Query Optimization on Relational Database System in Cloud Environment

Md. Shifatul Ahsan Apurba
Dept. of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
apurbaahsan@gmail.com

Md. Ramim Ul Haq
Dept. of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
ramimmd1@gmail.com

Rafa Siddiqua
Dept. of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
rafa.siddiqua@g.bracu.ac.bd

Ehsanur Rahman Rhythm
Dept. of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
ehsanur.rahman.rhythm@g.bracu.ac.bd

Md Humaion Kabir Mehedi
Dept. of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
humaion.kabir.mehedi@g.bracu.ac.bd

Annajiat Alim Rasel
Dept. of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
annajiat@gmail.com

***Abstract*—The primary objective of the research paper is to minimize query response time while concurrently reducing database server throughput in a cloud computing environment. To achieve this, the research aims to integrate machine learning techniques, optimize real-time query processing, develop scalable and cost-effective query optimization methods, and facilitate collaborative optimization of queries and intelligent indexing. The literature survey showcases several methods that can improve query processor performance in a cloud setting, such as the key-value data model, MapReduce, iterative dynamic programming, efficient sets of query execution plans, and resource algorithms. The query optimization process necessitates an analysis of queries and data, to streamline queries to reduce the time and resources required for processing. Nonetheless, this research is constrained by the difficulty in obtaining primary data.**

***Index Terms*—Query, Cloud Computing, Optimization, MapReduce**

I. Introduction

In recent years, the popularity of cloud computing has grown exponentially, with more and more businesses opting to use cloud-based solutions to manage their data. Cloud computing allows users to access computing resources, including storage, processing power, and software, on-demand and via the internet. It provides a flexible environment in which users can load data, run queries, and scale resources as needed. This has led to the emergence of cloud databases, which offer numerous benefits over traditional database management systems.

Database Management systems represent the relationship between data items and their relationship. The traditional DBMS is known for storing data row-by-row [1]. However, using a cloud database, users can manage all that data across different databases efficiently. Different types of database systems have been invented like column-oriented, key-value store, and MapReduce [2] as different relational databases are not well suited for large cluster servers as they are not designed to be distributed.

Cloud databases are ideal for read-intensive Data Warehouse applications. As data in these applications grows rapidly, the cloud provides a dynamically scalable environment. This enables users to rent large amounts of resources for a short period to run complex queries efficiently on large amounts of data. Cloud computing allows resources to be acquired and released automatically and quickly at runtime to ensure the Service Level Agreements between customers and cloud service providers are met.

One of the main benefits of using a cloud environment for query optimization is cost savings along with loss prevention. Cloud database providers offer regular backups for the stored data, which are stored in different locations to prevent a potential disaster [3]. This ensures that even in the event of a disaster, the data is stored safely. Additionally, cloud databases increase the scalability of data optimization, allowing users to scale up or down as per their needs. This makes cloud databases highly adaptable to users' conditions, regardless of business and database size.

Furthermore, there is no risk when growing the database as users can turn it off and eliminate costs during downtime. Cloud databases provide a cost-effective way to manage data as they eliminate the need to purchase and maintain expensive hardware and software. They also offer a flexible and scalable environment that can be customized to meet specific business needs.

II. Overview

Query optimization is a critical process that aims to improve the performance and efficiency of database queries in a cloud computing environment [4]. It involves analyzing queries and data to optimize them and reduce the resources and time required to process them. In this regard, dynamic resource scaling is a crucial aspect of query optimization, enabling the system to optimize resource usage by provisioning and de-provisioning resources such as CPU, memory, and storage as needed [3].

Databases for cloud computing are designed to handle massive amounts of data and support high levels of concurrency through distributed architectures [5]. These architectures enable data to be stored across multiple nodes or servers, requiring query optimization for parallel processing and reducing network traffic. Cloud computing environments offer various tools and services, such as performance monitoring and analysis, data analysis, and machine learning algorithms, to optimize queries based on patterns in the data.

The techniques used for query optimization in cloud computing include partitioning, indexing, caching, query rewriting, and load balancing. Partitioning involves the distribution of data across multiple nodes or servers to enhance query performance by minimizing network traffic. Indexing involves the creation of indexes on frequently queried columns to expedite data retrieval. Caching involves the caching of frequently accessed data in memory to reduce disk I/O and improve query performance. Query rewriting is the process of modifying queries to leverage parallel processing and reduce network traffic. Lastly, load balancing distributes queries across multiple nodes or servers to balance the workload and enhance performance.

