Sci. J. Fac. Sci. Menoufia Univ. Vol. VIII (1994)

A NEW HEURISTIC ALGORITHM FOR QUERY PROCESSING IN DISTRIBUTED DATABASE ENVIRONMENT

By

Dr. Hany M. Harb*, Prof. Nabawia Al-Ramly and Ahmed Zayed****

* *Azhar university, faculty of engineering, computers and systems engineering department.*

** *Menoufia University, faculty of science, computer department.*

ABSTRACT

This paper introduces a new algorithm for query processing in distributed database environment. The conventional approaches that reduce the amount of data transmission, are mainly applying a sequence of semijoins to reduce the relation, Ship the intermediate results to the final site, and perform the final operations. The query processing algorithm proposed in this paper is based on performing a parallel execution of the semijoins or distributive joins. The aim of this method is to minimizing the response time for processing the queries in distributed database.

keywords: semijoin, distributive join, parallel execution, response time.

1- INTRODUCTION

Query processing is the mapping of high level query in relational calculus language into a set of low level query in relational algebraic operations. Database Management System (DBMS) is

responsible for this mapping without need to specify the procedure.

Where the Distributed Database System (DDBS) is a collection of multiple, logically interrelated database which distributed over the computer network. The query processing problem is affected by the design of DDBS and also by directory management which defined as information about the database. The design of distributed database depends upon the level of sharing (data plus program sharing, data sharing or no sharing), access pattern (static or dynamic) and level of knowledge (partial or complete information).

There are two methods of DDBS design, top down and bottom up. Each of these methods is based on several factors namely fragmentation, locations and replication.

Fragmentation is a subrelations of the base relation which satisfies correctness (completeness, disjoint and reconstruction). Fragmentation approaches are horizontal, vertical and hybrid. Horizontal fragmentation is the partition of the base relation into a set of rows. But, vertical fragmentation is the partition of the base relation into a smaller subrelations. While, hybrid fragmentation is a combination of horizontal and vertical fragmentation. There are many techniques for proving the correctness of the fragmentation, such as **PHORIZONTAL** and **PARTITION** [1].

Allocation is the place at which the base relation as well as the fragmented subrelations are found. Thus, allocation is the main issue of the DDBS design [2] For the replication, it is the number of copies of the base relation or the fragmented subrelations (fully replicated, partially replicated or nonreplicated).

The data directory includes the information about the base relations and the fragmented subrelations that found in the database system. The data directory is either a global directory or distributed into many local directories that include information about the relations which exist in their sites. The global and local directory may be replicated (fully replicated, partially replicated or nonreplicated). Also, the allocation of global directory and local directory is the main issue of the directory management.

The query processing in centralized database system consists of three processing phases which are **DECOMPOSITION** Phase, **OPTIMIZATION** Phase and **EXECUTION** Phase. The decomposition phase consists of rewriting the query in normalized form [3], analyzing the normalized form [4], simplification and elimination the redundancy and reconstruction of the algebraic operations. The optimization phase is either static optimization or dynamic optimization.

The distributed database system is an extension of the centralized database system. Subsequently, the query processing in distributed database system is an extension of the query processing in centralized database system. The essential difference which characterizes the query processing of distributed database, is the **LOCALIZATION** Phase, that is not applicable in centralized database. The localization phase translates the algebraic operations applied on global relation into minimal algebraic operations expressed on physical fragmentation.

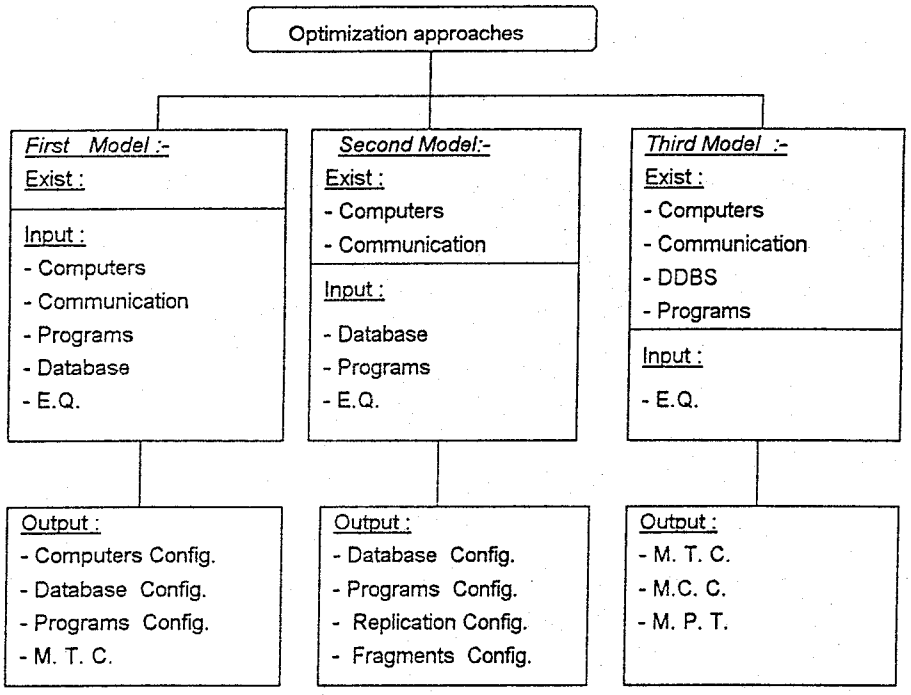
The query processing optimization is a common important phase in both centralized and distributed query processing. This phase

