DESDE 2013
https://repository.uaeh.edu.mx/revistas/index.php/icbi/issue/archive
Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI

ISSN: 2007-6363

Publicación Semestral Pädi Vol. 11 No. Especial 2 (2023) 47-54

Dynamic Hybrid Fragmentation Method for Multimedia Databases
Método de Fragmentación Híbrida Dinámica para Bases de Datos Multimedia

O. Crescencio-Rico[a,*], L. Rodríguez-Mazahua[a], F. Castro-Medina[a], G. Alor-Hernández[a], J.L. Sánchez-Cervantes[b]

[a] *Tecnológico Nacional de México, Instituto Tecnológico de Orizaba, Orizaba, Ver., México.*
[b] *CONACYT, Tecnológico Nacional de México, Instituto Tecnológico de Orizaba, Orizaba, Ver., México.*

Abstract

Multimedia databases store high-volume data, which causes problems in efficient information retrieval, and increases execution costs and response times of the queries. To solve this problem, data fragmentation techniques exist to improve query performance, increase information availability, and efficiently execute more operations accessing less irrelevant data. This article presents a comprehensive review of 34 methods related to hybrid fragmentation and subsequently proposes the design of a hybrid fragmentation method that adapts the scheme according to workload changes to maintain efficient retrieval of multimedia data. The proposed technologies are Java as a programming language, Java Server Faces (JSF) as a framework, MySQL and MongoDB database management systems, and NetBeans as an Integrated Development Environment (IDE), following the UWE methodology (Unified Modeling Language-based Web Engineering).

Keywords: Hybrid Fragmentation, Dynamic Fragmentation, Cost Model, Multimedia Database.

Resumen

Las bases de datos multimedia almacenan datos de gran tamaño, provocando problemas en la recuperación eficiente de la información, aumentando los costos de ejecución y tiempos de respuesta de las consultas. Para resolver estos problemas, existen técnicas de fragmentación de datos que permiten mejorar el desempeño de las consultas, aumentar la disponibilidad de información y ejecutar eficientemente más operaciones accediendo menos a datos irrelevantes. En este artículo, se presenta una revisión exhaustiva de 34 métodos de fragmentación híbrida y posteriormente, se propone el diseño de un método de fragmentación híbrida que adapte el esquema de acuerdo con los cambios en la carga de trabajo para mantener la recuperación eficiente de datos multimedia. Las tecnologías deseadas para la implementación del diseño propuesto son el lenguaje de programación Java, el marco de trabajo JSF (JavaServer Faces), los sistemas gestores de bases de datos MySQL y MongoDB, y el entorno de desarrollo integrado NetBeans; siguiendo la metodología de ingeniería Web basada en el lenguaje unificado de modelado (UWE).

Palabras Clave: Fragmentación Híbrida, Fragmentación Dinámica, Modelo de costos, Base de Datos Multimedia.

1. Introduction

Multimedia databases in addition to containing text, store and retrieve audio, image, and video with their large size as the main feature; therefore, it is complicated to manage them since the growth of multimedia content causes increases in the response time and execution cost of the queries; for this reason, it is necessary to apply a fragmentation technique to improve their management (Castro-Medina et al., 2020). Fragmentation is a widely used design technique in multimedia databases, which benefits are reduced response times and lower execution costs of the queries (Saad et al., 2006); there are three types of fragmentation, vertical, horizontal, and hybrid. Existing methods in the literature focus either on horizontal or vertical fragmentation, but most are not able to adapt the fragmentation scheme to database access patterns, that is because they are static.

This article presents the results of the analysis of 34 hybrid fragmentation methods to find if there exists one complete and easy to implement that considers multimedia data, dynamic fragmentation, and a cost model to obtain its fragmentation scheme. In addition, the workflow and cost model of a hybrid fragmentation method for multimedia databases is proposed that produces new schemes when sufficient changes in

*Autor para la correspondencia: msc.ocrescencio@ito-depi.edu.mx
Correo electrónico: msc.ocrescencio@ito-depi.edu.mx (Oscar Crescencio-Rico), lrodriguezm@ito-depi.edu.mx (Lisbeth Rodríguez-Mazahua), dci.fcastro@ito-depi.edu.mx (Felipe Castro-Medina), giner.ah@orizaba.tecnm.mx (Giner Alor-Hernández), jlsanchez@conacyt.mx (J.L. Sánchez-Cervantes).

Historial del manuscrito: recibido el 28/03/2023, última versión-revisada recibida el 30/05/2023, aceptado el 07/06/2023 publicado el 11/09/2023. **DOI:** https://doi.org/10.29057/icbi.v11iEspecial2.10716

workload occur, to ensure the good performance of queries at all times.

The rest of this work is organized as follows: Section 2 addresses the research methodology and comparative analysis of hybrid fragmentation methods. Section 3 presents the workflow of the proposed dynamic hybrid fragmentation technique for multimedia databases. Finally, Section 4 sets out the conclusions and future work.

2. Comparative Analysis of Articles

The following subsections provide an in-depth analysis of the state of the art. The first subsection includes the methodology used and the study of each article. The second subsection deals with the classification of the papers considering the publisher and the year of publication. The third subsection shows if they employed a benchmark. The fourth subsection addresses completeness and ease of implementation. The fifth subsection indicates the number of articles implementing dynamic or static hybrid fragmentation. The sixth subsection presents the types of databases utilized, and finally, the seventh subsection describes if the paper includes content-based queries.

