International Journal on Advanced Science Engineering Information Technology

Vol.14 (2024) No. 4
ISSN: 2088-5334

Upscaling Current Data Caching and Prefetching Strategies for Online Databases in Nigeria

Olatunji Austine Kehinde [a,b], Zahidah Zulkifli [c], Ely Salwana [d], Nur Leyni Nilam Putri Junurham [a], Murni Mahmud [c], Salem Bahmaid [e]

[a] Department of Library and Information Science, Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, Kuala Lumpur, Malaysia
[b] University Library, Bamidele Olumilua University of Education, Science and Technology, Ikere-Ekiti, Ekiti State, Nigeria
[c] Department of Information Systems, Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, Kuala Lumpur, Malaysia
[d] Institute of Visual Informatics, Universiti Kebangsaan Malaysia, Bangi, Malaysia
[e] Faculty of Computer Studies, Arab Open University, Riyadh, Saudi Arabia
*Corresponding author: *kennyolatunji77@gmail.com*

***Abstract*— This study investigated upscaling current data caching and prefetching strategies for online databases in Nigeria, with a focus on practical implications. The research design adopted for this study was the descriptive survey. The population comprised of all undergraduate's library students in public tertiary institutions in Ekiti State. A simple random sampling technique was adopted to select 200 library students from Ekiti State University in the study area. The instrument used for data collection was a structured 4 Likert type questionnaire. The questionnaire was distributed to the respondents to find out the effectiveness of caching and prefetching techniques on online databases. The instrument was both face and content validated by two experts from the department of Library studies in Ekiti State University, Ado-Ekiti State. The reliability of the instrument was ensured using Pearson Product Moment Correlation formula which yielded a coefficient of 0.99. The data collected were analyzed using descriptive statistics such as mean and standard deviation. The result showed that the current caching and prefetching techniques employed in online databases are highly effective; the different access patterns have effect on the effectiveness of caching and prefetching techniques in online databases and there are impacts of cache coherence mechanisms on the efficiency of caching and prefetching techniques in online databases. It was therefore recommended that the inclusion of caching and prefetching in curriculum is important across all educational level in Nigeria, with a clear emphasis on the practical implications. In addition, caching and perfecting have come under fire for focusing mostly on computer science.**

***Keywords*— Caching; prefetching; online; database; performance.**

Manuscript received 7 Aug. 2023; revised 30 Dec. 2023; accepted 12 Jun. 2024. Date of publication 31 Aug. 2024.

I. INTRODUCTION

The advent of technology and the use of online databases have become increasingly important in academic research across the globe. This made it essential for Universities in Nigeria and the southwest, in particular, to provide access to online databases to enhance teaching, learning, and research activities. Online databases have gained popularity among librarians and users due to their speed, flexibility, wide range, and currency Fang et al. [1]. These databases are accessible worldwide and crucial for tertiary students, providing access to information resources and engaging in learning processes without physical constraints. Digital libraries have transformed the Internet into an essential platform for academic research. Therefore, with the increasing availability of online databases, students can conveniently access a wide range of information remotely, enabling them to solve academic tasks, conduct research, and enhance their knowledge [2], [3].

According to Blackman [4], online database systems optimize the web-based environment's utility by integrating the most authoritative content into the workflow by surfacing the most crucial relevant data. The material of the online database is regularly updated to reflect current practices in response to the constantly evolving standards of sustainability and corporate social responsibility. Online databases also help busy professionals write concise reports by citing thousands of sources and providing access to millions of documents. Professionals can now quickly locate high-quality articles for publications they otherwise might not have had access to or the time to check.

Online databases also help busy professionals write concise reports by citing thousands of sources and providing access to millions of documents. Professionals can now quickly locate high-quality articles for publications they otherwise might not have had access to or the time to check. User-friendly platforms deliver the articles and other publications [5], [6]. Online databases are a trustworthy source of information technology and material. They have a proven track record of offering hosted solutions, timely service delivery, and excellent partner communications Kinner and Rigda [7].

As a result, University libraries have made significant investments in online database subscriptions, aiming to provide students unrestricted access to these invaluable resources, regardless of their geographical location [8]. However, the exponential growth of data and the escalating demand for real-time access to information have presented significant challenges for online databases. One of the critical challenges is ensuring efficient data access performance, as it directly influences the overall efficiency and user experience of online applications. To effectively optimize data access performance in online databases, it is necessary to consider several factors, including data popularity, access patterns, and cache coherence by Akinola et al. [2].