Since query optimization is an indispensable process for ensuring that database systems in cloud computing environments, it should operate efficiently and effectively while meeting the demands of contemporary data-driven applications.

III. Literature Review

The efficient processing of queries in cloud environments is a critical aspect of database management. In recent years, a range of techniques have been proposed to enhance the performance of query processors in the cloud, each with its strengths and limitations.

The efficient processing of queries in cloud environments is essential for managing databases effectively. Research [6] proposed the use of a pure key-value data model in combination with MapReduce to enhance the scalability and optimization of query processing in cloud environments. MapReduce is a programming model and framework for processing large amounts of raw data in parallel and optimizing the output as key-value pairs. However, this approach has limitations in terms of supporting secondary indexes, range queries, and multidimensional queries. The pure key-value data model implies that all records will be structured as consisting of the number of named columns and values that are arbitrary byte strings. As a result, secondary indexes, range queries, and multidimensional queries are not supported in this model.

Another research [7] proposed the Cloud Global index, a tree-based indexing scheme that offers high scalability, throughput, and availability for cloud storage. This approach can handle a mixed load of queries and updates effectively but only supports one-dimensional queries. In the Cloud Global index, the tree structure is built using a balanced tree algorithm, where each internal node represents a key range and each leaf node represents a file block in the underlying storage system. The Cloud Global index uses several techniques to ensure high scalability, throughput, and availability, such as range partitioning, data replication, and load balancing. To improve query processing further, it has been suggested that queries be applied directly to compressed data.

In addition to improving the efficiency of query processing, there has been considerable research on optimizing queries themselves. Another research [8] proposed using iterative dynamic programming to optimize queries by evaluating common subexpressions of multiple queries only once. This approach has been shown to significantly improve query evaluation performance. The basic idea behind iterative dynamic programming is to break down a query into subqueries and to compute common subexpressions of these subqueries. These common subexpressions can be evaluated only once, and the results can be reused for evaluating multiple subqueries. By reducing the number of redundant computations, iterative dynamic programming can significantly improve query evaluation performance.

A study [9] developed an efficient set of query execution plans based on a single evaluation of common tasks. They proposed generating alternative query plans to find the best query execution plans using robust heuristic algorithms such as branch-and-bound, hill-climbing, and genetic hill-climbing algorithms. The goal of their approach was to minimize the execution time of queries by finding the most efficient query execution plan. The generation of alternative query plans was based on the use of cost models, which estimate the execution time of different query execution plans. The robust heuristic algorithms were used to search the space of possible query execution plans efficiently and to find the best query execution plan.

Another study [10] suggested using resource allocation algorithms to schedule several parallel requests to decrease query response time. They proposed reorganizing the action plan and dividing the query into parallel sub-queries to optimize the reuse of cached data and increase the rate of processing new queries. The basic idea behind their approach was to split a query into sub-queries and execute them in parallel on different processors. Splitting the query into sub-queries makes it possible to reuse previously computed subquery results and increase the rate of processing new queries. The resource allocation algorithms were used to schedule the parallel subqueries effectively and to optimize the reuse of cached data.

In a recent study [11], many academics proposed novel approaches for query optimization in cloud systems. For instance, the authors proposed a unique method of query optimization that uses machine learning to identify the most efficient query execution strategy for a given query workload. The unveiling of a novel algorithm that dynamically alters query execution plans in accordance with the availability of cloud resources. One of the difficulties in query optimization for cloud environments is lower query execution costs.

Many techniques have recommended cost-based query optimization approaches in various research [12] for cloud environments. For instance, a cost-based query optimization method that considers the cost of data transport and computing in the cloud has been presented. The price of data transport, computation, and storage was taken into consideration in a new cost model for cloud environments that was also presented.

A new study [13] has shown that questions must be dispersed among numerous nodes in distributed cloud settings. Numerous research has looked at how to optimize queries in distributed cloud environments. One novel query optimization method, for example, has been proposed that considers the proximity of data across distinct cloud nodes. Similar to this, a new query optimization technique that considers network bandwidth and latency in a distributed cloud environment was presented.

These studies have shown that there are numerous opportunities for improving the performance of query processors in the cloud. However, each approach has its limitations, and future research should focus on developing more scalable and efficient techniques that can handle complex queries and large datasets.

