International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT217612

Design and Implementation of EFSC Algorithm for Hybrid Representation of Distributed Databases

[1]D. Sahithi, [2]Dr. J. Keziya Rani

[1]Research Scholar, Department of Computer Science & Technology, S. K University, Anantapuramu, Andhra Pradesh, India

[2]Associate Professor, Department of Computer Science & Technology, S.K University, Anantapuramu, Andhra Pradesh, India

Article Info
Volume 7, Issue 6
Page Number: 89-99

Publication Issue :
November-December-2021

Article History
Accepted : 10 Nov 2021
Published : 22 Nov 2021

ABSTRACT

In distributed database management systems, fragmenting base connections increases concurrency and hence system throughput for query processing. User queries use hybrid fragmentation methods focused on vector bindings, and deductive database implementations lack query-access-rule dependence. As a result, for hierarchical deductive information implementations, a hybrid fragmentation solution is used. The method considers the horizontal partition of base relations based on the bindings placed on user requests, then produces vertical fragments of the horizontally partitioned relations, and finally clusters rules based on attribute affinity and query and rule access frequency. The suggested fragmentation approach makes distributed deductive database structures easier to develop.

Keywords : Distributed Database Management Systems, Fragmentation

I. INTRODUCTION

Nowadays, various database management system (DBMS) evolving fast and emerging toward instinctively and systematically tackle the rising varieties of data size, scopes, schemas, speeds and data structures(Raj, 2018). These data type share consciously and carefully cleansed, collected and crunched toward squash out timely and actionable insights out of blowing up data heaps. In order to devise precise and perfect strategies, the data is one of important asset for any glowing and growing organization across world that roll out robust and accurate activities in timely manner with continuity, clarity and data confidence. When subjected to deeper and decisive investigations of data (both external and internal), it can emit the venerable and viable information that insights to the more in-depth analysis. Here, not only with the data, in addition to this, data ingestion, processing, transformation, mining and analytics for efficient data management, representation, persistence, engineering, governance, exchange and science that require effective technique, and unique tool development(Raj, 2018; Mordvanyuk et al., 2021). As we are heading toward the dreamt knowledge era, the role and responsibility of new generation database systems are bound to escalate in the days ahead. Especially the growth of data communication among the user and in the field of information technology, the distributed database system (DDBS) plays a significant role (Özsu et al., 2011).DDB is a collection of logical and numerous designs that are inter-related to the computer network. The database is distributed amongst various sites and linked with multiple link speeds which frequently affect the system performance of database system, transmission cost and response time of database system (Suganya & Kalaiselvi, 2013). So, there is a need for enhancement with the design of DDBS help to improvise the data transaction availability, response time, reliability, and its performance of the system. In most of the cases, the distributed system performance is majorly affected the three design issues, namely, data allocation, fragmentation and degree of replication (Mehta et al., 2018). The term fragmentation is the procedure of portioning data into disjoint fragments, whereas the strategy for assigning fragments over multiple or single sites are referred to as data allocation. The term replication is to decide like a procedure to allocate fragments, describe the fragment and procedure for replication of fragments. However, these approaches have a significant impact on minimizing the communication cost and minimizing the response time of query amongst various sites. But, most of the application platform, the data workloads are dynamic in nature with simultaneous variation in pattern access at multiple sites. Hence, these approaches are required to understand profoundly and enhance the performance of DDBS (Nashat & Amer, 2018). Hence, the present study focuses on the optimal design of distributed database systems while consideration of data allocation and fragmentation.

1.1 Motivation of the Study

The design of DDBS is one of the most complex issues that include allocation and fragmentation (Mehta et al., 2018). The fragmentation is the distributed database system design approach toward segregate the class of database or data relation into two or more data partitions which offers database without any loss of data (Shahidul Islam Khan & Hoque, 2010). This minimizes the access of irrelevant data through database applications hence its minimize the ratio of disk accesses. The fragmentation design process is classified into three types, such as vertical-fragmentation (VF), horizontal-fragmentation (HF), hybrid or mixed-fragmentation (MF) (Mehta et al., 2018). VF permits a class or relationship to be separated data into disjoint sets of attributes or columns except for the primary key. HF allows a class or relationship to be divided data into disjoint instances or tuples (Navathe et al., 1995). The term allocation is the procedure of assigning database fragments on various sites of a computer network. It may either maintain or replicated as a single copy when data is allocated. The efficiency of read-only queries, and reliability is improved by replication of fragments; however, it may increase the query update (Shahidul Islam Khan & Hoque, 2010; Ozsu & Valduriez, 1999). The data fragmentation relation is primary used to, improve availability and reliability of data, reference query locality is increased to DDM, enhance performance of a system, communication costs will be minimized among sites and balance the storage capacities (Baiao et al., 2004). Most of the existing approaches of MF, VF and HF face the issues like, fragmentation-based min-term predicates' affinity or attribute affinity matrix (AAM) which uses the frequency of queries. This required enough research data which is not available in the initial stage of DDBS. So, most of the researcher only focused on overlooked allocation or only on fragmentation issues to minimize the complexity (Shahidul Islam Khan & Hoque, 2010). Additionally, the data size increases, there exists data leakage, lack of security, etc. Thus, we focus on design, develop an efficient secure frequency counting method for partitioning the data in vertical and horizontal manner.