Data popularity, which refers to the frequency of data items being accessed, is significant in determining which data should be cached or prefetched [9]. Though distributed systems offer a valuable approach to address the challenges posed by large-scale applications that generate massive volumes of data, if data reliability, availability, accessibility, and fault tolerance together with reduced data access time and network traffic will be achieved, the replication of data across diverse locations within the system should be overlooked [10]. Therefore, by prioritizing popular data, the system can better anticipate and meet user demands, improving overall performance.

Access patterns also play a crucial role in understanding how data is accessed in online databases. Analyzing the patterns of user queries and transactions can help identify recurring sequences and dependencies in data access. This knowledge can be leveraged to design efficient caching and prefetching strategies that align with these access patterns, further enhancing the database's performance [11].

Another crucial consideration is cache coherence, which ensures that multiple cache copies of the same data item remain consistent. Maintaining cache coherence becomes vital in online databases, where multiple users may concurrently access and modify shared data. Conflicts and inconsistencies can be minimized by implementing appropriate mechanisms, such as cache invalidation or data synchronization protocols, leading to a more reliable and efficient data access environment [12].

Prefetching technology is divided into heuristic prefetching and reasoning prefetching. Heuristic prefetching technology is inspiring by analyzing the history of file access, in which data access patterns are found, and a file access sequence model is established as a basis for data prefetching. Reasoning prefetching technology is inferencing based on relevant information and rules provided by the upper application. It detects the data it needs and reads it into the buffer. The upper application is transparent to the cloud storage system, so the heuristic prefetching technology has been widely studied and used [10], [13]. Heuristic prefetching algorithms are roughly divided into methods based on timing analysis and data mining-based methods. The timing analysis method is the most widely used based on the Markov chain model, in which the basic idea is to use the file as a state, the file access process is abstracted as a probability matrix, and the subsequent file access is predicted in turn.

Block-level prefetching is relatively simple and easy to implement. It does not require sophisticated prediction models and is often used in underlying storage systems. File-level prefetching requires strong dependencies between files access, or it will exacerbate I/O stress and pollute caches PhoenixNAP [14]. The upper layer of the cloud storage system is often a public user; user access often has behavioral regularity, and Kroeger et al.'s research proves that an access rule between files exists. Access patterns can be found by analyzing and mining the access sequence. For small files, the random read on the disk can be transferred into a sequential read through the file relevance prefetching. It can significantly improve the efficiency of file reading. The disadvantages of these methods lie in the inefficiency of updates and the mismatch between the granularity of synchronization and the data transfer [15].

According to Zamin [13], caches manage memory latency by storing frequently accessed data in faster, low-latency memory. They use locality to reduce delays in accessing main memory. Caches consist of a cache controller, data arrays, and tag arrays, with high-performance systems having multiple caches and low-power embedded systems focusing on instruction caches to minimize area and power consumption. This ensures high availability and parallel processing, reducing the load on individual systems and ultimately leading to improved performance [13]. Caches improve data access speed and reduce transportation costs by storing data on fast technologies near users. Strategies prioritize relevant content within limited cache space for maximum benefits [14].

Modern processor design increasingly depends on the memory system as a bottleneck because the on-chip core count expands faster than the memory bandwidth. Researchers have been obliged to employ aggressive strategies to cover up memory delays, such as using large-size caches, multithreading, and prefetching. Prefetching, among others, offers unique benefits. Large caches cost more energy and take up valuable chip space that could be used for more cores [15], [16].

The caching and prefetching strategies for online databases are to discover the potential rules from many access records and generate a pattern library, which can be used to predict future file access. For cache strategy, the throughput of sequential access is much higher than random access. Suppose the files relevant to the access can be centralized in one file (cache). In that case, disk pressure can be effectively alleviated by prefetching data since random accesses are changed to sequential accesses [17], [18].