2.1. Research Methodology

Based on the articles published in the main scientific publishers, an analysis was carried out using the methodology proposed in Figure 1, to classify and observe the works that relate to the proposed topic, to answer the following question: Is there a dynamic hybrid fragmentation method for multimedia databases based on a cost model, complete and easy to implement?

Figure 1 shows the process used to select and analyze each article; this process has three steps. The first step consisted of searching in the most prominent digital libraries of scientific publishers with an extensive bibliography of relevant research on hybrid fragmentation published from 2010 to 2022. The digital libraries considered were: 1) ACM Digital Library, 2) IEEE Xplore Digital Library, 3) ScienceDirect (Elsevier), and 4) SpringerLink.

In the second step, a keyword-based search was used to select the most relevant articles. The keywords were: 1) Hybrid Fragmentation, 2) Mixed Fragmentation, 3) Hybrid Partitioning, and 4) Mixed Partitioning. Publications that did not meet these characteristics and papers that were not suitable for the study were discarded. The following sentence establishes the criteria considered for the omission of a research paper:

- Unpublished working papers, non-peer-reviewed articles, non-English language articles, textbooks, and Master and Doctoral dissertations.

The selection process resulted in 34 articles, the papers were grouped by publisher and year, and later analyzed by seven characteristics. The properties considered are 1) Benchmark: Standard database used to validate the method, 2) Completeness: If the paper has all the information needed to implement the method, 3) Ease of implementation: Whether the method is easy to develop or implement, 4) Cost model: If the method uses a cost model to determine a hybrid fragmentation scheme, 5) Dynamic or static: Whether the method can adapt the scheme according to changes in access patterns or database elements (attributes and/or tuples) are assigned to a fragment only once at the time of creation, and their locations are never changed, 6) Repository type: Type of storage on which the research or method is focused, and 7) Content-based query: If the method considers the retrieval of multimedia objects by content.

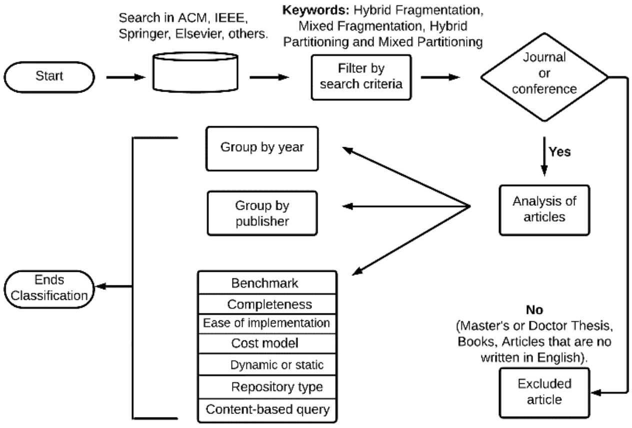
Search in ACM, IEEE, Springer, Elsevier, others.
Keywords: Hybrid Fragmentation, Mixed Fragmentation, Hybrid Partitioning and Mixed Partitioning
Start
Filter by search criteria
Journal or conference
Group by year
Yes
Group by publisher
Analysis of articles
Ends Classification
Benchmark
Completeness
Ease of implementation
Cost model
Dynamic or static
Repository type
Content-based query
No
(Master's or Doctor Thesis, Books, Articles that are no written in English).
Excluded article

Figure 1: Selection criteria flow diagram.

Table 1: describes the record of the articles and compares all the papers found in the digital libraries. The selected articles were evaluated using the methodology shown in Figure 1 to determine whether they have the desired characteristics.

Some of these meets most of the qualities (Jindal & Dittrich, 2012; Patel et al., 2021; Wang et al., 2014), while, (Rodríguez-Mazahua et al., 2016) considers four characteristics but no dynamic fragmentation.

Table 1: Comparison of related works.

Article	1	2	3	4	5
Chbeir & Laurent (2010)	X				X
Kling et al. (2011)	X		X		
Jindal & Dittrich (2012)	X	X	X	X	
Gorla et al. (2012)	X		X		
Song & Chen (2013)	X	X	X		
Chen et al. (2013)	X		X		
Wang et al. (2014)	X	X	X	X	
Kechar & Nait Bahloul (2014)	X				
Chen et al. (2015)	X		X	X	
Harikumar & Ramachandran (2015)	X				
Al-Kateb et al. (2016)			X		
Rodríguez-Mazahua et al. (2016)	X	X	X		X
Padiya et al. (2016)	X	X	X		
Sun et al. (2016)			X		
Rani et al. (2017)	X	X		X	
Mourão & Magalhães (2018)					X
Durand et al. (2018)			X		
Koong et al. (2018)		X			

1) Completeness, 2) Ease of implementation, 3) Cost model, 4) Dynamic fragmentation, and 5) Multimedia database.

Table 1: Comparison of related works (continued).

Article	1	2	3	4	5
Schreiner et al. (2018)					
Vogt et al. (2018)		X	X	X	
Chawla et al. (2019)	X				
Awad et al. (2019)		X			
Schreiner et al. (2019)	X	X			
Badran et al. (2020)					
Pinnecke et al. (2020)			X		
Kulba & Somov (2020)			X	X	
Noraziah et al. (2021)	X			X	
Kang et al. (2021)	X		X	X	
Badran et al. (2021)	X	X			
Azila et al. (2021)		X			
Patel et al. (2021)	X	X	X		X
Ahmed & Alluhaibi (2022)	X		X		
Safaei (2022)	X	X			X
Cantini et al. (2022)	X				
This work	X	X	X	X	X

1) Completeness, 2) Ease of implementation, 3) Cost model, 4) Dynamic fragmentation, and 5) Multimedia database.