The rest of the paper is structured as follows. Sections 2 presents a literature review based on a distributed database system, vertical and horizontal fragmentation then and discuss the issues still we are facing then framed the solution for the same. Section 3 and 4 presents the concepts, methods, metrics and theoretical background about hybrid fragmentation of distributed database system. Section 5 discusses the obtained results of a proposed model; then we were compared the performance with traditional techniques. Finally, Section 6 discusses the study contribution, findings and concluding remarks and suggested a possible direction for further work.

II. Related Work

The previous studies related to data fragmentation, allocation over a computer network are discussed as follows:

A research by Wu et al., (2020) presented a model based on community discovery of distributed business procedure fragmentation approach. Then, they have optimized the server communication cost together with the functionally related process partitions during the execution of the business process. Also, they have adopted a control flow model and advanced fuzzy models for community discovery. They have obtained optimized business process, efficiently maintained the server resource load balancing and minimized the communication cost between data access and servers by applying community-based process fragmentation approach. In future, they have planned to suggest a optimization approach that will help to improve the cluster performance through effectively fragment the data.

A study by Song et al.(2018) presented a scalable approach for heterogeneous data representation which is also called as Distributed Semantic Network (DSN). The suggested model helps to mine semantic data from various data sources. Furthermore, the researcher has utilized Wikipedia and Word Net toward scale out the data of both vertical and horizontal manner. The researchers have constructed the base knowledge efficiently in distributed and parallel computing via applying Map Reduce framework. The simulation results demonstrated that DSN gives better results for semantic textual data with a high precision score. However, some of the extracted information has a complex structure of destructive grammatical forms that leads to multiple order semantic parsing and incorrect dependency parse. So that has to be addressed in the near future study.

A work by Herce-Zelaya et al. (2019) presented a web platform for helping the students learning the processing and optimization of queries in distributed databases. The novelty of this platform is that there is no similar graphical tool as far as we know. It allows visualizing step by step the different phases of distributed query processing, showing how are they forming, making it easier for the students to understand these concepts. The results of the developed test to validate the platform's functionalities and student's satisfaction were very positive. Despite the promising results, we are aware that the system still has room for improvement that will be addressed in the next versions. In particular, it needs to improve the implemented lexical and syntactic analyser, because, in their current state, their use is very restrictive and returns syntax errors that should not return.

A research by Rahimi et al. (2018) presented a hybrid combination of data allocation and fragmentation. Significantly allocates fragments to suitable sites and frequently fragments data vertically across the network using bond energy approach. Also, the attributes of a generated cluster have been improved with better affinity measures. Furthermore, allocates each cluster to a suitable site then measures the allocating cost of the entire site and frequently generates sets of attributes. In future, they have planned to extend this model by applying a better approach to find the weights of the similarity measure, by updating each attribute frequency range and by optimizing the data allocation cost function of the network.

A study by Amer et al. (2017) developed a data allocation and optimized heuristic horizontal fragmentation method. By this hybrid combination, productivity promotion of distributed database system is set to mark as fulfilled. Furthermore, they have illustrated that external and internal evaluation of the model. The obtained results have proven that the suggested hybrid method gives better performance result in a distributed database system.

A research by Harikumar and Ramachandran, (2015) presented a hybrid fragmentation toward minimizing the database access time based ona subspace clustering approach. Based on the attribute and tuples patterns, the data fragments were generated, which is closely correlated information was assembled together. Meanwhile, research by Hauglid et al., (2010) developed a data decentralized method for allocation and fragmentation of dynamic table in distributed database system toward improves the data locality performance on the basis of recorded access history. Furthermore, they have experimentally demonstrated that suggested model feasibility. The same model has been enhanced by Raouf et al., (2017) to perform data allocation, replication and initial stage fragmentation at the execution time of cloud platform. Also, they have tackled site clustering and then enhanced the distributed database system performance via increasing the local data accessibility.

A work by Kechar & Bahloul (2014) developed a hybrid fragmentation algorithm combining a vertical fragmentation based on XPath expressions and a horizontal fragmentation based on selection predicates. To control the number of fragments, we use the K-Means algorithm. Finally, they have validated the approach under Oracle Berkeley DB XML by several experiments on XML data, derived from the XWB benchmark. The results obtained allowed a user to confirm the effectiveness of our proposed hybrid fragmentation. In future work, they have planned to conduct an experimental comparison between the fragmentation algorithms proposed in (Boukraa et al., 2011) and (Mahboubi & Darmont, 2008), and our hybrid fragmentation algorithm.

A work by Shahidul Islam Khan and Hoque, (2010) focused on fragmentation approach, which is applied for partitioning the relations of data over the initial and final stages of DDBS. Simultaneously, they have performed the data allocation of fragments. The demonstrated results show that the suggested model able to resolve relational database fragmentation issues for distributed systems. Further, they have constructed attribute locality precedence for each relationship of a distributed database system on the basis of cost function and CRUD (create, read, update and delete). With the help of the suggested model, there is no need to add additional complexity for fragments allocation to various sites of DDBS, whereas the fragmentation process is synchronized with the data allocation process. By the avoidance of high data transfer amongst sites and frequent remote access, the performance is enhanced significantly. In future, they have planned to extend this model with a distributed object-oriented database system.