consists of local, reduction and final layers. At the **local** processing layer all local operations (selection, projection) are processed. At **reduction** layer set of reduction operations are applied to reduce the amount of data. While, at **final** processing layer all intermediate relations contributed in the final result of the query are transmitted to the final site.

2- SURVEY

Query processing optimization in distributed database system has three different approaches as shown in Fig.(2).

Through the first approach, for a given sets of computers, communications, databases, programs and estimatated queries three main objectives could be achieved, the first one is the optimal distribution configuration of the computers (network topology), programs and databases. The second one is the best selection of communication mode and third one is the optimal cost function(response time or communication cost or both of them). Different models have been adopted to achieve these objectives. Chen and Akoka [5] have developed a complete model for optimizing a distributed information system, which considers the allocation of databases and programs, communication line capacities and return flow information. This model is a generalized model.



* E.Q.	: Estimated Queries.
* DDBS	: Distributed Database.
* Computers Config.	: Computers configurations.
* Database Config	: Database configurations.
* Programs Config.	: Programs configurations.
* Fragment. Config.	: Fragmentation configurations.
* Replication Config.	: Replication configurations.
* M.P.T.	: Minimum Processing Time .
* M.C.C.	: Minimum Communication Cost .
* M.T.C.	: Minimum Total Cost.

Figure(2): Distributed query processing optimization approaches.

Epstein [6] provided several algorithms according to the network topology and the objective function of the system. The optimal distribution of the system can be determined. While, Gavish and Pirkul [7] provided a heuristic and optimal solution algorithm for locating computers and allocating databases in distributed computer system.

Through the second approach, for a given sets of databases, programs and estimted queries, with the existence of computer configuration (network topology) and communication mode objectives such as the optimal configuration of distributed database (fragmented or replicated), the programs (distribution and replication) and optimal allocation of both could be achieved. Chu [8] developed a model for allocating multiple file copies in computer network objecting the minimization of overall operating costs, considering the memory size restriction in computer. Morgan and Liven [9] studied variations of basic model in which the allocation of the data files and programs were separated.

Through the third approach, for a given sets of estimated queries with the existence of computer configuration (network topology), communication mode, distributed databases and distributed programs, the objectives are the minimum response time (elapsed time from entry the query until get the result), or minimum communication cost or minimize the total cost (Communication cost + resources cost + database cost) could be achieved. Epstien [6] provided some algorithms according to the network topology to minimize either the response time or the communication cost. Chu and Hurley [10] developed a model for determining the optimal query processing policy based on selecting the sequences and the sites for executing a set of subquery to yield minimum operating cost. Yu et al. [11] developed an

algorithm that handling the query optimization in fragmented relation by processing the queries with the semijoin algorithm. Wang et. al. [12] provided a method for parallel execution of distributed query operations. This method optimizes the response time for processing of distributed queries. Epstein et. al. [13] provided a heuristic query optimization algorithm. This algorithm minimizes the response time by "Equalizing" the data size at each site (the same amount of data at each site which requires the same amount of processing time). But, the model is not realistic. Apers et. al. [14] adopted heuristic algorithm which allowed applying a parallel semijoin operations. But, it suffers from the sequential execution of semijoins. The objective function of this model was to minimize the total cost.

This paper proposes a method for generating a distributed query processing policy which provides a better response time. There are various ways for optimizing the response time by using parallel transmission techniques. These techniques use semijoin or distributive join as relation reducer. After reduction have been done, the reduced relation are transmitted to the final sites for completing the final processing.