2.2. *Classification of Articles by Publisher and Year of Publication*

Figure 2 shows the number of articles by year of publication. The graph shows that the highest number of papers (19) is between the years 2018 to 2022, where 32% used a benchmark, only 47% are complete and 42% are easy to implement, while 37% are based on a cost model, in 21% the fragmentation is dynamic, and 16% used multimedia data.

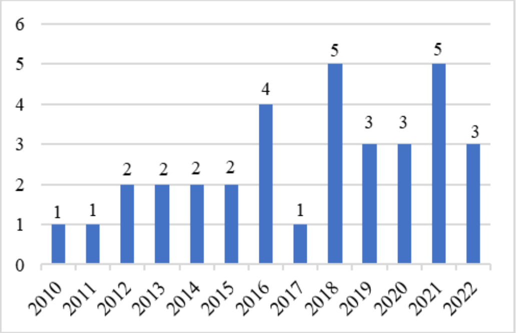

Figure 2: Number of articles per year.

Figure 3 shows a comparison of articles by the publisher, where most articles were published by Springer compared with ACM, IEEE, and Others.

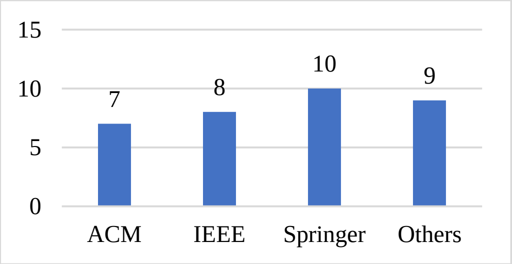

Figure 3: Number of articles per publisher.

In Springer's case, only one work considered both static and dynamic fragmentation, Rani et al. (2017), while the main feature that stands out for Gorla et al. (2012); Jindal & Dittrich (2012); Kling et al. (2011); Patel et al. (2021), and Song & Chen (2013) is that they used a cost model. Of the articles classified in the "Others" category, two used multimedia data (Chbeir & Laurent, 2010; Rodríguez-Mazahua et al., 2016); in contrast, Chen et al. (2015) and Wang et al. (2014) consider a dynamic fragmentation, while Azila et al. (2021); Cantini et al. (2022), and Koong et al. (2018) only meet one criterion. On the other hand, with respect to papers published in IEEE, Awad et al. (2019); Badran et al. (2021), and Chen et al. (2013) do not meet the features of completeness, easy to implement and cost model together, i.e., they have from zero to two criteria, and only Noraziah et al. (2021) and Vogt et al. (2018) are dynamic.

Finally, in ACM different databases to traditional were applied without taking into account multimedia data (Durand et al., 2018; Padiya et al., 2016); for example: in Schreiner et al. (2019) the type of database used was NewSQL.

2.3. *Benchmark*

Figure 4 presents that 16 articles applied benchmarks such as TPC-C (Transaction Processing Performance Council), and TPC-H, to mention a few, for example, Jindal & Dittrich (2012), Wang et al. (2014), and Kang et al. (2021). The rest of the work used their databases to validate their methods. Subsequently, Figure 5 compares the number of articles per type of benchmark, where we found that in the methods that used TPC-C and TPC-H, the fragmentation type was dynamic. While 18 articles do not use some type of benchmark. Besides, Figure 5 shows more detail about the benchmarks, where in some cases the authors used two, for instance, Jindal & Dittrich (2012) focused on TPC-H and SSB (Star Schema Benchmark).

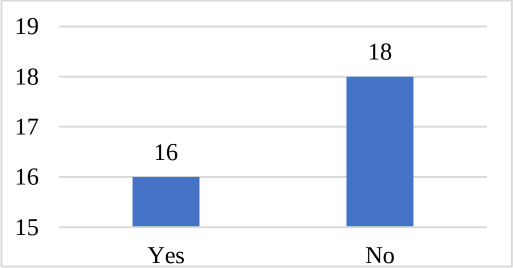

Figure 4: Number of articles that used benchmark.

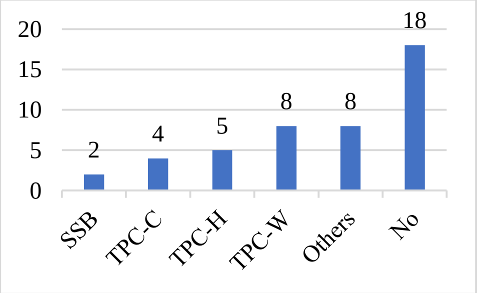

Figure 5: Number of articles per type of benchmark.

2.4. Completeness and Ease of Implementation

To consider a work complete and easy to implement, the following two criteria were established: 1) the article presents the algorithm, technologies, and results obtained at the time of its implementation, and 2) is easy to understand for its replication. The graphs of Figure 6 and Figure 7 show the results based on this classification, where in the first one it is observed that 22 works out of 34 are complete because they show the whole process and the information required for its replication, while, in the second graph 14 out of 34 are easy to implement because they do not require specialized knowledge for their replication, for example, advanced machine learning technics.

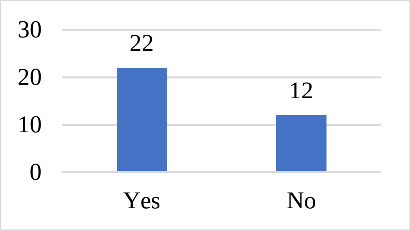

Figure 6: Number of articles per completeness.