It has been shown that there are a variety of barriers that prevent the effective utilization of Internet resources. In higher education institutions, staff and students have been having many issues using them. Even though numerous studies have confirmed this, studies have also shown that people are still willing to use Internet resources. Numerous studies have shown that there are difficulties that users of online databases must overcome. A few studies by [19]–[22] identified these difficulties as lacking training, slow Internet speed, and poor connectivity. According to Yang [23] and Rui [24], which examined the difficulties associated with accessing online databases, faculty members frequently used these resources but experienced concerns with sluggish download speeds, restricted access to back issues, and inability to access the resources from their homes. Thus, efficient prefetching techniques are crucial for modern computer architectures, ensuring improved performance in high-performance and power-constrained systems through high popularity, access patterns, and data coherence [25]–[28].

II. Materials and Method

This study adopted a descriptive survey design. A descriptive survey seeks to find out specific facts concerning an existing phenomenon. Survey design can be described as an outline, a general arrangement, or a plan from which something is observed and may be made. According to Coe [19], a research survey design is a plan or blueprint that specifies how data relating to a given problem should be collected and analyzed. In this regard, this method of inquiry has become a popular research methodology since the 1990s, and it was widely used in educational research. within that field, there are several well-documented standards to support its use.

The study population comprised all undergraduate library users in public tertiary institutions in Ekiti State. A purposive random sampling technique was adopted to select 200 undergraduates from three public tertiary institutions in the study area. The instrument used for data collection was a structured 4 Likert-type questionnaire. The questionnaire was titled "Upscaling Current Caching and Prefetching Strategies for Online Databases in Nigeria" (UCCPSODN) and divided into two sections. Section A contains items designed to obtain personal data about respondents, while section B was designed to answer research questions for the study. The respondents were guided to respond to each item thus: SA-Strongly Agree (4); A - Agree (3); D - Disagree (2) and SD - Strongly Disagree (1). Also, HE - Highly Effective (4), ME - Moderately Effective (3), LE – Low Effect (2) and NE - Not Effective (1).

The questionnaire provided five (5) different items on questions 1 to 4 that require the respondents to tick one out of four options on how effective the current caching and prefetching techniques employed in online databases, how different access patterns affect the effectiveness of caching and prefetching techniques in online databases; the impacts of cache coherence mechanisms on the efficiency of caching and prefetching techniques in online databases and how novel caching and prefetching techniques be designed to optimize data access performance in online databases, considering factors such as data popularity, access patterns, and cache coherence.

The questionnaire was distributed to the respondents to determine the effectiveness of caching and prefetching techniques on an online database. Two experts from the Department of Library Studies in Ekiti State University, Ado-Ekiti, Ekiti State, Nigeria, validated the instrument's face and content. The instrument's reliability was ensured using the Pearson Product Moment Correlation formula, which yielded a coefficient of 0.99.

The data collected were analyzed using descriptive statistics such as mean and standard deviation. The mean decision value was calculated thus: ($\frac{4+3+2+1}{4} = \frac{10}{4} = 2.5$). The mean values greater than or equal to 2.50 indicated "Agreed" by the respondents to the statement in question, otherwise "Disagree."

III. Results and Discussion

Four questions are being asked during the survey. The details of the questions and results are as follows:

A. *Question 1: How Effective are Online Databases' Current Caching and Prefetching Techniques?*

TABLE I
Effectiveness of Caching and Prefetching Techniques on Online Database

S/N	Items	Mean	Standard Deviation	Decision
1.	Caching and prefetching techniques are effective as they increase performance by decreasing access time for obtaining data	3.05	1.19	Effective
2.	Caching and prefetching techniques are effective as they help in retaining frequently accessed data within memory	3.03	0.89	Effective
3.	Caching and prefetching techniques are effective as they help to cut down the number of times that a hard drive must be accessed or a network connection utilized.	3.43	0.80	Effective
4.	Caching and prefetching techniques are effective as they help in storing data retrieved from queries that may be needed again	3.32	0.78	Effective
5.	Data is retained from across multiple HTTP request and can be reused without incurring additional access time	3.27	0.89	Effective
Grand Mean		**3.25**	**0.87**	**Effective**

Note: Mean greater than 2.50 'Effective' otherwise 'Not Effective'

B. Question 2: How Do Different Access Patterns Affect the Effectiveness of Caching and Prefetching Techniques in Online Databases?