3- QUERY OPTIMIZATION PROCEDURE

The query optimizer is responsible for ordering the database relations, accesses path for database, performing database operations as well as performing the intermediate relations operation. In the following section a trial to give more details to different layers of query optimizer method has been made.

3.1 Local Processing Layer

This section describes the general steps for handling the query entry to represent it in a sequence of equivalent algeberic operations. The query is expressed in a relational complete language [15] such as relational calculus language. Lexical and syntax errors are checked. If there are no such errors, the query is transformed into normalized form (conjunctive normal form or disjunctive normal form). The query in normalized form is analyzed to check the semantic errors (incorrectness) by using the relational connection graph or attribute connection graph [16]. If the query is incorrect then the query is either rejected or the disconnected subquery is removed from the query or infer the missing subquery. If the query is correct until this point, the query is simplified for better performance. Transformation rules can be used to simplify the query expression. The three types of simplification are the elimination of redundancy, transitivity and integrity rules. Where, redundancy, which leads to duplicate the work, is eliminated by applying the rules of boolean algebra. A query qualification may be transformed into an equivalent query qualification for more efficient processing. The transformation done by applying the transitivity rules. Semantic integrity rules can be useful to simplify queries. The simplified query is then transformed into a sequence of a relational algebra operations using graphical method as relational algebra tree as shown in Fig.(2). Some trees may provide much performance than others. Important factors for optimization are the ordering of relational operations and the size of intermediate relations generated. Relational algebra trees can be restructured using transformation rules [17]. The transformation rules proved by [18]. We illustrate the six most important transformation rules to restruct the relational algebra trees which are:

1) Associativity of binary operations $((R \infty S) \infty T \approx R \infty (S \infty T))$,

2) Commuting of binary operations $(R \infty S \approx S \infty R)$,

3) Grouping of the unary operation (p1(A1) (p2(A2) (R)) ≈ p1(A1) p2(A2)(R)),

4) Commuting the selection with projection operation,

5) Commuting the selection with binary operations (p(A1) ≈ (R ∞ S) (p(A1)(R))∞ S)

6) Commuting the projection with binary operations.

This six rules permit substantial transformation that can optimize query execution. A heuristic algorithm that apply the six transformation rules on the relational algebra tree. The reconstruction algorithm divided into the following sequential steps. Separating the unary operation on the same relation, pushing it down level on the tree as much as possible, commuting the unary and binary operation and finally ordering the binary operation. The result from this algorithm is the candidate algebric trees for execute the query.

3.2- REDUCTION BY USING SEMIJOIN AND DISTRIBUTIVE JOIN

The join operation is the most important operation in relational database system operations. In distributed database system the join operation is a serious operation because it determines the amount of data that transfer from a site to another through the communication network. Several algorithms have been developed for performing the join operation with reducing the size of join operandes. We concentrate on two algorithm to reduce the join operandes sizes which are semijoin (SJ) and distributive join(DJ).

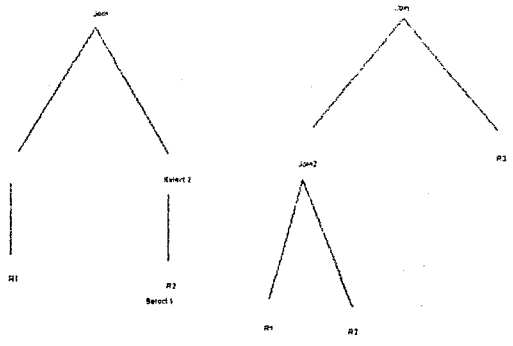
Semijoin between the relation Ri and relation Rj over attribute A is done as the follows, Firstly scan the relation Ri, generate the projection on the relation Ri in hashed form and transmit the semijoin projection to Rj. Secondly perform join in site of Rj where each tuple of Rj is checked against all the semijoin projection tuples through the hashing. The relation Ri is called the reducer and the relation Rj is reduced from the semijoin.

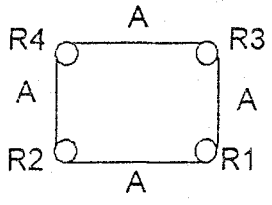
The distributive join belongs to the sort-based join algorithm (the operandes is sorted before the joining) [19]. This technique is used when we have limited size of main memory and when the relation size difference between them is double or more. Therefor the main idea is to distribute the largest relation into a set of small subrelations and applying a partial joining. The structure sequence of the distributive join is First, sort the small relation and produce the distribution table containing the boundary values of the join attributes. Second, distribute the largest relation according to the main memory by using the distribution table into subrelations. Finally, perform partial joins between the sorted relation and subrelations by using the nested join (substantial join).

