

## A Survey on Distributed Database Systems in the Era of Big Data

Kazheen Ismael Hasan<sup>1</sup>, Hajar Maseeh Yasin<sup>2</sup>

<sup>1</sup>Akre University for Applied Sciences Technical College of Informatics-Akre Information Technology Department

<sup>2</sup>Akre University for Applied Sciences, Technical College of Informatics, Department of Information Technology, Duhok, KRG - Iraq

**ABSTRACT:** Distributed database systems have evolved to satisfy the needs of scalability, performance, and fault tolerance due to the current digital era's fast data expansion. The design concepts, benefits, and drawbacks of modern distributed database architectures—such as cloud-native systems, NoSQL, and NewSQL—are thoroughly examined in this study. With an emphasis on using artificial intelligence and machine learning approaches to improve query speed and anomaly detection, key difficulties such as data integrity, latency optimization, and safe multi-cloud integration are covered. Despite notable progress, important concerns about data privacy and synchronization in diverse settings remain, and moral leadership endures. To create more robust and accountable database systems, this study promotes a well-rounded strategy that addresses the ethical and social aspects of distributed data management and increases technical efficiency.

**KEYWORDS:** Distributed Database Systems, Big Data Management, NoSQL and NewSQL, Query Optimization.

### 1. INTRODUCTION

An unparalleled surge of data characterizes the digital era[1]. In 2020, the global data volume exceeded 59 zettabytes and is anticipated to attain 175 zettabytes by 2025[2], a magnitude that undermines the principles of conventional database management[3]. This influx has initiated a new epoch for database systems, essential for efficiently storing, managing[4], and deriving value from extensive datasets that support contemporary analytics and decision-making[5].

Traditional relational database systems are inadequately prepared to manage the volume and diversity of big data, frequently encountering difficulties in scaling and accommodating varied[6], rapidly evolving datasets[7]. In response, contemporary data architectures, ranging from distributed NoSQL databases to cloud-native database services, have emerged to offer the scalability and flexibility required for big data management [8]. Nevertheless, despite these advancements, contemporary research emphasizes enduring challenges[9], including data security, regulatory compliance, and latency[10] highlighting the continued necessity for significant progress in the domain[11].

### 2. BACKGROUND THEORY

A dramatic increase in data volume, velocity, and variety marks the Big Data era[12]. The global datasphere is anticipated to exceed 100 zettabytes by 2025 [13], presenting considerable challenges in storing, retrieving, and managing heterogeneous data [14]. Conventional relational database management systems (RDBMS) are progressively insufficient in managing the scale and variety of contemporary datasets [15]. Consequently, novel paradigms

such as NoSQL, NewSQL, data lakes, and distributed databases have arisen[16]. These advancements signify technical innovation and a fundamental transformation in the organization, accessibility, and data optimization for real-time decision-making [17].

NoSQL databases have become prominent due to their schema flexibility, horizontal scalability, and capability to handle unstructured and semi-structured data [18]. They have become indispensable in web-scale applications such as social media and IoT systems. They sacrifice ACID compliance for eventual consistency within the BASE model, potentially complicating transaction integrity [19]. For instance, consistency models in NoSQL databases such as Cassandra impose performance penalties under stringent conditions [20]. The conflict between scalability and consistency[21] illustrates the overarching difficulty of creating database systems that are both efficient and dependable in the context of Big Data demands [22].

NewSQL systems have emerged to bridge this gap, providing ACID transactions and SQL interfaces while ensuring the distributed scalability characteristic of NoSQL[23]. Systems like Google Spanner and Cockroach DB utilize advancements such as distributed consensus and sharding to ensure global consistency[24]. Performance evaluations demonstrate that NewSQL can attain throughput similar to NoSQL while maintaining transactional integrity, rendering them suitable for high-volume, mission-critical workloads [25]. Nonetheless, challenges such as distributed query optimization and conflict resolution persist, underscoring the ongoing necessity for research to scale relational models while maintaining integrity[26].

The expansion of diverse data types has catalyzed the emergence of data lakes and data warehouses[27]. Data lakes provide economical storage for unstructured data in its original state, facilitating large-scale analytics and machine learning applications[28]. However, initial implementations exposed deficiencies in governance and query performance. The data warehouse model resolves these challenges by incorporating ACID transactions and schema management atop data lakes [28]. These hybrid systems combine the adaptability of lakes with the organization of warehouses, enhancing analytical capabilities across various industries [29]. Furthermore, multi-model and enable organizations to manage varied data within or across systems [30] facilitating integrated queries across relational, graph, and document-oriented sources[31].

In addition to storage, performance optimization has emerged as a paramount issue. Contemporary methodologies[32], including in-memory processing, adaptive indexing, and parallel query execution, have become prevalent. AI and machine learning are progressively incorporated into database management systems to automate tuning, optimize queries, and potentially supplant conventional components such as indexes [33]. Machine-learned models have surpassed heuristic methods in query optimization and workload forecasting [34]. Autonomous databases now employ AI for self-configuration, adaptation to fluctuating workloads[35], and performance optimization with minimal human intervention. This transition signifies a wider industry trend towards autonomous systems that diminish operational complexity while enhancing responsiveness and reliability[36].