From the review of literature, we observed that the design and development of DD-distributed database is facing an optimization issue, resolution of various sub-problems as fragmentation like data allocation (without or with redundancy), data fragmentation (HF,VF,MF) and allocation and optimization of operations like a selection of the best execution strategy, request transformation and allocation of operations to sites. Some of the researchers stated that the design of DDBS is very complex, during the frequent increase in data size. And very few presented the scenario in the case of object-oriented database and relational database (Runceanu, 2008). Also, noted that VF and HF (Shahidul Islam Khan & Hoque, 2010) is widely applied (Ozsu & Valduriez, 1999), but MF Hari kumar and Ramachandran, (2015) is rarely used for partitioning databases. However, making efficient data allocation and appropriate fragmentation relation in DDBS is challenging issues (Nashat & Amer, 2018).On the other hand, most of the researcher used query frequencies and empirical knowledge of data access. But proper allocation and fragmentation at the initial stage of a distributed database have not yet been addressed (Shahidul Islam Khan & Hoque, 2010). So, the present research developed a hybrid fragmentation and allocation of frequently accessed attributes and various selection conditions to give a clear understanding of designing distributed database system.

III. Methodology

This study presents a design and development of secure frequency counting method for partitioning the data based on horizontal and vertical fragmentation. This proposed algorithm gives adequate independence for privacy-preserving mining of both vertical and horizontal representation of partitioned data.

Especially, developed a hybrid fragmentation (a combination of vertical and horizontal fragmentation) relation of a distributed database system to handle the relationship of data partitions into disassociated tuples as well as fragment the vertical link into smaller partitions by optimizing the cost function of data allocation. This method helps to take decisions at the initial stage without knowledge of practical information and correctly partitioning the data amongst various network sites of DDBMS. Here the vertical fragmentation is used to cluster the attributes thatwere frequently accessed together, and horizontal fragmentation is used to group's records that fulfil the data selection requirement.

Furthermore, we adopted association rule mining goals to discover item sets with co-occurrence of frequently performed transactions in both horizontal and vertical database process. Association rule mining process creates the relation of each item set containing each process in the needed format. For instance, if regulation is held globally and locally on data items, this specific rule was achieved data item in the real-time progression of processing item set. In addition, meta-heuristic algorithms were developed to provide near-optimal solutions to design of distributed database system. This improves the fragmentation and data allocation performance. The resulted outcome shows that we were minimized the cost function and reduces computational complexity.

During the distribution of data, it is mandatory to provide security to that data so that another party will never know their detailed information. Several approaches come for ensuring the safety to data mining is described below.

1. The data is misrepresented earlier than conveying it to the data miner.
2. The data disseminated among two or more sites, which support through a semi-honest protocol to acquire global data mining consequences without revealing any information about the data at their places.
3. Whereas by a prototypical to catalogue data, the taxonomy outcomes are only revealed to the designated party, who does not learn whatever else other than the classification results. Still, they can confirm the occurrence of specific rules lacking showing the rules.

3.1 Hybrid Fragmentation Method

In DD system, the fragmentation is used in a relational database system. But the case of the object-relational database system, the development of the algorithm is more complex. Also, needs to consider various criteria for fragmentation in object-relational databases such as a relation among classes, attribute and sub-class usage through superclass, hierarchical structure and so on.

In our study, we are planning to use a hybrid combination of vertical and horizontal fragmentation method. Here the vertical fragmentation is used to split the relation into sub relation, which is projections of relation based on the subset of the attributes. These relationships are frequently accessed through queries and grouping with attributes. This will improve the transaction processing cost through minimizing the network communication cost in DD sytem. With the help of query predicates, the query relation is divided into subsets of rows while performing horizontal fragmentation. By selecting the horizontal fragments, we can minimize the query processing cost and also the actual query relations are reconstructed through a group of fragments. The hybrid combination of this fragmentation, we can easily minimize the communication cost, minimize the query execution time and maximizing the query locality evaluation during process of query.

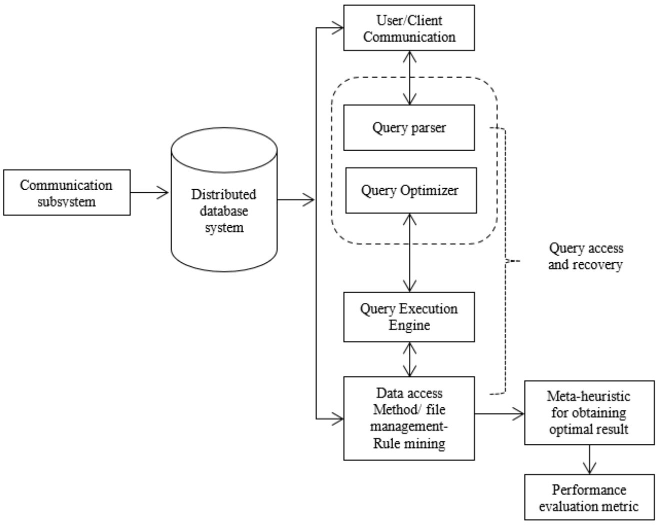

User/Client Communication
Query parser
Query Optimizer
Communication subsystem
Distributed database system
Query access and recovery
Query Execution Engine
Data access Method/ file management- Rule mining
Meta-heuristic for obtaining optimal result
Performance evaluation metric

Figure 1 : Proposed system architecture

3.2 Association Rule Mining

After completion query fragmentation method, the rule mining technique is applied to relation-based transaction database. This will create the creates the relation of each item set that contains each process in the needed format. For instance, if regulation is held globally and locally on data items, this specific rule was achieved in the real-time progression of processing item set. Furthermore, to generate the desired rules by utilizing frequent itemsets.Here the performance of query varies with specific to large data item sets, data size in a distributed system. Additionally, we have selected the best execution plan to enable the query optimizer with specified weight index. By rule mining technique, the row and column were indexed from the source table.