4- NEW QUERY OPTIMIZATION PROCEDURE

The local processing layer of the optimization phase in DDBS is insignificant. SO, our algorithm is not concern with local processing (selection and projection). Thus first, the algebraic tree which produced from the transformation rules is translate to processing tree (**PT**)(tree modelling an execution strategy where a leaf is a base relation and each nonleafe node represent an intermediate relation output from applying a basic operation on the base relation and the root is the final result as shown in Figure (2)). All local operations in

the processing tree is executed and translate it to join tree (JT)(tree modeling an execution strategy where each leaf is a base relation and a nonleaf is a join operation and each edge represent the attribute variable and the root is the final result as shown in Figure(3)). Our algorithm take the JT as input and convert it to Join Graph (JG) (each node is a base relation or fragmented subrelation and each edge labeled by the join attribute as shown in Fig.(4)). The conventional approaches are to reduce the amount of data transmission by applying a sequence of semijoin to reduce





Figure(2): Processing Tree, Join Tree and Join Graph

the relation and ship the intermediate result to the final site to perform the final operations. There is problem related to join processing which are Sequential JOIn operations will lead to Loss of paraldelism (we

can avoid it by apply multiple join operations in parallel), Serial join operation on the same relation (we can avoid it by scanning the query for each relation sharing in it) and Processing overhead (we can avoid it by scan one relation at one time). The solution of all this problem by applying a various parallelism phases at the same time as parallel join, parallel transmitting the reduced relation and final processing will be sequential.

Parallel Join can done in two ways depending on the join selectivity factor (proportion of the tuples in the database that satisfy the join) and the memory size. First, parallel smijion will be done by parallel scanning for all relations that contribute in the query at the same time. The parallel semijoin consists of parallel projection generating of the reduced relation and parallel transmitting of the reduced relation. Second, Parallel Distributive Join can be done by scanning the small relation and sorting it and construct the distribution table which give information about each minimum and maximum at each subrelations.

After the reduction layer is complete then beginning the final processing at the processing site and ship the final result to the final site or at the final site.

4.1- NEW QUERY OPTIMIZATION MODEL

Our approach model is a general computers configuration (sit to site connection network topology) and the communication mode is broadcasting which sent the same information to multiple site at the same time for increasing the parallelism. Assume there is n sites S1,S2,S3,...,Sn, the delay time for transmit the X bytes from Si to Sj

(i,j = 1,2,...n)is DELAY = D0 + D1*X Where D0 is initial setup time, D1 is actual time for transmit X. The database model is each relation is stored in single site, this mean for each relation Ri is stored at site Si. The processing model is master and slave processing where the master process is that program which at the site where the query is initiated and the slave process that process run at on each site that contain a relation involved in the query.

Our objective of this approach is to minimize the response time without regarding by the local processing cost, communication cost, the retransmission cost, transmission error.

NHA algorithm

INPUT:- Join query graph.

OUTPUT:- Set of parallel execution strategies.

START

1- Perform local processing.

2- For each join attribute A sharing in the query DO

2.1-Order the relations according to their sizes(nrelations)

card (projection (R_i)over A)< card(projection (R_j)over A).

i,j = 1,2,...,n

2.2- Permute between binary of the ordered relations

2.3- Estimate the cost of each permutation and select the minimum estimation, delete the redundant and construct attribute strategy set.

3- For each relation contribute in the query DO

3.1- Select the join attribute over the relation

3.2- From the attribute strategy set construct :-

3.2.1- Sender table 3.2.2- Receiver table

3.3- From the sender table if there is a contradiction between relations that send to the same site, then, select the small one, else if the same relation sends and receives at the same time then select sending operation to execute first.

3.4- Determine the relations that send in parallel at the same time, put the result into best strategies set and delete this relation from the sender table.

3.5- Repeat step 3.3 until the sender table becomes empty.

4- After parallel sending of all the relations :

4.1- If the final site included in the receiver sites then transmit all the intermediate results to it and perform the final processing.

4.2- Otherwise select the largest receiver site to be the processing site and after final processing transmit the final result to the final site.

END.

Theorem:

The complexity of HNA algorithm is $O(K^2)$, where k is summation of relation degrees.