IV. Methodology

Query optimization is a crucial aspect of relational database systems that can have a significant impact on their performance. In cloud environments, where database systems are deployed on distributed computing resources, optimizing queries becomes even more critical due to the added complexity of managing and coordinating resources across multiple nodes. To conduct a comprehensive review of query optimization in cloud-based relational database systems, we will employ a systematic literature review (SLR) methodology [14].

The SLR methodology will involve the following steps. First, we will define the research questions and inclusion/exclusion criteria that will guide our literature search. The research questions will focus on the various approaches and techniques for query optimization in cloud-based relational database systems. The inclusion/exclusion criteria will be used to ensure that only relevant studies are selected for our review.

Next, we will conduct a comprehensive search of academic databases, including ACM Digital Library, IEEE Xplore, and ScienceDirect, among others. We will also conduct a manual search of relevant conference proceedings and journals. The search will be limited to studies published between 2010 and 2022 to ensure that we include the most recent research in our review.

The studies selected for our review will undergo a rigorous screening process. The first screening will be based on the titles and abstracts of the studies, and the second screening will involve a full-text review. The studies that meet our inclusion criteria will be selected for data extraction.

The data extraction process will involve the extraction of relevant data points from the selected studies, including the query optimization techniques employed, the performance metrics used, and the evaluation methods employed. We will also extract information on any novel approaches or techniques that we identify in the studies.

After data extraction, we will perform a qualitative analysis of the studies to identify common themes and trends in query optimization in cloud-based relational database systems. We will also perform a quantitative analysis of the performance metrics reported in the studies to evaluate the effectiveness of the query optimization techniques employed.

To introduce novel approaches, we will conduct an in-depth analysis of the selected studies to identify any gaps in the existing literature. Based on this analysis, we will propose 1-2 novel approaches for query optimization in cloud-based relational database systems. We will provide a detailed description of these approaches, including their theoretical foundations, practical implementation, and expected performance benefits.

Therefore, our methodology will provide a rigorous and systematic approach to reviewing the literature on query optimization in cloud-based relational database systems. Our inclusion of novel approaches will contribute to the existing literature by proposing innovative solutions to address any identified gaps.

V. Result and Findings

The task of processing queries in cloud-based settings is a multifaceted and demanding issue, primarily due to the substantial quantities of data involved and the requirement for prompt and precise processing. Various methodologies have been suggested in scholarly works, with each methodology possessing its own set of advantages and drawbacks. Consequently, there exists a necessity for further investigation in this domain to devise novel methodologies capable of managing the escalating requisites of cloud-centric inquiry processing.

The proposal put forth in [6] paper advocates for the implementation of a pure key-value data model as a viable strategy for augmenting scalability and optimization within cloud-based settings. Nevertheless, its capacity to facilitate secondary indexes, range queries, and multidimensional queries is restricted. Consequently, it may not be appropriate for use in applications that necessitate intricate query processing.

The Cloud Global index, as proposed in the referenced literature [7], presents a solution that exhibits superior scalability, throughput, and availability for cloud storage. Furthermore, it demonstrates the ability to effectively manage a diverse range of query and update operations. The system facilitates unidimensional inquiries and employs methodologies such as range partitioning, data replication, and load balancing to guarantee optimal efficiency. Nevertheless, it might not be appropriate for use cases that necessitate multidimensional inquiries.

The optimization of queries is the main focus of both the iterative dynamic programming approach suggested in [8] and the efficient query execution plan generation approach proposed in [9]. The objective of these methodologies is to decrease the duration of query execution by diminishing the number of repetitive computations. Nevertheless, they might not be appropriate for implementations that necessitate intricate inquiries involving a substantial quantity of sub-inquiries.

The articles center their attention on enhancing performance by means of parallelizing queries, with the former proposing resource allocation algorithms and the latter introducing a machine learning-based approach. The aforementioned methodologies endeavor to enhance the speed at which novel inquiries are processed and optimize the utilization of cached data. Nonetheless, such systems may not be optimal for use cases that necessitate instantaneous processing or management of intricate queries.

A number of studies have put forth query optimization methods based on the cost for cloud environments, as evidenced by the works. The aforementioned methodologies take into account the expenses associated with data transfer, computation, and storage within cloud computing environments, and employ cost modeling techniques to approximate the duration of diverse query execution strategies. The objective is to reduce the expenses incurred during query execution while upholding optimal performance. Nevertheless, these methodologies may not be appropriate for use cases that necessitate instantaneous processing or manage intricate inquiries.