3.3 Meta-Heuristic Method

Third, a meta-heuristic approach was applied to obtain optimized distributed deployment schemas and partition the queries into a small subset. This model hasminimized the communication cost among servers in the actual process deployment of a relational database environment. The resulted outcome helps to improve fragmentation and data allocation performance.

The implementation process for query processing using the proposed approach is,

Step 1: Read and update a set of relational database queries.

Step 2: To perform separation of reading and update queries.

Step 3: Query partitions by applying hybrid fragmentation algorithm.

Step 4: query access, management and Partitions are allocated to corresponding sites by association rule mining approach

Step 5: obtain optimal results by meta-heuristic technique

Step 6: Stop iteration and validate the results regarding communication cost, blocking rate, execution time and standard deviation.

IV. Results

The performance of the both the Associatoin rule mining and meta-heuristic model is evaluated in terms of communication cost and execution time. The number of activity communication is obtained while simulating query instances, the number of server communication performed to execute the next activity in a distributed database. While affecting a model the number of times, an activity accesses data on another server is termed as a measure of data communication rate. During the simulation, the server usage is monitored frequently. The execution time is regarded base on the simulation time with communication.

4.1 Association Rule Mining

The rule to display the employee who is eligible for getting hike based on their hire date first we are creating the data table employee with the following field

EMP_NUMBER ,EMP_NAME, MANAGER, JOB, SAL, HIREDATE,COMM

To execute the date, create a table with sample data for emp and create two more table as same as emp but update that data using the rule which we derived. Hence

Update employee sal by 0.25 where hiredate between (1981-12-01 to 1981-12-31) in

EMP_new and

Update employee sal by 0.15 where hiredate between (1981-12-01 to 1981-06-30) in

EMP_new_1

Now join all the table with new table ‘dataset_emp’.

Here we have to specify two attributes.

Two major factors supports to generate the result that is support and confidence. Support is that if an employee satisfies both the rule then his/her support will be high when compared to other values. So we first initialize the support to 0 which means his/her support will be 2 which is maximum number of support. At the very first the confidence is 3 which means he /she may likely to get hike but if any of the rule failed he/she will get reduced in the confidence. After this computation the result is with employee having count of support and confidence

-if both the support and confident as 0 the employee does not get eligible for hike similarly and-update the commission pct value to 0

count of support and confidence is 1

then the employee may likely to get part hike

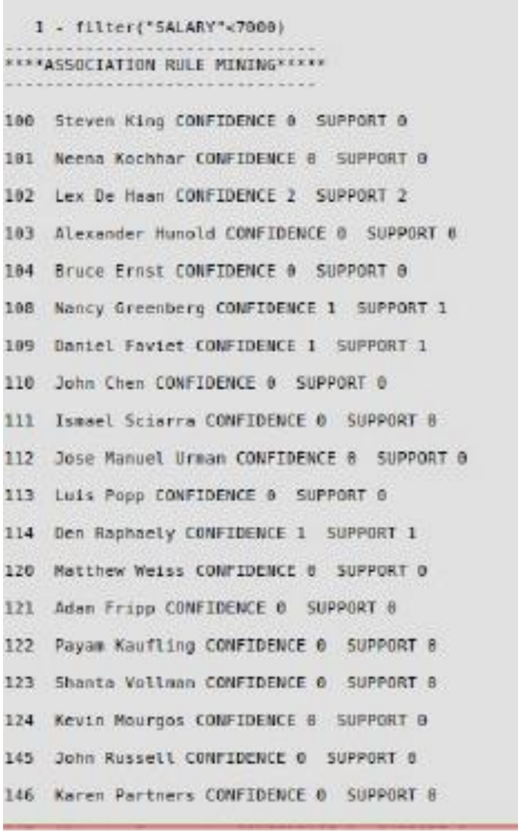
-update the commission pct value to 0.05

In such cases the employee who is having maximum amount of support and confidence which is 2

They will be eligible to receive a full hike.

-update the commission pct value to 0.10

Displayed the employee details along with sal.



```
  1 - filter("SALARY"<7000)
--------------------------------
****ASSOCIATION RULE MINING*****
--------------------------------

100  Steven King CONFIDENCE 0  SUPPORT 0

101  Neena Kochhar CONFIDENCE 0  SUPPORT 0

102  Lex De Haan CONFIDENCE 2  SUPPORT 2

103  Alexander Hunold CONFIDENCE 0  SUPPORT 0

104  Bruce Ernst CONFIDENCE 0  SUPPORT 0

108  Nancy Greenberg CONFIDENCE 1  SUPPORT 1

109  Daniel Faviet CONFIDENCE 1  SUPPORT 1

110  John Chen CONFIDENCE 0  SUPPORT 0

111  Ismael Sciarra CONFIDENCE 0  SUPPORT 0

112  Jose Manuel Urman CONFIDENCE 0  SUPPORT 0

113  Luis Popp CONFIDENCE 0  SUPPORT 0

114  Den Raphaely CONFIDENCE 1  SUPPORT 1

120  Matthew Weiss CONFIDENCE 0  SUPPORT 0

121  Adam Fripp CONFIDENCE 0  SUPPORT 0

122  Payam Kaufling CONFIDENCE 0  SUPPORT 0

123  Shanta Vollman CONFIDENCE 0  SUPPORT 0

124  Kevin Mourgos CONFIDENCE 0  SUPPORT 0

145  John Russell CONFIDENCE 0  SUPPORT 0

146  Karen Partners CONFIDENCE 0  SUPPORT 0
```