TABLE III
EFFECT OF DIFFERENT ACCESS PATTERNS ON EFFECTIVENESS OF CACHING AND PREFETCHING IN ONLINE DATABASE

S/N	Items	Mean	Standard Deviation	Decision
1.	Access pattern improves object layout and increase the effectiveness of caching and prefetching in online database	3.22	0.82	Agreed
2.	Access pattern reduces the number of cache misses by prefetching future memory references based on the similarity of access pattern among memory page	3.21	0.83	Agreed
3.	Access pattern keeps track of previous accesses to line in hot pages	3.27	0.83	Agreed
4.	Access pattern increases the amount of memory level parallelism	2.97	0.92	Agreed
5.	Access pattern increases availability by storing data in a faster and more accessible location	3.33	0.97	Agreed
Grand Mean		**3.09**	**0.93**	**Agreed**

Note: Mean greater than 2.50 'Agreed' otherwise 'Disagreed'

C. Question 3: What are the Impacts of Cache Coherence Mechanisms on the Efficiency of Caching and Prefetching Techniques in Online Databases?

TABLE IIIII
IMPACT OF CACHE COHERENCE MECHANISM ON THE EFFICIENCY OF CACHING AND PREFETCHING TECHNIQUES IN ONLINE DATABASE

S/N	Items	Mean	Standard Deviation	Decision
1.	The cache coherence mechanism tracks where each copy of each shared data block is located and stores it in a directory.	3.84	0.99	Agreed
2.	Cache coherence is a result of each CPU having its own private data cache, a similar duplicate of the shared memory location which helps to present simultaneously in multiple caches	3.88	1.12	Agreed
3.	The cache coherence mechanism enables recently used local variables to enter the proper cache and remain there through numerous reads and writes while using the protocol to maintain consistency of shared variables that may be present in multiple caches concurrently.	3.31	0.86	Agreed
4.	Cache coherence mechanism enables all processors to update to the new value updated by the new processor	3.21	0.82	Agreed
5.	Cache coherence mechanism guarantee that any new value is broadcast to all processors	3.17	0.98	Agreed
Grand Mean		**3.48**	**0.95**	**Agreed**

Note: Mean greater than or equal to 2.50 'Agreed' otherwise 'Disagreed'

D. Question 4: How Can Novel Caching and Prefetching Techniques be Designed to Optimize Data Access Performance in Online Databases, Considering Factors such as Data Popularity, Access Patterns, and Cache Coherence?

TABLE IVV
DESIGN OF NOVEL CACHING AND PREFETCHING TECHNIQUES TO OPTIMIZE DATA ACCESS PERFORMANCE IN ONLINE DATABASE

S/N	Items	Mean	Standard Deviation	Decision
1	Propose a cost-based caching model where different costs will be incurred depending on whether a missed data item is prefetched or fetched	3.35	0.86	Agreed
2	Formulate or reformulate the optimal caching and prefetching problem as a mini-cost flow problem	3.44	0.80	Agreed
3	Analytically characterized the optimal policy by providing sufficient conditions under which prefetching the missed data is the optimal choice	3.99	0.77	Agreed
4	Propose a lightweight "look-ahead" approximation policy based on the insights revealed by the characteristics of the optimal policy	3.04	1.16	Agreed
5	Conduct extensive experiments using CND traces and synthetic data requests that are generated from both heavy and light-tailed popularity distribution	3.94	0.78	Agreed
Grand Mean		**3.64**	**0.98**	**Agreed**

Note: Mean greater than or equal to 2.50 'Agreed' otherwise 'Disagreed'

E. Discussion

The analysis presented in Table 1 revealed how effective the current caching and prefetching techniques employed in online databases are. The mean values in the table range from

3.03 to 3.43, and they are more significant than 2.50, indicating that most respondents agreed that the current caching and prefetching techniques employed in online databases are highly effective. This implies that caching and prefetching techniques increase performance by decreasing access time for obtaining data (3.05). Caching and prefetching techniques help retain frequently accessed data within memory (3.03). It helps reduce the number of times a hard drive must be accessed or a network connection utilized (3.43). The techniques help store data retrieved from queries that may be needed again (3.32). Data is retained across multiple HTTP requests and can be reused without incurring additional access time (3.27). The table's grand mean (3.25) is more significant than 2.50, confirming that the current caching and prefetching techniques employed in online databases are highly effective. This finding is like the finding by [22] that caching and prefetching are two effective solutions to lessen Web service bottlenecks, reduce traffic over the internet, and improve the scalability of the Web system.