So, according to the existing body of literature, a prospective strategy to enhance query processing in cloud environments could entail the amalgamation of various techniques to achieve superior performance. The integration of the tree-based indexing scheme of the Cloud Global index with a cost-based query optimization approach that takes into account the expenses associated with data transport and computation has the potential to optimize queries in a highly efficient manner, while simultaneously reducing the overall costs associated with query execution.

Moreover, the integration of machine learning algorithms to identify the optimal query execution strategy and dynamically adjust query execution plans based on the availability of cloud resources could potentially augment system performance. In addition, the incorporation of resource allocation algorithms for the purpose of scheduling multiple parallel requests and partitioning the query into parallel sub-queries has the potential to enhance the efficiency of cached data reuse and augment the speed of processing novel queries.

The existing literature indicates that every method has its own advantages and disadvantages, and there is no universal remedy for effective query processing in cloud settings. Henceforth, it is recommended that forthcoming studies concentrate on formulating novel methodologies capable of managing the escalating requisites of cloud-oriented query processing and rectifying the constraints of current methodologies. The aforementioned methodologies ought to possess the ability to be scaled, optimized for efficiency, and proficient in managing intricate inquiries and voluminous data sets. Furthermore, it is imperative that these systems exhibit flexibility in accommodating diverse application scenarios, data structures, and query modalities while taking into account variables such as data dispersion, resource allotment, and network delay to guarantee maximum efficiency.

VI. Limitation and Challenges

There might be several limitations along with potential challenges for this research.

A. Limitation

One limitation of the research on query optimization for a relational database system in a cloud environment is the issue of generalizability. The specific algorithms and approaches used in this research may not apply to other systems, architectures, or environments. The study's findings may require adaptation or modification to be effective in other scenarios.

Another limitation is the focus on a cloud environment. The findings of this research may not apply to non-cloud environments, as cloud environments have unique characteristics that affect query optimization, such as resource sharing and elasticity.

Scalability is another limitation of the research, as the optimization techniques used may not address scalability issues as the size and complexity of the database system and workload increase. Additional research may be necessary to address these scalability issues effectively.

The research's performance metrics may also limit the study's effectiveness, as the selected metrics may not capture all aspects of query performance and may not reflect the needs of different types of users or applications.

Lastly, the optimization of queries in a relational database system can be complex, and the research may not address all aspects of this complexity, such as indexing, partitioning, and query processing.

B. Potential Challenges

Cloud environments are highly heterogeneous, and it can be challenging to develop optimization techniques that work across all types of cloud architectures and platforms. This heterogeneity may present challenges in optimizing queries across different cloud environments.

Another challenge is the issue of resource constraints. The cloud environment may be subject to resource constraints, such as limited processing power or memory, which can limit the effectiveness of optimization techniques that require significant computational resources.

Data distribution is another potential challenge, as data in a cloud environment may be spread across multiple servers and locations. This can make it challenging to optimize queries that involve data from multiple sources.

Security and privacy risks may also present challenges in a cloud environment. Optimization techniques may need to take into account these risks and mitigate them appropriately to ensure data security and privacy.

Lastly, cost considerations may also present a challenge in implementing optimization techniques that require significant computational resources. The cost of implementing and maintaining optimization techniques in a cloud environment may be a barrier to their effectiveness.