Figure 2 : Implementation of association rule mining

4.2 Meta Heuristic in Oracle Fragmentation

Query:

Find employee details like id ,name total salary,department id department name and locationalong with manager name for departments “Finance and Human Resources”

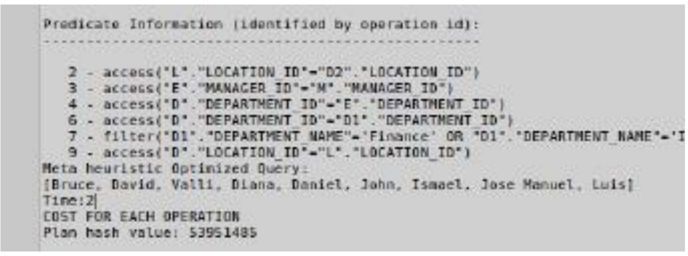
Normal Query:

select

distinct e.employee_id,

concat(e.first_name,' ',e.last_name) as name ,

```
    e.manager_id,
    e.salary as total_salary ,
    e.department_id,
    d.department_name,
    m.manager_name,
    d.location_id
    from employees e
    inner join managers m on
    e.manager_id = m.manager_id
    inner join departments d on
d.department_id=e.department_id and
    d.department_id in (select d1.department_id
from departments d1 where
    d1.department_name='Human Resources' or
    d1.department_name='Finance') and
    d.location_id in (select l.location_id from
departments d2
    inner join locations l on
    l.location_id=d2.location_id )
    order by e.employee_id;
```



```
Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("L"."LOCATION_ID"="D2"."LOCATION_ID")
   3 - access("E"."MANAGER_ID"="M"."MANAGER_ID")
   4 - access("D"."DEPARTMENT_ID"="E"."DEPARTMENT_ID")
   6 - access("D"."DEPARTMENT_ID"="D1"."DEPARTMENT_ID")
   7 - filter("D1"."DEPARTMENT_NAME"='Finance' OR "D1"."DEPARTMENT_NAME"='I
   9 - access("D"."LOCATION_ID"="L"."LOCATION_ID")
Meta heuristic Optimized Query:
[Bruce, David, Valli, Diana, Daniel, John, Ismael, Jose Manuel, Luis]
Time:2
COST FOR EACH OPERATION
Plan hash value: 53951485
```

Figure 3 : Query for Meta Heuristic Method

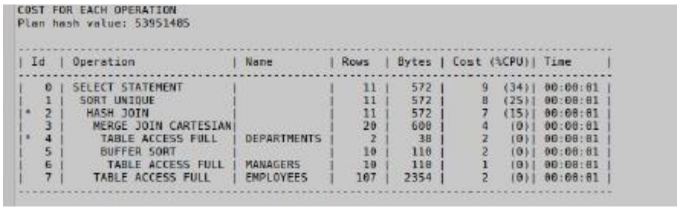
Query Explanation:

Here first we select the employee details from employee table then we join the required tables like department and manager so the steps will be more like selection,joining and subquery inside the join .

i.e NESTED LOOP SELECTION , NESTED LOOP SUB JOIN AND HASH JOIN

Optimized Query:

```
select
distinct e.employee_id,
concat(e.first_name,' ',e.last_name) as name ,
e.manager_id,
e.salary as total_salary ,
d.department_id,
d.department_name,
m.manager_name,
d.location_id
from managers m,departments d
inner join employees e on
e.department_id=d.department_id
where m.manager_id = e.manager_id and
( d.department_name = 'Finance' or
d.department_name = 'Human Resources' )
order by e.employee_id;
```



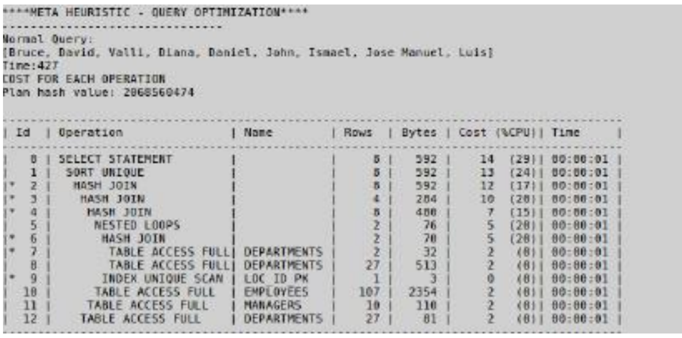
```
COST FOR EACH OPERATION
Plan hash value: 53951485

-----------------------------------------------------------------------------------
| Id  | Operation                | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------
|   0 | SELECT STATEMENT         |               |    11 |   572 |     9  (34)| 00:00:01 |
|   1 |  SORT UNIQUE             |               |    11 |   572 |     8  (25)| 00:00:01 |
|*  2 |   HASH JOIN              |               |    11 |   572 |     7  (15)| 00:00:01 |
|   3 |    MERGE JOIN CARTESIAN  |               |    20 |   600 |     4   (0)| 00:00:01 |
|*  4 |     TABLE ACCESS FULL    | DEPARTMENTS   |     2 |    38 |     2   (0)| 00:00:01 |
|   5 |     BUFFER SORT          |               |    10 |   110 |     2   (0)| 00:00:01 |
|   6 |      TABLE ACCESS FULL   | MANAGERS      |    10 |   110 |     1   (0)| 00:00:01 |
|   7 |    TABLE ACCESS FULL     | EMPLOYEES     |   107 |  2354 |     2   (0)| 00:00:01 |
-----------------------------------------------------------------------------------
```