The analysis presented in Table 2 revealed how different access patterns affect the effectiveness of caching and prefetching techniques in online databases. The mean values in the table range from 2.97 to 3.33, and they are more significant than 2.50, indicating that most of the respondents agreed with the statement in items 1–5 of Table 2. This implies that the access pattern improves object layout and increases the effectiveness of caching and prefetching in online databases (3.22). Access patterns reduce the number of cache misses by prefetching future memory references based on the similarity of access patterns among memory pages (3.21). Access patterns keep track of previous accesses to a line in hot pages (3.27). It increases memory level parallelism (2.97) and increases availability by storing data in a faster and more accessible location (3.33). The table's grand mean (3.09) is more significant than 2.50, implying that the different access patterns affect the effectiveness of caching and prefetching techniques in online databases. This result correlates to [22] that access patterns affect the effectiveness of caching and prefetching techniques. Web proxy caching exploits the temporal locality and web prefetching utilizes the spatial locality of the Web objects; web proxy caching and prefetching can complement each other. Thus, combining caching and prefetching helps improve the hit ratio and reduce the user-perceived latency.

The analysis presented in Table 3 revealed the impacts of cache coherence mechanisms on the efficiency of caching and prefetching techniques in online databases. The mean values in the table range from 3.17 to 3.88, and they are more significant than 2.50, indicating that most of the respondents agreed with the statement in items 1–5 of Table 3. This implies that the cache coherence mechanism keeps track of where each copy of each shared data block is located and stores it in a directory (3.84). Cache coherence results from each CPU having its own private data cache. A similar duplicate of the shared memory location helps present simultaneously in multiple caches (3.88). The cache coherence mechanism enables recently used local variables to enter the proper cache and remain there through numerous reads and writes while using the protocol to maintain the consistency of shared variables that may be present in multiple caches concurrently (3.31). The cache coherence mechanism enables all processors to update to the new value updated by the new processor (3.21). It guarantees that any new value is broadcast to all processors (3.17). The grand mean value (3.48) in the table is also greater than 2.50, indicating that cache coherence mechanisms impact the efficiency of caching and prefetching techniques in online databases. This implies that the coherence of caching and prefetching strategies can improve latency by up to 60%, whereas the caching strategy alone improves the latency by up to 26%. KeyCDN [11] suggested an application of web log mining to obtain web-document access patterns and used these patterns to extend the well-known GDSF caching and prefetching policies.

The result of the analysis presented in Table 4 reveals how novel caching and prefetching techniques can be designed to optimize data access performance in online databases. The mean values in the table range from 3.04 to 3.99, and they are more significant than 2.50, indicating that most of the respondents agreed with the statement in items 1–5 of Table four. This implies that novel caching and prefetching techniques can be designed by proposing a cost-based caching model where different costs will be incurred depending on whether the missed data item is prefetched or fetched (3.35), formulating or reformulating the optimal caching and prefetching problem in as a mini-cost flow problem (3.44), analytically characterized the optimal policy by providing sufficient condition under which prefetching the missed data is the optimal choice (3.99), proposing a lightweight "look-ahead" approximation policy based on the insights revealed by the characteristics of the optimal policy (3.04) and conducting extensive experiment using CND traces and synthetic data requests that are generated from both heavy and light-tailed popularity distribution (3.94). The table's grand mean value (3.64) is more significant than 2.50, which justified the claims. This implies that combining web caching and web prefetching doubles the performance compared to single caching, which enhances data access performance in online databases. Even though there has been development in data caching and fetching, data popularity, access pattern, and cache coherence mechanisms on caching and prefetching in the curriculum are essential across all educational levels in Nigeria [29], [30]. In addition, caching and perfecting have come under fire because they focus primarily on computer science.

IV. Conclusion

Based on the study's findings, it was concluded that the current caching and prefetching techniques employed in online databases are highly effective. This is because they increase performance by decreasing access time for obtaining data, they help in retaining frequently accessed data within memory, they help to cut down the number of times that a hard drive must be accessed or a network connection utilized, they help in storing data retrieved from queries that may be needed again, and data is retained from across multiple HTTP request and can be reused without incurring additional access time.

