are the commonly used distributed query optimization strategies e classes
(i) Local Optimization of Semi joins (LO)
(ii)Join Sequence Optimization (JSO)
(iii)Relation Semi joins on Broadcasting Networks (RSO)

(i) Local Optimization of Semi joins(LO)

The query optimizer of SDD-1 is among the first to apply semi joins to distributed query processing. The optimizer evaluates the benefit and cost of all candidate semi joins, performs the most profitable one, and update the cardinality of the relation reduced by this semi join accordingly. This procedure repeats until there is no profitable semi joins available. This approach is greedy and sub optimal.

A general version of the above approach is that the query optimizer computes for each relation R_i, instead of the most profitable semi join, the most profitable set of semi join to reduce R_i. Since all semi joins to R_i are considered at the same time, local optimality (with respect to R_i) can be attained. The solution obtain is local optimal in that only the reduction of one relation taken into account at a time. Such a problem is hence called local optimization of semi joins. This problem was studied earlier and heuristic was developed, its complexity is unanswered.

(ii) Join Sequence Optimization (JSO)

Using the join sequence method for distributed query processing, some joins are performed locally first, and the resulting relations are then sent to the assembly site. Join sequence optimization is formulated as graph problem.

(iii) Relation Semi join on Broadcasting Networks (RSO)

The advantage of the broadcasting property of local area networks, the relation semi join method can be applied to improve the query processing. Using this method, a distributed semi join is accomplished by sending the entire relation, rather than join attribute, to its recipient. The advantage is that multiple stations may receive the relation and apply different semi join at the same time. Further the assembly site also receives the relation broadcast, and each relation thus needs to be scanned only once. To optimize the transmission cost, the query optimizer has to determine broadcasting order of the relation.

ISSUES IN THE QUERY PROCESSING IN DISTRIBUTED ENVIRONMENT

- Idea:
  This is just an extension of centralizedqueryprocessing. (System R* et al. in the early80s)
- What isdifferent?
  - extend physical algebra: send&receive operators
  - resource vectors, network interconnectmatrix

- caching andreplication
- optimize for responsetime
- less predictability in cost model (adaptivealgos)
- heterogeneity in data formats and datamodels

ISSUES:

Distributed Query Plan

Cost

Response Time

PROBLEMS RELATED TO QUERY PROCESSING IN DISTRIBUTED ENVIRONMENT

- Requirement of Adaptive Algorithms
  - Deal with unpredictable events at run time
  - delays in arrival of data, burstiness of network
  - autonomity of nodes, change inpolicies
- Example: double pipelined hashjoins
  - build hash table for both input streams
  - read inputs in separatethreads
  - good for bursty arrival ofdata
- re-optimization at runtime
  - monitor execution ofquery
  - adjust estimates of costmodel
  - re-optimize if delta is toolarge

Heterogeneity

- Use Wrappers to „hide“heterogeneity
- Wrappers take care of data format,packaging
- Wrappers map from local to globalschema
- Wrappers carry outcaching
  - connections, cursors, data,...
- Wrappers map queries into localdialect
- Wrappers participate in queryplanning!!!
  - define the subset of queries that can behandled
  - give cost information, statistics
  - „capability-based rewrite“ (HKWY, VLDB1997)

PROPOSED RESEARCH WORK

A Query Driven Site Capacity Constrained Heuristic Scheme for Optimal Allocation of Fragments in Distributed Database Systems:

NOMENCLATURE & DEFINITIONS

f	Number of fragments in the Distributed database system
s	Number of sites in the Distributed databasesystem
c	Capacity of each site
N_S_I	Number of site Involved
NSI	A counter forcomparison
DTC(,)	A matrix of data transfer cost among the sites. DTC(i,j) For each i= 1 …f, j= 1 …s
Capacity( )	An array , which contains the storing capacity of eachsite
Alloc( )	An array, which signifies the boolean information of fragment allocation having value 1, if a fragment is allocated to site j, and 0,otherwise
COST(,0)	An array contain the index value of matrix DTC(,), which signifies the fragmentnumber
COST(,1)	An array contain the index value of matrix DTC(,), which signifies the site number
COST(,2)	An array contain the actual costvalue
ASSIGN ()	An array, storing the cost values of allocatedfragments
SITE()	An array storing those site numbers on which the fragments are allocated
FRAG()	An array storing the indices of allocatedfragments

ASSUMPTIONS

The proposed technique is based on the following assumptions:

1. The number of sites and number of fragments areequal.
2. The number of site involvement (no. of fragments / Capacity of each site) in storing all the fragment must be equal to the integervalue.
3. The fragment, once allocated to a site, can not be allocated to any othersite(s).
4. Replication of fragments is notallowed.
5. The sites are fullyconnected.
6. Each fragment is allocated to at least onesite.
7. No site allocates more fragments than the maximum permissiblenumber.
8. The inter site distance assumed to beunity.