Figure 4 : Optimized Query for Meta Heuristic Method

Query Explanation:

Here in the optimzed query there is minimal number of computation and hence we are opt out reqiured thing at the last only after simple join. Since A metaheuristic method helps in solving the optimization problem and a metaheuristic is also known as for computational Intelligence so our computation time gets reduced compared with normal query in the above.



```
****META HEURISTIC - QUERY OPTIMIZATION****
---------------------------------
Normal Query:
[Bruce, David, Valli, Diana, Daniel, John, Ismael, Jose Manuel, Luis]
Time:427
COST FOR EACH OPERATION
Plan hash value: 2068560474

-----------------------------------------------------------------------------------
| Id  | Operation                | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------
|   0 | SELECT STATEMENT         |               |     8 |   592 |    14  (29)| 00:00:01 |
|   1 |  SORT UNIQUE             |               |     8 |   592 |    13  (24)| 00:00:01 |
|*  2 |   HASH JOIN              |               |     8 |   592 |    12  (17)| 00:00:01 |
|*  3 |    HASH JOIN             |               |     4 |   204 |    10  (20)| 00:00:01 |
|*  4 |     HASH JOIN            |               |     8 |   400 |     7  (15)| 00:00:01 |
|   5 |      NESTED LOOPS        |               |     2 |    76 |     5  (20)| 00:00:01 |
|*  6 |       HASH JOIN          |               |     2 |    70 |     5  (20)| 00:00:01 |
|*  7 |        TABLE ACCESS FULL| DEPARTMENTS   |     2 |    32 |     2   (0)| 00:00:01 |
|   8 |        TABLE ACCESS FULL| DEPARTMENTS   |    27 |   513 |     2   (0)| 00:00:01 |
|*  9 |       INDEX UNIQUE SCAN | LOC_ID_PK     |     1 |     3 |     0   (0)| 00:00:01 |
|  10 |      TABLE ACCESS FULL  | EMPLOYEES     |   107 |  2354 |     2   (0)| 00:00:01 |
|  11 |     TABLE ACCESS FULL   | MANAGERS      |    10 |   110 |     2   (0)| 00:00:01 |
|  12 |    TABLE ACCESS FULL    | DEPARTMENTS   |    27 |    81 |     2   (0)| 00:00:01 |
-----------------------------------------------------------------------------------
```

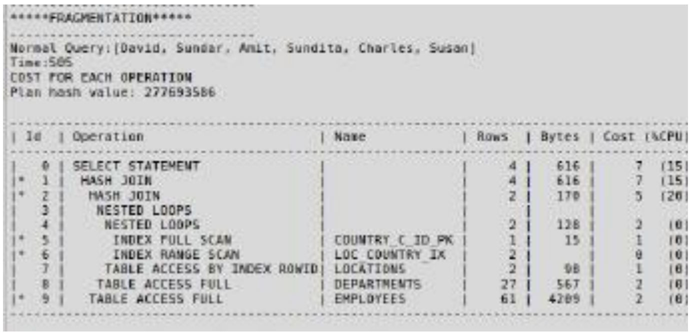
Figure 5 : Time Execution in Optimized Query

4.3 Hybrid Fragmentation

Here we are creating a table places from locations and countries with their region idas primary key join column. Also we are creating another table employee_info which contains all the all the employee details along with their locations. Here in the employee_info we are using the table places which we created for achieving verical Fragmentation. For achieving Hybrid fragmentation for instance, if we need to create a new table employee_info_usa. So if we want to access particular location employee it is easy to us to retrive datajust as giving country_name as USA and then we can access data for each employeeon the location USA.

4.3.1 Traditional Query Vs Hybrid Fragmentation

Normally if we want to create a above table employee_info_usa provided we are having the tables locations,employee and regions. So if we want to access the employee details in contry usa means we need to perform multiple join operation using the above table but in this Hybrid fragmentation we already created the employee_info using vertical fragmentation so the retrival of particular location seems faster than the traditional way of retrieving. Contains sample output for fragmentation with exection time and cost for each opration(inside a query). For example, the timing is set for 505and the execution is processed to retrieve the result.