The study also concluded that different access patterns affect the effectiveness of caching and prefetching techniques in online databases. Cache coherence mechanisms impact the efficiency of caching and prefetching techniques in online databases. Although data caching and fetching have developed, including them in the curriculum is essential across all educational levels in Nigeria. In addition, caching and perfecting have come under fire because they focus primarily on computer science.

The study contributes to the body of knowledge by introducing a novel relevancy-based replacement policy to replace the caches of data nodes and global cache nodes whenever space is needed to store incoming patterns. This will improve the effectiveness of data popularity, data access patterns, and cache coherence mechanisms in research.

References

[1] J. Fang, Y. Xu, H. Kong, and M. Cai, "A prefetch control strategy based on improved hill-climbing method in asymmetric multi-core architecture," *The Journal of Supercomputing*, vol. 79, no. 10, pp. 10570–10588, Feb. 2023, doi: 10.1007/s11227-023-05078-6.

[2] K. A. Eiriemiokhale, "Frequency of Use and Awareness of Electronic Databases by University Lecturers in South-West, Nigeria," *Library Philosophy & Practice*, 2020.

[3] J. A. Liasu, & S. F. Bakrin, "The Impact of Electronic Information Resources On The Reading Habits Of Library Users At Osun State University, Nigeria," *Library Philosophy & Practice*, 2022.

[4] N. Gupta, P. Zhang, R. Kannan, and V. Prasanna, "PaCKD: Pattern-Clustered Knowledge Distillation for Compressing Memory Access Prediction Models," *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, Sep. 2023, doi:10.1109/hpec58863.2023.10363610.

[5] M. K. Faeq and S. S. Omran, "Cache coherency controller for MESI protocol based on FPGA," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 2, p. 1043, Apr. 2021, doi: 10.11591/ijece.v11i2.pp1043-1052.

[6] O. A. Kehinde, Z. Zulkifli, E. S. M. Surin, N. L. N. P. Junurham, and M. Mahmud, "Elevating Database Performance: Current Caching and Prefetching Strategies for Online Databases in Nigeria," *Advances in Visual Informatics*, pp. 314–327, Oct. 2023, doi: 10.1007/978-981-99-7339-2_27.

[7] W. H. W. Ishak and N. F. Ismail, "Recommender System for Multiple Databases Based on Web Log Mining," *Annals of Emerging Technologies in Computing*, vol. 5, no. 5, pp. 187–193, Mar. 2021, doi: 10.33166/aetic.2021.05.023.

[8] J. K. Opele., "Inter-professional collaboration and knowledge management practices among clinical workforce in Federal Tertiary Hospitals in Nigeria," *Knowledge Management & E-Learning: An International Journal*, pp. 329–343, Sep. 2022, doi:10.34105/j.kmel.2022.14.018.

[9] Fortinet. What is Catching Data? Retrieved from https://www.fortinet.com/resources/cyberglossary/what-is-caching. 2023.

[10] A. Awad, R. Salem, H. Abdelkader, and M. A. Salam, "A Novel Intelligent Approach for Dynamic Data Replication in Cloud Environment," *IEEE Access*, vol. 9, pp. 40240–40254, 2021, doi:10.1109/access.2021.3064917.

[11] G. Hasslinger, M. Okhovatzadeh, K. Ntougias, F. Hasslinger, and O. Hohlfeld, "An overview of analysis methods and evaluation results for caching strategies," Computer Networks, vol. 228, p. 109583, Jun. 2023, doi: 10.1016/j.comnet.2023.109583.

[12] N. D. Keycdn, "What is Prefetching and Why Use it". Retrieved from https://www.keycdn.com/support/prefetching. 2023

[13] Y. Al Moaiad, Z. A. Bakar, A. M. E. Diab, N. Zamin, and Y. Yahya, "Cloud Service Provider Cost for Online University: Amazon Web Services versus Oracle Cloud Infrastructure," *Advances in Visual Informatics*, pp. 302–313, Oct. 2023, doi: 10.1007/978-981-99-7339-2_26.

[14] PhoenixNAP. What is Distributed Database? Retrieved from https://phoenixnap.com/kb/distributed-database. 2021.