Proof:

For a given query which contains n relations. The complexity of step 2.1 is depended on the type of sorting algorithm used, in worst case it is O(n2). The complexity of step 2.2 is K*(K-1) where, K = (degree of R_1 + degree of R_2 +....+ degree of R_n) / 2. The complexity of step 3 is O(n). While, K always greater then n then the complexity is $O(K^2)$.

Example of query distributed processing; suppose the databases are R_1, R_2, R_3, R_4,and R_5, each relation is stored in a separate site. The query in this database is :

SELECT B, D, F FROM R1, R2, R3, R4, R5 WHERE R1.A = R2.A AND R1.B = R2.B AND R2.C = R5.C AND R2.F = R3.F AND R3.E = R4.E AND R4.D = R5.D

The given statistics are. | A | = 1000, | B | = 1200 , | C | = 1200, | D | = 800, | E | =1000, | F | = 600, Access delay time is 0.1 sec. and transfer rate is 1000 bit/sec. The rest of the statistics given in table 1.

Total communication time = Total Access Delay time + (Total Data Size / Transmit Rate).

Three different ways were proposed to execute the given query by different polices as follows:

Table 1

R. Name	| Ri |	Attribute	| Ri A |	SFi A	Size A	Size of R
R_1	1190	A	850	0.85	2	3570
		B	1100	0.92	1	
R2	3440	B	900	0.75	1	17200
		C	1000	0.83	3	
		F	480	0.80	1	
R3	1180	E	900	0.90	1	2360
		F	450	0.75	1	
R4	3100	D	700	0.88	1	6200
		E	800	0.80	1	
R5	2152	A	800	0.80	2	12912
		C	100	0.83	3	
		D	720	0.90	1	

POLICY 1 :

* This represent the conventional approach which transfer the relations R1, R2, R4, R5 to S3 and processing the query in S3 by join the R1, R2, R3, R4 and R5.

Total comm. time = 0.1*5 + (570+17200+2360+6200+12912)*8/1000

= 0.5 + 332.16 = 332.66 sec.

Res. T = 332.66 sec.

POLICY 2 (Chen and Yu): * Apply a semijoin in parallel on relations R3 to R2 over F and R4 to R5 over D. Also, apply a sequential join and semijoin on rest of the query.

Total Comm. time = 0.1 + 700*8/1000 + 0.1 + 450*8/1000 + 0.1 + 900*8/1000 + 0.1 + 2678*8/1000 + 0.1 + 8388*8/1000 + 0.1 + 12*8/1000 + 0.1 + 62*8/1000 = 0.7 + 13190*8 /1000= 106.2 sec.

Res. T = Max(5.7,3.7) + 96.8 = 102.5 sec.

OUR POLICY 3 : According the algorithm, construct the join attribute set for each attribute, construct the Sender and receiver table as shown in table 2, table 3. From our algorithm we can detect that the First, R1 will transmit the parallel projection semijoin to R2, R5 and R3 Will transmit the projection semijoin to R4. Second, R5 will transmit the parallel semijoin projection to R2, R4, Finally R3 will transmit the semijoin projection to R2. The processing site is R2 while it is not the final site and also it is largest processing site, So, after local join processing with R4 transmit the final result to R2, also transmit the processing of R5 to R2 and complete the final processing in R2, Finally transmit the final result to the final site R3.

Table 9. Sender table.

Site		
R1	R2,893	R5, 652
R5	R2, 2480	R4, 944
R3	R2, 1786	R4, 1894

Table 10. Reciver table.

Site			
R2	R1	R5	R3
R4	R3	R5	
R5	R1		

R.T1 = Max (952, 893, 944)*8/1000 + 0.1 = 7.71 sec.

R.T2 = Max (1786,1894)*8/1000 + 0.1 = 15.25 sec.

R.T3 =(2480)*8/1000 + 0.1 = 19.94 sec.

R.T4 = (944 * (SF 4,D)0.88)*8/1000 + 0.1 = 6.74 sec.

R.T5 = 952*8/1000 + 0.1 = 7.71 sec.

R.T6 = 62*8/1000 + 0.1 = 0.6 sec.

Total Response Time = 7.71+15.25+19.94+6.74+7.71+0.6 = 57.95 sec. The total comparison of all these polices in table 11.

Table 11.

	Conventional	AHY	NHA
Response Time	332.66	102.50	57.95
Precent form Conven.	100 %	30.8 %	17.4 %

CONCLUSION

Conventional query processing algorithms are poorly for parallel processing. In this paper, we discused the details for the query processing phases which are decomposition, optimization, localization and execution phase. Also, we proposed new heuristic algorithm for processing the distributed queries. The proposed algorithm introduces a better response time than [20]. Our algorithm is based on the semijoin and distributive join in reduction layer of optimization phase. The proposed algorithm is also based on parallel join, parallel transmitting.