Cloud-native deployments and hybrid data architectures facilitate access to advanced database systems. Cloud platforms provide flexible, scalable services that facilitate global applications without substantial infrastructure expenditure. Simultaneously[37], various sectors- from healthcare to finance are adopting analogous data strategies, prioritizing integrated storage, real-time analytics, and adherence to data governance regulations [38]. These changes signify a fundamental transformation in data infrastructure, transitioning from monolithic systems to intelligent, integrated ecosystems[39].

### 3. LITERATURE REVIEW

This section delineates several prior studies relevant to this review article. Therefore, the current review incorporated findings from several earlier studies to interpret the key results and proposals, enhancing the background theory. Consequently, the previous studies will be delineated chronologically from the oldest to the most recent studies, as follows:

Topcu and Rmis (2020) [40] assessed the efficacy of the Riak KV NoSQL database within a distributed cluster setting utilizing the Basher-bench benchmarking instrument. They simulated diverse workloads that were read-only, update-

intensive, and mixed across varying data sizes and thread counts to evaluate throughput and latency. The findings indicated that read-only operations consistently attained superior throughput and reduced latency, whereas update operations diminished performance, particularly with larger datasets. Augmenting the thread count enhanced performance; however, scalability was constrained beyond a specific data volume. Their research offers insights into enhancing Riak KV for big data applications, especially for read-intensive scenarios.

Mosharraf and Adnan (2020)[41] introduced two optimization strategies for distributed Big Data systems utilizing Cuckoo Filters instead of Bloom Filters. The initial scheme improves lookup efficiency post-data deletion by facilitating key removal from filters, thus addressing a significant constraint in Bloom-based systems employing eventual consistency. The second scheme enhances remote query efficiency by implementing node filters that prevent superfluous network roundtrips when remote nodes do not possess the requested data. Both methodologies were executed and evaluated on Apache Cassandra, utilizing an authentic dataset, attaining a performance enhancement of up to 100% (2x) in the circumstances involving deleted or absent data. The experiments further validated that these enhancements impose negligible CPU and network overhead, rendering the approach feasible for real-world implementation.

Dioulasso and Tiendrebeogo (2020) [42] proposed a distributed Big Data storage system utilizing Distributed Hash Tables (DHT) to address the scalability and fault-tolerance limitations inherent in conventional MapReduce frameworks. Their model incorporates hyperbolic geometry and Poincaré disk-based addressing to facilitate decentralized, topology-independent routing and autonomous node organization. The system implements several controller nodes via virtual addresses, improving parallel processing and load distribution while preventing single points of failure. The proposed model demonstrates enhanced robustness and scalability compared to current DHT-based architectures like ChordReduce and P2P-MapReduce without depending on inflexible network topologies. Future endeavours involve the implementation of a hybrid DHT-MapReduce framework to assess performance in practical applications.

Aswal (2020) [43] examined the function of distributed database systems (DDBS) in managing extensive data across multiple sectors, including healthcare, e-commerce, and IoT. The research emphasizes the benefits of DDBS, such as scalability, real-time analytics, fault tolerance, and security. It also delineates significant challenges, including data consistency, replication, synchronization, and system complexity. A comparative analysis of systems such as Cassandra, DynamoDB, and Spanner highlights the trade-offs among performance, cost, and deployment flexibility.

Jowan et al. (2021) [44] examined the shift from conventional RDBMS to NoSQL databases prompted by the difficulties

associated with Big Data, unstructured data, and cloud-based applications. The paper classifies NoSQL systems into four categories: key-value, document, column-family, and graph databases, each engineered for flexibility and scalability. It elucidates how NoSQL utilizes the CAP theorem and BASE principles to guarantee high availability and partition tolerance in distributed systems. The study concludes that NoSQL databases are crucial for managing contemporary application requirements that involve real-time, large-scale, and diverse data.

Jinadu et al. (2021) [45] introduced a distributed database optimization model utilizing a Distributed Storage Pool (DSP) enhanced by virtualization and hybrid RAID technology to enhance service delivery in mobile and cloud Big Data applications. Their architecture utilizes semi-join operations, storage replication, and mobility transparency to improve data access efficiency and reduce latency in distributed transactions. Simulations utilizing M-TCP in WLAN environments exhibited substantial enhancements in throughput and response time relative to traditional TCP configurations. The research validates that DSP utilizing virtualization diminishes overhead and guarantees high availability and fault tolerance, rendering it appropriate for real-time, latency-sensitive cloud services.

Hongwei and Ligetu (2021)[46] examined distributed storage technologies to tackle the increasing difficulties of managing big data in cloud computing settings. They highlighted the constraints of centralized systems and advocated for the implementation of object-based distributed storage and virtualization to enhance scalability, efficiency, and data security. Their system accommodates diverse data types and provides adaptable, economical, and resilient storage appropriate for rapid data expansion. The research underscores the significance of adaptive storage architecture in fulfilling the performance requirements of big data applications.