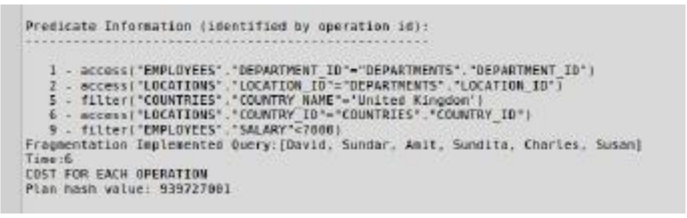
```
*****FRAGMENTATION*****

Normal Query:[David, Sundar, Amit, Sundita, Charles, Susan]
Time:505
COST FOR EACH OPERATION
Plan hash value: 277693586

| Id  | Operation                    | Name            | Rows  | Bytes | Cost (%CPU|
------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                 |     4 |   616 |     7  (15|
|*  1 |  HASH JOIN                   |                 |     4 |   616 |     7  (15|
|*  2 |   HASH JOIN                  |                 |     2 |   170 |     5  (20|
|   3 |    NESTED LOOPS              |                 |       |       |           |
|   4 |     NESTED LOOPS             |                 |     2 |   128 |     2   (0|
|*  5 |      INDEX FULL SCAN         | COUNTRY_C_ID_PK |     1 |    15 |     1   (0|
|*  6 |      INDEX RANGE SCAN        | LOC_COUNTRY_IX  |     2 |       |     0   (0|
|   7 |     TABLE ACCESS BY INDEX ROWID| LOCATIONS     |     2 |    98 |     1   (0|
|   8 |    TABLE ACCESS FULL         | DEPARTMENTS     |    27 |   567 |     2   (0|
|*  9 |   TABLE ACCESS FULL          | EMPLOYEES       |    61 |  4209 |     2   (0|
------------------------------------------------------------------------------------
```

Figure 6 : Hybrid Fragmentation

And in next step the fragmentation implementation occurs for the employee name and the time is set for 6.

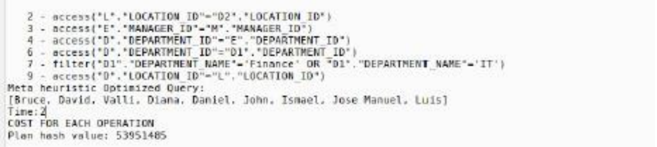


```
Predicate Information (identified by operation id):
---------------------------------------------------

   1 - access("EMPLOYEES"."DEPARTMENT_ID"="DEPARTMENTS"."DEPARTMENT_ID")
   2 - access("LOCATIONS"."LOCATION_ID"="DEPARTMENTS"."LOCATION_ID")
   5 - filter("COUNTRIES"."COUNTRY_NAME"='United Kingdom')
   6 - access("LOCATIONS"."COUNTRY_ID"="COUNTRIES"."COUNTRY_ID")
   9 - filter("EMPLOYEES"."SALARY"<7000)
Fragmentation Implemented Query:[David, Sundar, Amit, Sundita, Charles, Susan]
Time:6
COST FOR EACH OPERATION
Plan hash value: 939727001
```

Figure 7 : Fragmentation Implementation

The access happens for all the field and one particular attribute alone is filtered with the help of meta-heuristic method.



```
   2 - access("L"."LOCATION_ID"="D2"."LOCATION_ID")
   3 - access("E"."MANAGER_ID"="M"."MANAGER_ID")
   4 - access("D"."DEPARTMENT_ID"="E"."DEPARTMENT_ID")
   6 - access("D"."DEPARTMENT_ID"="D1"."DEPARTMENT_ID")
   7 - filter("D1"."DEPARTMENT_NAME"='Finance' OR "D1"."DEPARTMENT_NAME"='IT')
   9 - access("D"."LOCATION_ID"="L"."LOCATION_ID")
Meta heuristic Optimized Query:
[Bruce, David, Valli, Diana, Daniel, John, Ismael, Jose Manuel, Luis]
Time:2
COST FOR EACH OPERATION
Plan hash value: 53951485
```

Figure 8 : Fragmentation using Meta-Heuristic method

V. Conclusion

The functional goal is to incorporate and improve a user interface application that aids in the fragmentation of hybrid databases. With advancements in networking technology, the popularity of distributed database networks has grown ever further. Effective database delivery using hybrid fragmentation is crucial to the database's consistency and availability.For hybrid fragmentation, a new formulation for the problem of fragmenting and allocating certain fragments in the shortest time is proposed in this paper. The results of applying these formulations to a vast range of data sets can be used.

VI. REFERENCES

[1]. Amer, A.A., Sewisy, A.A. & Elgendy, T.M.A. (2017). An optimized approach for simultaneous horizontal data fragmentation and allocation in Distributed Database Systems (DDBSs). Heliyon. [Online]. 3 (12). pp. e00487. Available from: https://www.sciencedirect.com/science/article/pii/S2405844017314421.

[2]. Baiao, F., Mattoso, M. & Zaverucha, G. (2004). A distribution design methodology for object DBMS. Distributed and Parallel Databases. [Online]. 16 (1). pp. 45–90. Available from: https://link.springer.com/content/pdf/10.1007/978-3-030-26253-2.pdf.

[3]. Boukraa, D., Boussaid, O. & Bentayeb, F. (2011). Vertical fragmentation of XML data warehouses using frequent path sets. In: International Conference on Data Warehousing and Knowledge Discovery. [Online]. 2011, Springer, pp. 196–207. Available from: https://link.springer.com/chapter/10.1007/978-3-642-23544-3_15.