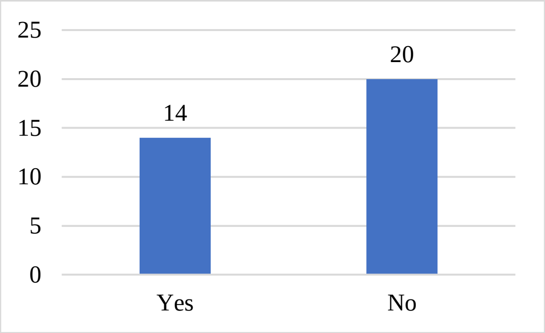

Figure 7: Number of articles per ease of implementation.

2.5. Cost Model

Works that considered a cost model focused on estimating query processing time, disk access, and communication between nodes, to evaluate system performance at the time of fragmentation. For example, in Rodríguez-Mazahua et al. (2016) a cost model was used to evaluate hybrid partition schemes in multimedia databases. Figure 8 shows the works that implemented a cost model in their method, where only 18 use it, e.g., Vogt et al. (2018) and Patel et al. (2021).

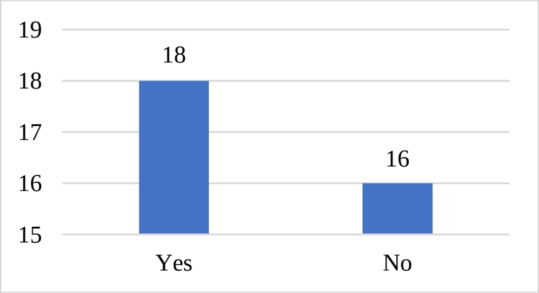

Figure 8: Number of articles per cost model.

2.6. Dynamic or Static

The graph of Figure 9 shows a comparison of the number of articles based on the type of technique (static or dynamic) used in the works. The technics that are capable of adapting the hybrid fragmentation scheme to the workload correspond to dynamic methods (Chen et al., 2015; Jindal & Dittrich, 2012; Kang et al., 2021; Kulba & Somov, 2020; Noraziah et al., 2021; Rani et al., 2017; Vogt et al., 2018; and Wang et al., 2014).

When fragmentation does not consider access pattern changes is considered static, as in Chbeir & Laurent, (2010); Mourão & Magalhães, (2018); Rodríguez-Mazahua et al., (2016); Safaei, (2022), and Song & Chen, (2013). In particular, Rani et al., (2017) is a special case because in this work both static and dynamic cases are applied.

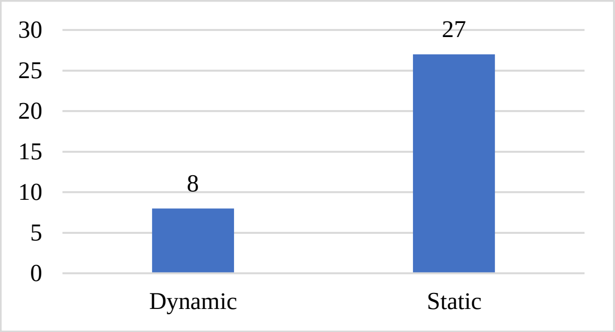

Figure 9: Number of articles per dynamic or static technique.

2.7. Repository Type

The database type used in each work is shown in Figure 10. As noted, the most widely used databases were relational, only six out of these 17 works (Chen et al., 2015; Jindal & Dittrich, 2012; Kang et al., 2021; Noraziah et al., 2021; Rani et al., 2017; and Wang et al., 2014) used dynamic fragmentation. In contrast, two meet all criteria (Jindal & Dittrich, 2012; Wang et al., 2014) but do not use multimedia data. While the works that are not focused on relational databases only consider that the scheme is static, four take into account cost models (Chen et al., 2013; Gorla et al., 2012; Padiya et al., 2016; Song & Chen, 2013), the rest (Awad et al., 2019; Badran et al., 2021; Safaei, 2022) met from zero to three criteria. As mentioned in section 2 (Research Methodology) the analysis process resulted in 34 articles, however, in Figure 10 it is observed that according to the type of repository, the summation is 44 works, because some methods were applied to more than one database type, e.g., relational and distributed databases.

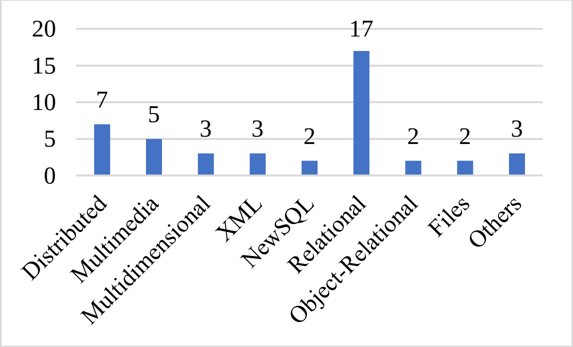

Figure 10: Number of articles per repository type.

2.8. Content-based Query

Most papers did not consider content-based queries as can be observed in Figure 11. Only two make use of this quality: Chbeir & Laurent (2010) and Mourão & Magalhães (2018), nevertheless, although they contemplated multimedia data, they did not meet the required criteria.