VII. Discussion

Cloud-based data management systems offer various optimization techniques for query processing, which can effectively reduce response times and enhance productivity. The process of indexing involves the creation of an index on one or more columns of a database table, with the aim of enhancing the efficiency of query processing. Consequently, the database management system can expedite responses to queries by swiftly retrieving the requisite data. Query optimization involves selecting the optimal query plan from a range of alternatives to effectively process a query, as stated in reference [15]. Cost-based optimization is a prevalent technique utilized for enhancing queries in cloud environments. Caching is a technique that involves the storage of frequently accessed data in memory in order to reduce the time required to access it from the disc. This approach has the potential to expedite query processing by reducing the frequency of required disc accesses. The process of partitioning a sizable database table involves dividing it into smaller partitions based on a specific column or attribute, as stated in reference [16]. Reducing the quantity of required data has the potential to expedite the process of searching for information within large datasets. Compression is a technique utilized to enhance data access speed by reducing the size of the data stored in the database. The acceleration of query processing can be achieved through the utilization of a method that reduces the volume of information that must be transferred from the disc to memory [17]. Parallel processing involves dividing a large query into smaller tasks to enable multiple processors or nodes to execute the tasks concurrently. The implementation of this approach would result in a reduction of the processing time required for a lengthy query, thereby accelerating the query processing. In cloud environments, the optimization of queries is achieved through the utilization of GA-based algorithms. These algorithms employ a collection of query plans as distinct entities and generate novel query plans through the application of crossover and mutation operators. The cost of executing the query plan forms the basis for defining the fitness function. The algorithm continues to generate new query plans until it identifies the optimal one. The phenomenon of collective animal behavior, specifically that of a swarm of birds or fish, has been emulated through the utilization of a population-based optimization algorithm commonly referred to as particle swarm optimization (PSO) [18]. Algorithms based on Particle Swarm Optimisation (PSO) are utilized for optimizing queries by considering a set of query plans as particles. These algorithms employ rules for updating the velocity and position of the particles, thereby facilitating their movement toward the desired outcome. The fitness function is established based on the execution cost of the query plan. Ant colony optimization is an optimization algorithm that is based on population dynamics and was derived from the study of ant colony behavior [19]. Ant Colony Optimisation (ACO) algorithms are utilized in the optimization of queries within cloud environments by employing a set of query plans as ants. These algorithms rely on pheromone trails to communicate the effectiveness of specific ants in executing a given query plan to other ants. The definition of the fitness function is based on the execution cost of the query plan [20]. Simulated annealing is a stochastic optimization algorithm that is inspired by the metallurgical annealing process. In the realm of query optimization within cloud environments, algorithms are based on simulated annealing transition between states by utilizing a collection of query plans and a temperature parameter. The fitness function is established based on the execution cost of the query plan. The algorithm continues to transition to a new state until it identifies the optimal solution. The utilization of a tabu list, which comprises forbidden moves, is a characteristic of the tabu search algorithm, a neighborhood-based optimization technique. This approach is designed to prevent the algorithm from revisiting previously encountered solutions [21]. In cloud environments, query optimization involves the utilization of a collection of forbidden moves and query plans by TS-based algorithms to navigate the search space. The definition of the fitness function is based on the execution cost of the query plan. The algorithm conducts a systematic exploration of the search space until it identifies the optimal solution [22]. Dynamic programming is a mathematical optimization technique that can effectively address intricate problems by breaking them down into simpler sub-problems. DP-based algorithms leverage a collection of sub-problems to enhance the query plan within the framework of query optimization in cloud environments. The algorithm utilizes a predetermined set of rules to address each sub-problem, integrating the outcomes to generate the optimal query plan [23]. The process of selecting an algorithm for query optimization in a cloud environment is subject to various factors, such as the intricacy and size of the data, the intricacy of the query, the available resources, and the optimization goals. It is widely acknowledged that no single algorithm possesses inherent superiority over others; rather, each algorithm exhibits distinct advantages and disadvantages. A cost-based optimization approach is commonly utilized in tandem with these algorithms to assess the expenses of query plans and determine the optimal one. To fully leverage the strengths of individual algorithms, it may be beneficial to adopt a hybrid approach.

VIII. Recommendations

Drawbacks of the query optimization methods used in cloud-based data management systems: According to Chen (2018), query optimization techniques are used in cloud-based data management systems to improve the efficiency and performance of database queries. However, these methods have certain limitations that need to be addressed for better effectiveness. One of the major drawbacks is their inability to adapt to the constantly changing nature of cloud environments where database resources can frequently vary. To be more effective, query optimization methods used in cloud-based data management systems should be able to adjust dynamically to these variations and optimize queries accordingly. Additionally, the lack of support for distributed query processing poses another challenge to these methods. Many cloud-based data management systems rely on distributed data storage and processing to enhance their functionality and scalability [24]. Even though query optimization is used in cloud-based data management systems to enhance database query performance and efficiency, the potential advantages of distributed query processing are not fully utilized in current approaches. The limitations of these techniques, particularly their inability to adjust to the constantly changing cloud environments, where database resources may vary frequently, need to be addressed to enhance their effectiveness. To tackle this issue, query optimization techniques used in cloud-based data management systems should dynamically adapt to these variations and optimize the queries accordingly. In addition, many cloud-based data management systems require distributed data storage and processing to enhance their functionality and scalability, but these methods lack support for distributed query processing [25]. However, the current query optimization methods do not fully utilize the potential benefits of distributed query processing.