REFERENCES

[1] S.K. Chang and W.H. Cheng. "A Methodology for Structured Database Decomposition", IEEE Transaction on Software Engineering, Vol., SE-6, No. 2,pp 205-218, March 1980.

[2] S.K. Chang and A.C. Liu. "File Allocation in a Distributed Database", Int. J. Comput. Inf. Sci., Vol., 11, No. 5. pp 9-36, 1982.

[3] Jark, M. and J. Kochi. "Query optimization in database system", ACM Computing Surveys, Vol., 16 No. 2, 1984.

[4] D. J. Rosenkrantz and H. B. Hunt. "Processing Conjunctive

Predicate and Queries", Int. Conf. on VLDB, Montreal (Canada), October 1980.

[5] Peter Pin-shan Chen, Jacob Akoka. "Optimal Design of Information System", IEEE Transaction on computers,Vol., C-29, No. 12,pp 1068-1080, December 1980.

[6] Robert S. Epstein. "Query Processing Techniques for Distributed Relational Database System", Computer Science: Distributed Database System, No. 13, 1982.

[7] Bezalel Gavish and Hasan Pirkul. "Computer and Database Location in Distributed Computer System", IEEE Transaction on computers, Vol., C-35, No. 7,pp 583-590, July 1986.

[8] W. W. Chu. "Optimal file allocation in multiple computer system", IEEE Transaction on computers, Vol., C-18, pp 885-889, 1969.

[9] H.L.Morgan and K.D. Liven. "Optimal program and data location in computer network", Commun. Ass. Comput. mach., Vol., 20, No. 5, pp315-321,1977.

[10] Wesley W. Chu and Paul Hurley. "Optimal Query Processing for Distributed Database System", IEEE Transaction on computers, Vol., C-31, No. 9, pp 835-850, 1982.

[11] Clement T. Yu, Keh-Chang Guh AND Arbee L. P. Chen. "Algorithms to Process Distributed Queries in Fast Local Network",IEEE Transaction on computers, Vol., C-36, No. 10, pp 1153-1164, October 1987.

[12] Chiphing Wang, Arbee L. P. Chen and Shiow-Chen Shyu. "Aparallel Execution Method for Minimizing Distributed Query Response time", IEEE Transaction on parallel and distributed system, Vol., 3, No. 3 pp 325-333, May 1992.

[13] M. Epstien, M. Stonerbaker and E. Wong. "Distributed query processing in a relational database system", ACM SIGMOD Int. Conf. Management Data, May 1978, pp 169-180.

[14] P. M. G. Apers, A. R. Hever and S. B. Yao. "Optimization algorithms for distributed queries", IEEE Transaction on Software Engineering, Vol., SE-9, pp 57-68, Jan 1983.

[15] Codd, E. F. "Relational completeness of database sublanguage Database system, couraut Computer Science series, Vol., 6, Prentice-Hall, 1972.

[16] P. A. Bernstien and D. M. Chiu. "Using semi-join to solve Relational queries", Journal of the ACM, Vol., 28, No. 1, 1981.

[17] J. D. Ullman. "Principles of Database System", Computer Science Press, 379 pages, 1982.

[18] A. V. Aho, Y. Sagiv and J. D. Ullman. "Equivalence among Relational Expression", SIAM Journal of Computing, Vol., 8, No. 2, pp 218-246, May 1979.

[19] M. Negri and G. Pelagatti. "Distributive join : A New Algorithm for joining Relations", ACM transaction on database system, Vol., 16, No. 4, pp 655-669, Decmber 1991.

[20] Ming-Syan Chen and Philip S. Yu. "Interleaving a Join Sequance with Semijion in Distributed Query Processing" IEEE Transaction on parallel and distributed system, Vol., 3, No. 5 pp 611-621, September 1992.

[21] M. Tamer Ozsu, Patrick Valuriz. "Principle of Distributed Database system", Prentice - Hall, Inc. 1991.

الملخص العربي

البحث يقدم نظاما جديدا لتشغيل الإستفسارات فى قواعد البيانات الموزعة ويعتمد النظام الجديد على التنفيذ على التوازي لتقليل زمن الإستجابة وهو أحد الأهداف المطلوبة فى قواعد البيانات الموزعة.