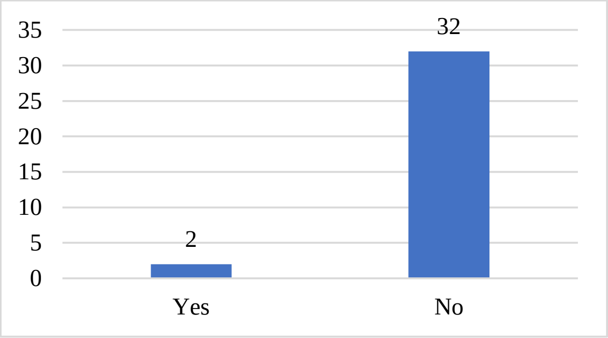

Figure 11: Number of articles per content-based query.

3. Workflow for the dynamic hybrid fragmentation method

Figure 12 presents the workflow for the dynamic hybrid fragmentation method for multimedia databases. The first step is to get the horizontal and vertical fragmentation schemes. The horizontal-vertical scheme is then created respecting horizontal allocations, and the vertical-horizontal scheme is created respecting vertical allocations. Then, these schemes are compared to choose the most inexpensive to apply in the multimedia database.

When the scheme is applied, the permanent log analysis (the dynamic part) starts internally, which will stop when the database administrator (DBA) deems it necessary.

To perform dynamic fragmentation, the following steps will be performed:

1. The operation threshold with an initial value that the DBA deems appropriate will be evaluated, if the threshold is not reached, no changes will be made, and otherwise, step two will be initiated.
2. The cost analysis of the operations in the current scheme will be performed and it continue with step three.
3. The performance threshold will be evaluated, if this threshold is not exceeded by the results obtained from the cost analysis, no change will be made, and otherwise, step four will be initiated.
4. A re-fragmentation will be performed and the mapping of fragments over the current schema.
5. Finally, the observer-fragmenters are updated with the new schema on the child nodes and first step begins again.

Figure 13 shows the process to perform the hybrid fragmentation in more detail. The joining of horizontal and vertical fragmentation results in hybrid fragmentation.

The hybrid fragmentation workflow of Figure 13 is made up as follows:

The two types of hybrid fragmentation are horizontal-vertical and vertical-horizontal; however, only one type of approach will be used. To carry out the selection, the two types of fragmentation will be compared to choose the lowest cost scheme as shown in Figure 12.

First, the horizontal fragmentation process is shown, consisting of two steps, which are mentioned and described below:

Step 1. Operation Cost Matrix Creation: This is an array that shows the cost of operations performed at a given site under a certain predicate. The cost of operations (*co*) is determined by the sum of the values of the operations (*vo*) performed as shown in (1).

$$co = \sum_{u=1}^{H} vo \qquad (1)$$

Where:
u = each different predicate used in the table or fragment.
H = total of predicates.

The equation (2) is used to obtain the value of operations (*vo*), as seen by multiplying the number of predicate-related tuples (*tMax*) by the assigned value of operations (*VAO*) by the remote value (*RV*).

RV takes the value of 1 when the operation is local and 2 when it is processed from a different site of that containing the fragment.

$$vo = tMax * VAO * RV \qquad (2)$$

The assigned value of operations (VAO) is presented in Table 2.

Table 2: Values for each type of operation.

Type of Operation	Value
Create	2
Delete	2
Update	3
Read	1

The cost of each predicate is determined by the sum of the operating cost present in each site. The cost of operations per site is calculated by the sum of all operating costs in each predicate.

Step 2: Creation of the ALP (Attribute Locality Precedence) Table: The ALP table contains the cost of each attribute (Ca), determined by the sum of the costs of the predicates that contain the attribute. The horizontal scheme is obtained by the predicates of the attribute with the highest ALP value.

At the end of these steps, the horizontal fragmentation scheme is created.

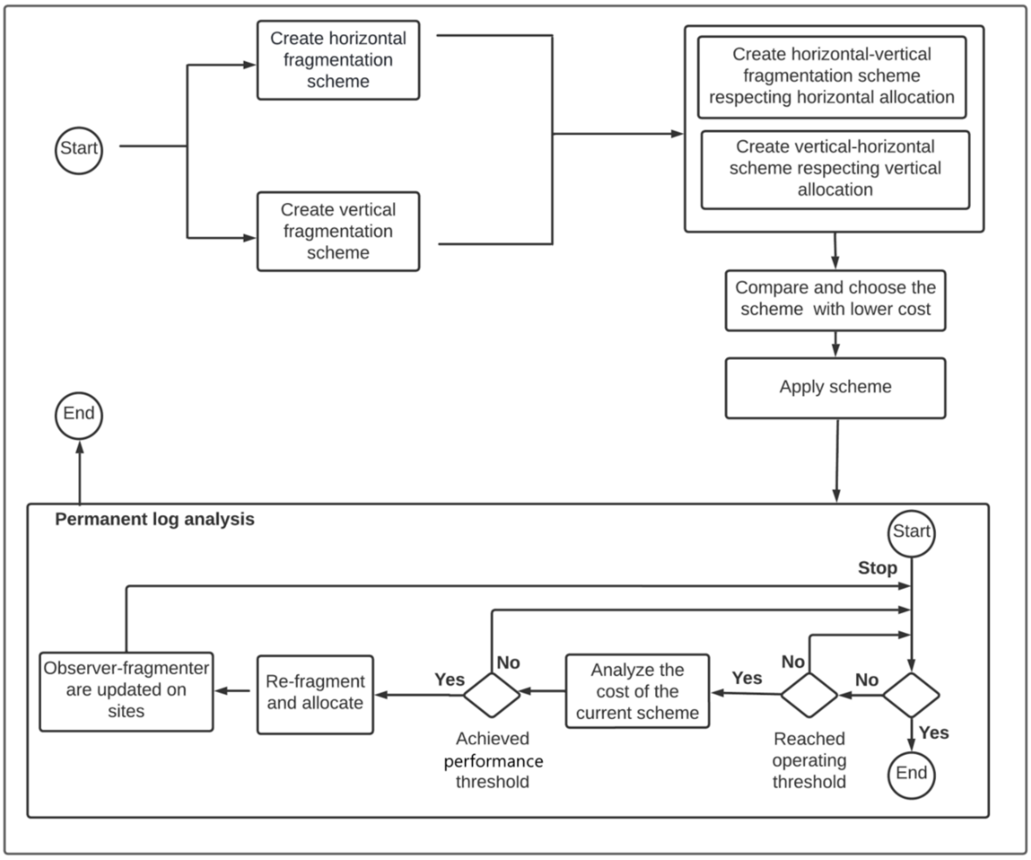
Start
Create horizontal fragmentation scheme
Create vertical fragmentation scheme
Create horizontal-vertical fragmentation scheme respecting horizontal allocation
Create vertical-horizontal scheme respecting vertical allocation
Compare and choose the scheme with lower cost
Apply scheme
End
Permanent log analysis
Start
Stop
Observer-fragmenter are updated on sites
Re-fragment and allocate
Yes
No
Achieved performance threshold
Analyze the cost of the current scheme
Yes
No
Reached operating threshold
No
Yes
End