ALLOCATION PROBLEM

Formally the allocation problem can be stated as follows:
Let F = {F_0 , F_1,….,F_{n-1} }be a set of fragments and S = { S_0 , S_1,….,S_{m-1} }be set of n sites connected by communication network on which a set of applications Q={q_0 , q_1,…,q_{g-1}} are running. A link between two sites S_i and S_j has a positive integer $C_{i,j}$ associated with it,

giving the cost of a unit data transferred from site S_i to site S_j. If the two sites are not directly connected by a communication link then the cost for the unit data transferred is given by the sum of the costs of the links of a chosen path from site S_i to S_j. Each query q_g can be executed from any site with a certain frequency. Let $FREQ_{i,j}$ be the frequency with which query q_i is executed from site S_j . These frequencies of execution of queries, at all sites, can be represented FREQ of order mxn. A query may access one or more fragments.

Query Execution Strategy

The optimal orderings of binary operations is based on a query execution strategy in Distributed databases. A query execution strategy can be:

1. *Move Small:* If a binary operation involves two fragments located at two different sites then ship the smaller fragment to the site of the largerfragment.
2. *Query Site* : Ship all the fragments to the site of origin of query and execute the query.

Here, the objective of the data allocation is (i) to minimize the total data transfer cost to process all the queries by using „Query Site" query execution strategy (ii) to maximize the locality of the fragments for executing the queries (iii) to incorporate the query execution strategy when a query needs to access fragments from multiple sites and reduce the total data transfer cost to process all the queries.

Evaluation of Query: Operator Tree

An operator tree is a tree in which a leaf node is a relation stored in the database, and a non leaf node is an intermediate relation produced by a relational algebra operator. The sequence of operation is directed from the leaves to the root, which represents the answer to the query.

COST FUNCTIONS OF DATA TRANSFER IN DDB

We define an allocation of fragments, as the optimal allocation, that optimizes the total data transfer cost within the constraints. So, it is desirable to know the size of data for every fragment that may be required by any site for processing a query. The fragments located at different sites may be of different sizes. Thus, in the fragment dependency graph for „Query Site" query processing strategy, the size of the data of a fragment required by query site does not vary with the location of other fragments, since there is no dependency between the fragments accessed by the query.

Let $r_{i,j}$ be defined as size of data of F_j needed to be transported to the site where query $q_i$is initiated. The corresponding matrix R is of order gxn. Let frequency of query $q_{i,}$ initiated at site $S_{j,}$ be $FREQ_{i,j}$. And let the query q_i request for the fragment Fk and each request require $r_{i,k}$ amount of data transfer from the site where F_k is located. The amount of data, need to be transferred from the site where fragment F_k is allocated to the site S_i where the query is initiated, is given by matrix FREQ of ordermxn.

Thus, the amount of data transfer for query q_i from site S_j can be expressed as

$$ADT = \sum_{j=0}^{n-1} FREQ_{i,j} * r_{i,j} \quad (1)$$

The total data transfer cost is:

$$ADTC = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} C_{site(Fk),i} * ADT_{i,j} \quad (2)$$

The communication cost $C_{i,j}$ represents the communication in terms of bytes transferred, between the site (F_k) and site(F_i).

THE PROPOSED METHOD AND ALGORITHM

A fragment is allocated to a site in such a way that extensive data transfer cost is avoided and the capacity of the site suit to the execution environment of the system. The proposed algorithm involves stepwise refinement of matrix of Data Transfer Cost (DTC (,)) among the sites, an array storing those site numbers on which the fragments are allocated (SITE ( )) and an array containing those fragments numbers, which get allocated (FRAG ( )) during allocation process of m fragments to n sites. These fragments are assigned to the sites in such a way that the total data transfer cost remains minimum.

A brief description of the proposed method is as follows:

1. Arrange the DTC (,) in ascending order and store in an array COST (, 2). Their row and column indices containing into an array COST (, 0) and COST (, 1) respectively.
2. Begin with the first element of the COST (, 2), COST (, 0) COST (, 1) and assign to arrays ASSIGN ( ), SITE ( ) and FRAG ( )respectively.
3. Proceed with next values of COST (, 0) and COST (, 1) that gives rise two cases:
   Check the SITE ( ) and FRAG ( ) values corresponding to the previously assigned values. If COST (, 1) differs the previousvalue
   And COST (, 0) equals the previous value (it indicates that capacity is available and fragment get allocated). Store these values to arrays ASSIGN ( ), SITE ( ) and FRAG ( ) respectively, and update the information regarding site and fragment accordingly.
   Check the SITE ( ) and FRAG ( ) values corresponding to the previously assigned values. If value of COST (, 0) and COST (, 1) differ than their previously assigned values (it shows that capacity is available, allocation is not made yet) store these value to ASSIGN ( ), SITE ( ) and FRAG ( ) and update the information regarding site and fragmentaccordingly.
4. When both of the above cases fail then search a site in the array SITE ( ). If the site indices match with the corresponding indices of COST (, 0), check its capacity, if available, allocate the fragment to the site and update the information regarding SITE ( ) and FRAG ().