[15] A. K. U. Deepmala, & P. K. Sharma, "Online Data Bases: A Review of Literature," *Ilkogretim Online*, vol.19, no. 4, pp. 7111-7123, 2020.

[16] S. D. Adhikari, E. R. Van Teijlingen, P. R. Regmi, P. Mahato, B. Simkhada, and P. P. Simkhada, "The Presentation of Academic Self in The Digital Age: The Role of Electronic Databases," *International Journal of Social Sciences and Management*, vol. 7, no. 1, pp. 38–41, Jan. 2020, doi: 10.3126/ijssm.v7i1.27405.

[17] A. K. Sandhu, "Big data with cloud computing: Discussions and challenges," *Big Data Mining and Analytics*, vol. 5, no. 1, pp. 32–40, Mar. 2022, doi: 10.26599/bdma.2021.9020016.

[18] H. Shukur, S. Zeebaree, R. Zebari, O. Ahmed, L. Haji, and D. Abdulqader, "Cache Coherence Protocols in Distributed Systems," *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, pp. 92–97, Jun. 2020, doi: 10.38094/jastt1329.

[19] H. Adeli and P. R. Vishnubhotla, "Parallel Processing and Parallel Machines," *Parallel Processing in Computational Mechanics*, pp. 1–20, Aug. 2020, doi: 10.1201/9781003066750-1.

[20] A. Thomasian, "Storage Systems: Organization, Performance, Coding, Reliability, and Their Data Processing," *ACM SIGKDD Explorations Newsletter*, vol. 25, no. 2, pp. 22–24, Mar. 2024, doi:10.1145/3655103.3655107.

[21] M. Forsell, J. Roivainen, V. Leppänen, and J. L. Träff, "Realizing multioperations and multiprefixes in Thick Control Flow processors," *Microprocessors and Microsystems*, vol. 98, p. 104807, Apr. 2023, doi: 10.1016/j.micpro.2023.104807.

[22] M. F. Plauth, "Improving the Accessibility of Heterogeneous System Resources for Application Developers using Programming Abstractions," *Doctoral dissertation, Universität Potsdam*, 2022.

[23] H.-J. Yang, J. Fang, M. Cai, and Z. Cai, "A Prefetch-Adaptive Intelligent Cache Replacement Policy Based on Machine Learning," *Journal of Computer Science and Technology*, vol. 38, no. 2, pp. 391–404, Mar. 2023, doi: 10.1007/s11390-022-1573-3.

[24] L. Rui, X. Huang, S. Song, Y. Kang, C. Wang, and J. Wang, "Time Series Representation for Visualization in Apache IoTDB," *Proceedings of the ACM on Management of Data*, vol. 2, no. 1, pp. 1–26, Mar. 2024, doi: 10.1145/3639290.

[25] G. Ayers, H. Litz, C. Kozyrakis, and P. Ranganathan, "Classifying Memory Access Patterns for Prefetching," *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, Mar. 2020, doi:10.1145/3373376.3378498.

[26] G. F. Oliveira et al., "DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks," *IEEE Access*, vol. 9, pp. 134457–134502, 2021, doi: 10.1109/access.2021.3110993.

[27] A. Naithani, J. Roelandts, S. Ainsworth, T. M. Jones, and L. Eeckhout, "Decoupled Vector Runahead," *56th Annual IEEE/ACM International Symposium on Microarchitecture*, Oct. 2023, doi:10.1145/3613424.3614255.

[28] R. Coe, M. Waring, L. V. Hedges, & L. D. Ashley, (Eds.), *Research methods and methodologies in education*. Sage, 2021.

[29] M. M. Adam, L. Zhao, K. Wang, and Z. Han, "Beyond 5G networks: Integration of communication, computing, caching, and control," *China Communications*, vol. 20, no. 7, pp. 137–174, Jul. 2023, doi:10.23919/jcc.2023.00.039.

[30] M. Mahbub and R. M. Shubair, "Contemporary advances in multi-access edge computing: A survey of fundamentals, architecture, technologies, deployment cases, security, challenges, and directions," *Journal of Network and Computer Applications*, vol. 219, p. 103726, Oct. 2023, doi: 10.1016/j.jnca.2023.103726.