Figure 12: Dynamic hybrid fragmentation workflow for multimedia databases.

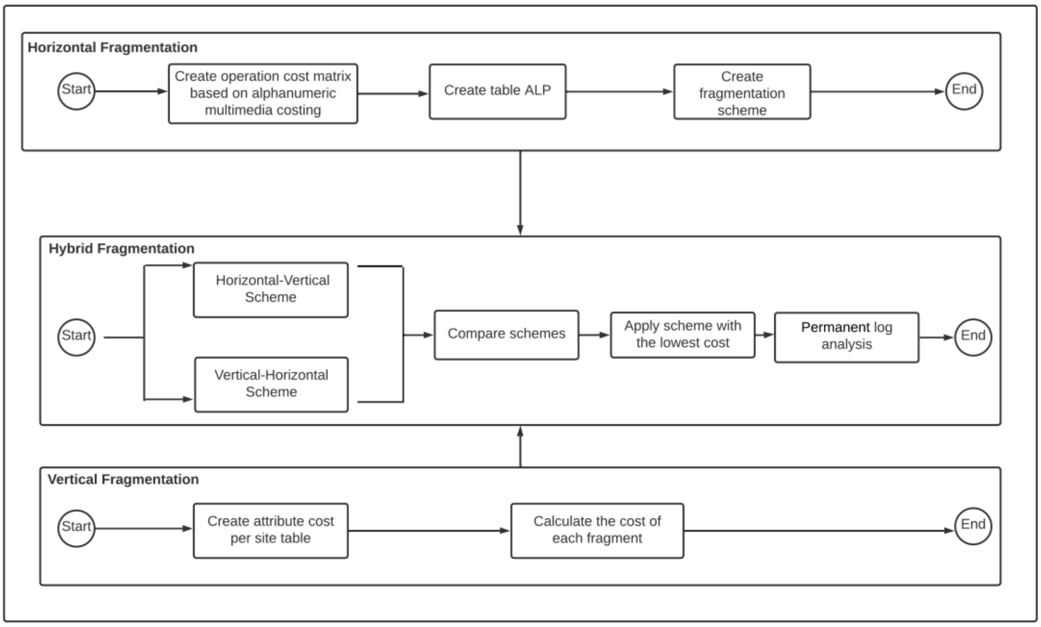
Horizontal Fragmentation
Start
Create operation cost matrix based on alphanumeric multimedia costing
Create table ALP
Create fragmentation scheme
End
Hybrid Fragmentation
Start
Horizontal-Vertical Scheme
Vertical-Horizontal Scheme
Compare schemes
Apply scheme with the lowest cost
Permanent log analysis
End
Vertical Fragmentation
Start
Create attribute cost per site table
Calculate the cost of each fragment
End

Figure 13: Dynamic hybrid fragmentation process flow for multimedia databases.

Secondly, vertical fragmentation is performed (bottom of Figure 13). The attribute cost table is created, which contains information about in what site every attribute is most required by (3) multiplying the size of the attribute (*Sa*) by the value of operation (*VAO*) and the join cost (*Cj*) which when the operations use attributes that are not present in the same table, this takes the value *n* + 1, where *n* is the number of joins.

$$Ca = \sum_{u=1}^{H} VAO_u * \ Cj_u * VR_u * Sa \qquad (3)$$

The cost of each fragment is then calculated by adding up the cost for each attribute, and finally, the scheme of fragments per site is created.

Finally, the process for hybrid fragmentation is in the middle part of the diagram, when both the Horizontal-Vertical and Vertical-Horizontal schemes are created, finally the lowest cost scheme is selected and applied. The dynamic part of hybrid fragmentation begins with the permanent log analysis as described in Figure 12.