5 Continue the above process till all the fragments areallocated.

6 Get the total data transfer cost by summing the values in the array ASSIGN ().

CONCLUSION AND FUTURE WORK

In this paper we have proposed a mathematical model and an algorithm of heuristic approach for the optimization of load balancing in distributed system. Mathematically it is shown that up to some extant this technique may optimize the load balancing in distributed environment.

In future we will try or anyone can try to simulate this approach in suitable simulation tools to help for real application environment.

REFERENCES

1. Fronto N. Afrati and Jeffrey D. Ullman "Optimizing multiway join in Map-Reduced Enviorment" IEEE Transaction on Knowledge and Data Engineering" VOL.23, No.9, SEP2011.

2. Lijiang Chen, Bin Cui, Hua Lu "Constrained Skyline Query Processing against Distributed Database" IEEE Transaction on Knowledge and Data Engineering" VOL.23, No.2, FEB2011.

3. Crlos Ordonez "Optimization of Linear Recursive Queries in SQL" IEEE Transaction on Knowledge and Data Engineering" VOL.22, No.7, SEP2010.

4. Sangeetha Seshadri, Vibhor Kumar, Brian Cooperand Ling Liu " A Distributed Stream Query Optimization Frame work Through Integrated Planning and Deployment" IEEE Transaction on Parallel and Distributed System, VOL.20, No.10, OCT2009.

5. R. Srinivasa Rao, S. V. L. Narasimham "A New Heuristic Approach for Optimal Network Reconfiguration in Distribution System" International Journal of Engineering and Applied Sciences 5:1 2009

6. Jean Ghanem "Implementation of Load Balancing Policies in Distributed Systems" The University of New Mexico Albuquerque, New Mexico June, 2004

7. Elsadek/Wells *The International Journal of Computers and Their Applications,* Vol. 6, No. 1, March 1999. **"**A Heuristic model for task allocation in heterogeneous distributed computingsystems"

8. Proceedings 21st European Conference on Modeling and Simulation Ivan Zelinka, Zuzana Oplatková, Alessandra Orsoni ©ECMS 2007 ISBN 978-0-9553018-2-7 / ISBN 978-0-9553018-3-4 (CD) "A GENETIC ALGORITHM FOR PROCESS SCHEDULING IN DISTRIBUTED OPERATING SYSTEMS CONSIDERING LOADBALANCING**"**

9. "Effective Load Balancing Strategies in Distributed Systems" Justin E Clareburt Department of Computer Science and Electrical Engineering University ofQueensland

10. Aly E. El-Abd, Mohamed I. El-Bendary "A Neural Network Approach for Dynamic Load Balancing In Homogeneous Distributed Systems" Proceedings of The Thirtieth Annual Hawwaii International Conference on System Sciences ISBN 0-8186-7862-3/97 © 1997IEEE

11. Query processing Tore Risch Information Technology Uppsala University 2010-02-17

12. Cem Evrendilek, Asuman Dogac " Query Decomposition, Optimization and Processing in Multidatabase Systems" Software Research and Development Center Scientific and Technical Research Council of Turkiye Middle East Technical University (METU) 06531 AnkaraTurkiye.

13. Ahamad Ishfaq, Karlapalem K, Kwok V. K., So S. K., "*Evolutionary algorithms for allocating data in distributed database systems*", Kluwer Academic Publisher, Distributed and Parallel Databases, 11, 5-32,2002.

14. Barker K. and Bhar S., "*A Graphical approach to allocating class fragments in distributed Objectbase Systems*", International Journal of Distributed and Parallel Databases, Kluwer Academic Publishers, Vol. 10, 207-239,2001.

15. Blankinship, R., "*Query Optimization and Data Allocation on Distributed Database Systems : An Integrated Solution Approach*", Ph. D. Thesis,1991.

16. Chaudhuri S and Shim K, "*Optimization of queries with user defined predicates*" in proceedings of the 22nd Int. Conf. on VLDB, Bombay, India, 1996.

17. Chiu G. M. and Raghvendra C.S., *"A model for optimal database allocation in distributed computing systems",* IEEE, INFOCOM"90, pp. 827-833 ,1990.

18. Cyrus Shahabi, et al. *"A probe based technique to optimize join queries in distributed Internet databases*", Knowledge and information Systems, pp. 373-385,2000(2).

19. Donald Kossman, *" The state of the art in distributed query processing*", ACM Computing surveys, 32 (4) ,pp. 422- 469, Dec2000.

20. Gallindo-Legaria, C. Pellenkroft A. and Kersten M. *"Fast, randomized join-order slection – why use transformation* ?", In proc. of the 20th Int. Conf. on VLDB, Berkley, CA, 85-95,1994.

21. Graefe G. and Mckenna W. J. *"The volcano optimizer: Extensibility and effeiceint search*", In proceedings of the 9th Int. Conf. on Data engineering, IEEE computer society, Washington , DC, 209-218, 1993.

22. HuangY-FuandChenJ-Her,*"Fragmentallocationindistributeddatabase design"*, Journal of Information Science and Engineering, 17, 491-506, 2001