[4]. Harikumar, S. & Ramachandran, R. (2015). Hybridized fragmentation of very large databases using clustering. In: 2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES). 2015, IEEE, pp. 1–5.

[5]. Hauglid, J.O., Ryeng, N.H. & Norvag, K. (2010). DYFRAM: dynamic fragmentation and replica management in distributed database systems. Distributed and Parallel Databases. 28 (2–3). pp. 157–185.

[6]. Herce-Zelaya, J., Porcel, C., Bernabé-Moreno, J., Tejeda-Lorente, Á. & Herrera-Viedma, E. (2019). Web platform for learning distributed databases' queries processing. Procedia Computer Science. 162. pp. 827–834.

[7]. Kechar, M. & Bahloul, S.N. (2014). Hybrid Fragmentation of XML Data Warehouse Using K-Means Algorithm. In: [Online]. pp. 70–82. Available from: http://link.springer.com/10.1007/978-3-319-10933-6_6.

[8]. Mahboubi, H. & Darmont, J. (2008). Data mining-based fragmentation of XML data warehouses. In: Proceedings of the ACM 11th international workshop on Data warehousing and OLAP. [Online]. 2008, pp. 9–16. Available from: https://dl.acm.org/doi/abs/10.1145/1458432.1458435.

[9]. Mehta, S., Agarwal, P., Shrivastava, P. & Barlawala, J. (2018). Differential bond energy algorithm for optimal vertical fragmentation of distributed databases. Journal of King Saud University-Computer and Information Sciences.

[10]. Mordvanyuk, N., López, B. & Bifet, A. (2021). vertTIRP: Robust and efficient vertical frequent time interval-related pattern mining. Expert Systems with Applications. [Online]. 168. pp. 114276. Available from: https://linkinghub.elsevier.com/retrieve/pii/S0957417420309842.

[11]. Nashat, D. & Amer, A.A. (2018). A comprehensive taxonomy of fragmentation and allocation techniques in distributed database design. ACM Computing Surveys (CSUR). [Online]. 51 (1). pp. 1–25. Available from: https://dl.acm.org/doi/abs/10.1145/3150223.

[12]. Navathe, S., Karlapalem, K. & Ra, M. (1995). A mixed fragmentation methodology for initial distributed database design. Journal of Computer and Software Engineering. [Online]. 3 (4). pp. 395–426. Available from: https://pdfs.semanticscholar.org/2564/a92add8b1749dbe3648ff14421691b6bd7d8.pdf.

[13]. Ozsu, M.T. & Valduriez, P. (1999). Principles of distributed database systems. [Online]. Springer. Available from: https://link.springer.com/content/pdf/10.1007/978-3-030-26253-2.pdf.

[14]. Özsu, M.T., Valduriez & Patrick (2011). Principles of Distributed Database Systems. 3rd Ed. [Online]. Available from: https://www.springer.com/in/book/9781493941742#otherversion=9781441988331.

[15]. Rahimi, H., Parand, F.-A. & Riahi, D. (2018). Hierarchical simultaneous vertical fragmentation and allocation using modified Bond Energy Algorithm in distributed databases. Applied Computing and Informatics. 14 (2). pp. 127–133.

[16]. Raj, P. (2018). A Detailed Analysis of NoSQL and NewSQL Databases for Bigdata Analytics and Distributed Computing. In: [Online]. pp. 1–48. Available from:

https://linkinghub.elsevier.com/retrieve/pii/S0065245818300020.

[17]. Raouf, A.E.A., Badr, N.L. & Tolba, M.F. (2017). Distributed database system (DSS) design over a cloud environment. In: Multimedia Forensics and Security. Springer, pp. 97–116.

[18]. Runceanu, A. (2008). Fragmentation in Distributed Databases. [Online]. Available from: https://link.springer.com/chapter/10.1007/978-1-4020-8735-6_12.

[19]. Shahidul Islam Khan & Hoque, M.L. (2010). A New Technique for Database Fragmentation in Distributed Systems. International Journal of Computer Application. [Online]. 5 (9). pp. 5. Available from: https://pdfs.semanticscholar.org/5ad7/4b2bddc7ba6e2c5446cbe7d72e1fc1407080.pdf.

[20]. Song, S., Lin, Y., Guo, B., Di, Q. & Lv, R. (2018). Scalable Distributed Semantic Network for knowledge management in cyber physical system. Journal of Parallel and Distributed Computing. [Online]. 118. pp. 22–33. Available from: https://linkinghub.elsevier.com/retrieve/pii/S074373151730326X.

[21]. Suganya, A. & Kalaiselvi, R. (2013). Efficient fragmentation and allocation in distributed databases. International Journal of Engineering Research & Technology (IJERT). 2. pp. 1–7.

[22]. Wu, B., Liu, R., Lin, R. & Chen, J. (2020). A distributed business process fragmentation method based on community discovery. Future Generation Computer Systems. [Online]. 108. pp. 372–389. Available from: https://www.sciencedirect.com/science/article/pii/S0167739X19325580.s

Cite this article as :

D. Sahithi, Dr. J. Keziya Rani, "Design and Implementation of EFSC Algorithm for Hybrid Representation of Distributed Databases", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 7 Issue 6, pp. 89-99, November-December 2021. Available at
doi : https://doi.org/10.32628/CSEIT217612
Journal URL : https://ijsrcseit.com/CSEIT217612