4. Conclusions

According to the comparative analysis of the state-of-the-art, it was observed that only 15% of the proposed fragmentation methods take into account multimedia data, in addition, most focus on the static partitioning so, the fragmentation scheme is not adapted to the changes in access patterns; there are very few works that address the dynamic part (24%). Therefore, this project is of utmost importance and provides a contribution to the state-of-the-art in the area of databases, in this way, researchers will have a new technique that they can use to compare it with their proposals; and from this, further research in dynamic hybrid fragmentation will be carried out. It is important to note that the multimedia databases, in addition to containing text, also include videos and audio, so they are accessed when querying these databases increasing response time and execution cost of the queries, so, it is necessary to use a fragmentation technique to achieve efficient retrieval of multimedia objects.

This paper presented the research methodology described in Section 2, and the workflow of a dynamic hybrid fragmentation technique whose goal is to reduce response time and execution cost of the queries in multimedia databases. The dynamic fragmentation method is able to adapt to new queries, as the scheme will be continuously modified based on operations. In the future, the proposed dynamic hybrid fragmentation method will be implemented in multimedia databases to efficiently and effectively partition and allocate the fragments when sufficient changes in workload occur, to ensure the good performance of queries at all times.

For the validation of the proposed method in the future, a case study will be used through the HITO database (History of the Technological Institute of Orizaba). Finally, the method and cost model will be compared with a selected algorithm and model from the comparative analysis.

Acknowledgments

The authors are very grateful to the National Technological of Mexico (TecNM) for supporting this work. This research paper was sponsored by the National Council of Science and Technology (CONACYT).

References

Ahmed, Z. J., & Alluhaibi, S. T. (2022). Hybrid Data Fragmentation Using Genetic Killer Whale Optimization-Based Clustering Model. *Journal of Pharmaceutical Negative Results*, *13*(SO2). https://doi.org/10.47750/pnr.2022.13.s02.39

Al-Kateb, M., Sinclair, P., Au, G., & Ballinger, C. (2016). Hybrid row-column partitioning in teradata ®. *Proc. of the VLDB Endowment*, *9*(13), 1353–1364. https://doi.org/10.14778/3007263.3007273

Awad, A. S., Yousif, A., & Kadoda, G. (2019). Enhanced Model for Cloud Data Security based on Searchable Encryption and Hybrid Fragmentation. *2019 Int. Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, 1–4. https://doi.org/10.1109/ICCCEEE46830.2019.9070918

Azila, A., Fauzi, C., Fariza, W., Rahman, W. A., Fauzi, A., & Weigelt, F. (2021). Managing Fragmented Database in Distributed Database Environment. In *Journal of Mathematics and Computing Science* (Vol. 7, Issue 1).

Badran, S., Arman, N., & Farajallah, M. (2020, November 28). Towards a hybrid data partitioning technique for secure data outsourcing. *Proc. - 2020 21st Int. Arab Conference on Information Technology, ACIT 2020*. https://doi.org/10.1109/ACIT50332.2020.9300064

Badran, S., Arman, N., & Farajallah, M. (2021). An Efficient Approach for Secure Data Outsourcing using Hybrid Data Partitioning. *2021 Int. Conference on Information Technology, ICIT 2021 - Proceedings*, 418–423. https://doi.org/10.1109/ICIT52682.2021.9491745

Cantini, R., Marozzo, F., Orsino, A., Talia, D., Trunfio, P., Badia, R. M., Ejarque, J., & Vazquez, F. (2022). *Block size estimation for data partitioning in HPC applications using machine learning techniques*. http://arxiv.org/abs/2211.10819

Castro-Medina, F., Rodríguez-Mazahua, L., López-Chau, A., Cervantes, J., Alor-Hernández, G., & Machorro-Cano, I. (2020). Application of dynamic fragmentation methods in multimedia databases: A review. In *Entropy* (Vol. 22, Issue 12, pp. 1–25). MDPI AG. https://doi.org/10.3390/e22121352

Chawla, T., Singh, G., & Pilli, E. S. (2019). HYPSO: Hybrid partitioning for big RDF storage and query processing. *ACM International Conference Proceeding Series*, 188–194. https://doi.org/10.1145/3297001.3297025

Chbeir, R., & Laurent, D. (2010). Enhancing Multimedia Data Fragmentation. In *Journal of Multimedia Processing and Technologies* (Vol. 1, Issue 2).

Chen, K., Zhou, Y., & Cao, Y. (2015). Online data partitioning in distributed database systems. *EDBT 2015 - 18th International Conference on Extending Database Technology, Proceedings*, 1–12. https://doi.org/10.5441/002/edbt.2015.02

Chen, S., Chi, C. H., Ding, C., & Wong, R. K. (2013). Data decomposition based partial replication model for software services. *Proceedings - IEEE 10th International Conference on Services Computing, SCC 2013*, 256–263. https://doi.org/10.1109/SCC.2013.83

Durand, G. C., Pinnecke, M., Piriyev, R., Mohsen, M., Broneske, D., Saake, G., Sekeran, M. S., Rodriguez, F., & Balami, L. (2018). GridFormation: Towards Self-Driven Online Data Partitioning using Reinforcement Learning. *Proc. of the First Int. Workshop on Exploiting Artificial Intelligence Techniques for Data Management*, 1–7. https://doi.org/10.1145/3211954.3211956

Gorla, N., Ng, V., & Law, D. M. (2012). Improving database performance with a mixed fragmentation design. *Journal of Intelligent Information Systems*, *39*(3), 559–576. https://doi.org/10.1007/s10844-012-0203-x

Harikumar, S., & Ramachandran, R. (2015, April 21). Hybridized fragmentation of very large databases using clustering. *2015 IEEE Int. Conference on Signal Processing, Informatics, Communication and Energy Systems, SPICES 2015*. https://doi.org/10.1109/SPICES.2015.7091488

Jindal, A., & Dittrich, J. (2012). *Relax and Let the Database Do the Partitioning Online* (Vol. 126, pp. 65–80). Springer. https://doi.org/10.1007/978-3-642-33500-6_5

Kang, D., Jiang, R., & Blanas, S. (2021). Jigsaw: A Data Storage and Query Processing Engine for Irregular Table Partitioning. *Proc. of the ACM SIGMOD Int. Conference on Management of Data*, 898–911. https://doi.org/10.1145/3448016.3457547

Kechar, M., & Nait Bahloul, S. (2014). *Hybrid Fragmentation of XML Data Warehouse Using K-MEANS Algorithm*.