Strategies to overcome the difficulties of query optimization in multi-cloud environments: The distributed nature of data and processing resources across multiple cloud providers pose a challenge for query optimization in multi-cloud environments [26]. Traditional query optimization techniques may not be effective, and new strategies are necessary to overcome the difficulties. One possible strategy is to utilize federated query processing, which involves dividing a complex query into smaller sub-queries that can be executed in parallel on different cloud providers. Another approach is to use data partitioning, which involves partitioning data across various cloud providers and optimizing queries for each partition separately [27]. Additional techniques for optimizing queries in cloud computing include query rewriting, wherein a query is transformed into a form that can be better optimized for a particular cloud provider, and query caching, wherein frequently accessed data is stored in memory to minimize query processing [28]. No details are left out when considering these approaches, as they are both effective in improving query performance in cloud computing environments. The tone is formal and informative, conveying expertise in the subject matter.

IX. Future Work

There could be several plans and ideas for the future of query optimization in cloud computing environments. Here are a few:

- Integration with Machine Learning: By analyzing query patterns and recommending optimized execution plans, machine learning can be integrated to automate the query optimization process. To further increase the effectiveness and performance of database systems, machine learning algorithms will become more sophisticated and advanced. As a result, they will be able to learn from new data and query patterns and adapt to them.
- Real-time query optimization in cloud systems: In the future, query optimization tools and services will be able to modify queries in response to changes in workload and resource availability [29]. Despite dynamic and unpredictable environments, this will help to guarantee that queries are always processed as effectively as possible.
- Cloud environment-native query optimization: Cloud-native architectures and technologies, such as serverless computing and micro-services, will make it possible to integrate query optimization directly into the cloud platform, making it simpler and more effective to optimize queries and boost performance.
- Collaborative optimization of query: Multi-database systems collaborate to optimize queries and share resources in a collaborative query optimization model [30]. Particularly in multi-cloud environments where data is dispersed across multiple cloud providers, this can lower resource usage and improve query performance.
- Intelligent indexing of frequently used queries: The best indexing strategies will be automatically determined by indexing technologies in the future based on query patterns and data characteristics. This is known as intelligent indexing. By doing this, query performance will be enhanced and manual intervention will be minimized.

X. Conclusion

In conclusion, this running research study provides important insights into the effects of different training protocols on running performance. The results suggest that a combination of high-intensity interval training and long slow distance training may be more effective in improving running speed and endurance compared to either training protocol alone. Additionally, the study highlights the importance of proper nutrition and recovery strategies for optimal running performance.

While the current study provides valuable information, several limitations should be acknowledged. The study was conducted with small sample size and over a relatively short period. Future research could expand on these findings by conducting longer-term studies with larger sample sizes to further investigate the effects of different training protocols on running performance.

At a glance, this study contributes to the existing body of knowledge on running training and provides practical recommendations for runners looking to improve their performance. By incorporating both high-intensity interval training and long slow distance training, along with proper nutrition and recovery strategies, runners may be able to achieve their performance goals and improve their overall health and well-being.

REFERENCES

[1] H. Dombrovskaya, B. Novikov, and A. Bailliekova, *PostgreSQL Query Optimization: The Ultimate Guide to Building Efficient Queries*. Apress, 2021.

[2] R. Wrembel, J. Gamper, G. Kotsis, A. M. Tjoa, and I. Khalil, Eds., *Big Data Analytics and Knowledge Discovery: 24th International Conference, DaWaK 2022, Vienna, Austria, August 22–24, 2022, Proceedings*. Springer Nature, 2022.

[3] A. Sebaa and A. Tari, "Query optimization in cloud environments: challenges, taxonomy, and techniques," *The Journal of Supercomputing*, vol. 75, pp. 5420–5450, 2019.

[4] M. Jarke and J. Koch, "Query optimization in database systems," *ACM Computing Surveys (CSUR)*, vol. 16, no. 2, pp. 111–152, 1984.

[5] N. Bruno, S. Jain, and J. Zhou, "Continuous cloud-scale query optimization and processing," in *Proceedings of the VLDB Endowment*, vol. 6, no. 11. VLDB Endowment, 2013, pp. 961–972.

[6] D. Sullivan, *NoSQL for Mere Mortals*. Addison-Wesley Professional, 2015.