Kling, P., Özsu, M. T., & Daudjee, K. (2011). Scaling XML query processing: Distribution, localization and pruning. *Distributed and Parallel Databases*, *29*(5–6), 445–490. https://doi.org/10.1007/s10619-011-7085-8

Koong, K.-L., Haw, S.-C., & Soon, L.-K. (2018). Labeling-Based Hybrid XML Fragmentation Model. *Advanced Science Letters*, *24*(2), 1177–1181. https://doi.org/10.1166/asl.2018.10711

Kulba, V., & Somov, S. (2020). Dynamic Fragment Allocation in Distributed System with Time-Varying Parameters of its Operation. *2020 13th Int. Conference "Management of Large-Scale System Development" (MLSD)*, 1–4. https://doi.org/10.1109/MLSD49919.2020.9247671

Mourão, A., & Magalhães, J. (2018). Balancing search space partitions by sparse coding for distributed redundant media indexing and retrieval. *Int. Journal of Multimedia Information Retrieval*, *7*(1), 57–70. https://doi.org/10.1007/s13735-017-0140-0

Noraziah, A., Fauzi, A. A. C., Ubaidillah, S. H. S. A., Alkazemi, B., & Odili, J. B. (2021). BVAGQ-AR for Fragmented Database Replication Management. *IEEE Access*, *9*, 56168–56177. https://doi.org/10.1109/ACCESS.2021.3065944

Padiya, T., Kanwar, J. J., & Bhise, M. (2016). Workload aware hybrid partitioning. *ACM Int. Conference Proc. Series*, *21-23-October-2016*, 51–58. https://doi.org/10.1145/2998476.2998479

Patel, M., Yadav, N., & Bhise, M. (2021). *Workload aware Cost-based Partial loading of Raw data for Limited Storage Resources*. https://doi.org/https://www.researchgate.net/publication/357050760

Pinnecke, M., Campero Durand, G., Broneske, D., Zoun, R., & Saake, G. (2020). GridTables: A One-Size-Fits-Most H2TAP Data Store. *Datenbank-Spektrum*, *20*(1), 43–56. https://doi.org/10.1007/s13222-019-00330-x

Rani, S., Koshley, D. K., & Halder, R. (2017). Partitioning-insensitive watermarking approach for distributed relational databases. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10720 LNCS*, 172–192. https://doi.org/10.1007/978-3-662-56266-6_8

Rodríguez-Mazahua, L., Alor-Hernández, G., Cervantes, J., López-Chau, A., & Sánchez-Cervantes, J. L. (2016). A hybrid partitioning method for multimedia databases. *DYNA*, *83*(198), 59. https://doi.org/10.15446/dyna.v83n198.50507

Saad, S., Tekli, J., Chbeir, R., & Yetongnon, K. (2006). Towards Multimedia Fragmentation. In *Advances in Databases and Information Systems; Lecture Notes in Computer Science; Manolopoulos, Y., Pokorny, J., Sellis, T.K., Eds* (Vol. 4152, pp. 415–429). Springer. https://doi.org/10.1007/11827252_31

Safaei, A. A. (2022). Hybrid fragmentation of medical images' attributes for multidimensional indexing. *Cluster Computing*, *25*(1), 215–230. https://doi.org/10.1007/s10586-021-03356-7

Schreiner, G. A., Duarte, D., Bianco, G. D., & dos Santos Mello, R. (2019). A Hybrid Partitioning Strategy for NewSQL Databases: The VoltDB Case. *ACM Int. Conference Proc. Series*, 353–360. https://doi.org/https://doi.org/10.1145/3366030.3366062

Schreiner, G. A., Duarte, D., Ronaldo, O. :, & Mello, S. (2018). *An autonomous hybrid data partition for NewSQL DBs*.

Song, S., & Chen, L. (2013). Indexing dataspaces with partitions. *World Wide Web*, *16*(2), 141–170. https://doi.org/10.1007/s11280-012-0163-7

Sun, L., Franklin, M. J., Wang, J., & Wu, E. (2016). Skipping-oriented partitioning for columnar layouts. *Proc. of the VLDB Endowment*, *10*(4), 421–432. https://doi.org/10.14778/3025111.3025123

Vogt, M., Stiemer, A., & Schuldt, H. (2018). Polypheny-DB: Towards a Distributed and Self-Adaptive Polystore. *2018 IEEE Int. Conference on Big Data (Big Data)*, 3364–3373. https://doi.org/10.1109/BigData.2018.8622353

Wang, X., Fan, X., Chen, J., & Du, X. (2014). Automatic Data Distribution in Large-scale OLTP Applications. *Int. Journal of Database Theory and Application*, *7*(4), 37–46. https://doi.org/10.14257/ijdta.2014.7.4.04