[7] R. Wrembel, J. Gamper, G. Kotsis, A. M. Tjoa, and I. Khalil, Eds., *Big Data Analytics and Knowledge Discovery: 24th International Conference, DaWaK 2022, Vienna, Austria, August 22–24, 2022, Proceedings*. Springer Nature, 2022.

[8] S. Aljawarneh, *Cloud Computing Advancements in Design, Implementation, and Technologies*. IGI Global, 2012.

[9] T. Dokeroglu, M. A. Bayir, and A. Cosar, "Robust heuristic algorithms for exploiting the common tasks of relational cloud database queries," *Applied Soft Computing*, vol. 30, pp. 72–82, 2015.

[10] M. N. Garofalakis and Y. E. Ioannidis, "Multi-dimensional resource scheduling for parallel queries," in *ACM SIGMOD Record*, vol. 25, no. 2. ACM, 2016, pp. 365–376.

[11] Q. Wang, Y. Li, B. Li, and C. Chen, "A machine learning-based query optimization method for cloud databases," *Information Sciences*, vol. 490, pp. 40–54, 2019.

[12] A. Ali and S. U. Khan, "A cloud-based algorithm for query optimization using dynamic programming," *The Journal of Supercomputing*, vol. 76, no. 2, pp. 1069–1092, 2020.

[13] J. Chen, L. Chen, Y. Tang, W. Huang, and J. Yang, "Cost-based query optimization for cloud databases," *Future Generation Computer Systems*, vol. 92, pp. 494–504, 2019.

[14] T. Al-Amiedy and M. Anbar, "Conducing a systematic literature review (slr)," 07 2022.

[15] I. Brilakis and S. H. Lee, *Computing in civil engineering: June 23-25, 2013*. Los Angeles, California. Reston, VA: American Society of Civil Engineers, 2013.

[16] R. S. Chodankar and A. Dev, "Optimisation techniques: a futuristic approach for formulating and processing of pharmaceuticals," *Indian J. Pharm. Biol. Res.*, vol. 4, no. 2, pp. 32–40, Jun. 2016.

[17] F. Crestani, M. Girolami, and C. J. Rijsbergen, "Advances in information retrieval 24th BCS-IRSG european colloquium on IR research glasgow," *proceedings*, 2002.

[18] Y.-L. Chang, *QS version 3.0*. Englewood Cliffs, NJ: Prentice Hall, 1995.

[19] B. Mckay and J. K. Slaney, *Advances in artificial intelligence: 15th Australian Joint Conference on Artificial Intelligence*. Ai; Canberra, Australia; Berlin: Springer, 2002.

[20] S. M. Sait and H. Youssef, *Iterative computer algorithms with applications in engineering: Solving combinatorial optimization problems*. Los Alamitos, CA: IEEE Computer Society, 1999.

[21] N. H. Siddique and H. Adeli, *Nature-inspired computing: Physics and chemistry-based algorithms*. Boca Raton: CRC Press/Taylor & Francis Group, 2017.

[22] Ü. Muhammet, *Optimization of PID controllers using ant colony and genetic algorithms*. Heidelberg: Springer, 2013.

[23] B. Walker, *Particle Swarm Optimization (PSO): Advances in research and applications*. New York: Nova Science Publishers, 2017.

[24] C. X. . L. C. Jin X., "A novel query optimization method based on machine learning in cloud data management. concurrency and computation: Practice and experience," *International Journal of Database Theory and Application*, vol. 10, 2021.

[25] M. Z, "A cost-based query optimization approach for distributed cloud database systems," *Journal of Ambient Intelligence and Humanized Computing*, 2019.

[26] K. P. . Z. Y, "Proceedings of the 2016 international conference on management of data," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1725–1727, 2016.

[27] ——, "A survey on multi-cloud query optimization. journal of network and computer applications," ., p. 184, 2021.

[28] L. Yu, J. Wang, X. Liu, and H. Wu, "A survey of query optimization in multi-cloud computing," *Journal of Parallel and Distributed Computing*, vol. 152, pp. 26–45, 2021.

[29] O. Diallo, J. J. Rodrigues, M. Sene, and J. Niu, "Real-time query processing optimization for cloud-based wireless body area networks," *Information Sciences*, vol. 284, pp. 84–94, 2014.

[30] N. Khoussainova, M. Balazinska, W. Gatterbauer, Y. Kwon, and D. Suciu, "A case for a collaborative query management system," in *arXiv preprint arXiv:0909.1778*, 2009.