Chang and Cui (2021)[47] introduced a distributed storage strategy to manage economic big data distinguished by spatial, temporal, and semantic diversity. A multilevel partitioning algorithm that integrates Geohash and Hilbert curves was introduced to enhance storage efficiency and facilitate cross-modal analysis. Their system was deployed on a NoSQL database (Cassandra) and evaluated with simulated workloads to verify resource efficiency and adherence to SLA requirements. The findings validated that their spatiotemporal-semantic-aware storage strategy markedly improves performance and adaptability for extensive economic data applications.

Thamar (2023)[48] investigated how distributed computing augments big data engineering by optimizing data ingestion, processing, and analysis within contemporary data architectures. The paper examines essential distributed models MapReduce, MPP, BSP, and in-memory computing, emphasizing their advantages in scalability, velocity, and real-time processing. It examines integrating distributed

principles using tools such as Apache Kafka, Spark, Delta Lake, and dbt to construct reliable and fault-tolerant pipelines. The research highlights that distributed systems are essential for handling large, rapidly evolving data in contemporary analytical settings.

Zhang et al. (2024)[49] introduced MultiLog, a multivariate log-based anomaly detection technique for distributed databases. The initial extensive dataset comprising 900 million log entries encompasses 11 categories of anomalies across various nodes. MultiLog extracts sequential, quantitative, and semantic features from distributed logs, employing an LSTM with self-attention and a cluster classifier for precise detection. The methodology attained an F1 score of up to 12%, which was superior to leading techniques and diminished false positives in multi-node settings.

Olusegun et al. (2024)[50] investigated the Secure Multi-Party Computation (SMPC) application in cloud-based big data analytics to safeguard data privacy during collaborative processing. The research analyzed protocols, including secret sharing, homomorphic encryption, and federated learning, demonstrating their capacity to facilitate secure computations while preserving the confidentiality of individual data inputs. SMPC was utilized in practical applications such as secure machine learning and privacy-preserving queries in multi-cloud environments. The authors determined that although scalability and communication overhead persist as challenges, SMPC is a viable method for secure data collaboration in sensitive areas.

Munawar et al. (2024)[51] systematically reviewed big data applications in smart real estate and disaster management, examining 139 studies published between 2010 and 2020. The document underscored the significance of the seven Vs volume, velocity, variety, value, veracity, variability, and visualization as essential facets of big data. It proposed cohesive frameworks demonstrating how big data can improve decision-making, service delivery, and emergency response utilizing IoT, AI, and social media analytics. The research identified persistent challenges, including data quality, system integration, and scalability in real-time, multi-source contexts.

Ibrahim (2024)[52] proposed a data synchronization method for heterogeneous distributed databases that amalgamate both row-oriented and column-oriented storage systems. The research presents a bi-directional synchronization model utilizing custom "Dsync" logs to monitor updates, deletions, and insertions across various databases independent of timestamps. It utilizes parallel processing, routing options, and centralized coordination to guarantee consistent, real-time data exchange among independent database environments. The methodology tackles critical issues, including format discrepancies, communication lags, and system autonomy, offering a scalable resolution for practical distributed applications.

Gadde (2024)[53] proposed an AI-driven framework to enhance transactional integrity in distributed database systems by incorporating machine learning and anomaly detection with conventional consensus protocols. The system comprises a predictive analytics engine and an integrity monitoring unit that forecasts transaction conflicts and identifies irregularities in real-time. Experimental findings in a simulated cloud environment demonstrated a 50% enhancement in transaction throughput, a 40% decrease in response time, and an 80% reduction in integrity violations. The research validates that AI can substantially improve efficacy and dependability in distributed database systems.

Aryan et al. (2024)[54] introduced a Rust-based framework for implementing Decentralized Autonomous Database Systems (DADBS) to tackle scalability and autonomy in distributed settings. The system incorporates a Proof of Work consensus mechanism, smart contracts, and a peer-to-peer network developed utilizing Rust's concurrency model and SQLite for data storage. Performance testing demonstrated a throughput of 3,000 transactions per second, high consistency, and nearly linear scalability up to 500 nodes. The findings underscore Rust's appropriateness for developing secure, efficient, decentralized databases with autonomous functionalities.

Zhu et al. (2025)[55] presented RAPO, an automated optimization instrument for Redis clusters employed in distributed metadata storage systems. The tool optimizes performance by balancing loads among primary nodes via greedy and random iterative algorithms, resulting in a load distribution improvement of up to 29.36%. It also employs read-write separation strategies, such as smooth weighted round-robin, to diminish metadata read latency by as much as 30.75% during periods of high concurrency. The research validates that RAPO markedly enhances the efficiency and scalability of Redis clusters in extensive distributed settings. Sato (2025)[56] examined the evolution of database architects' roles in distributed systems, transitioning from centralized schema design to overseeing scalability, partitioning, and consistency trade-offs. The study examines fundamental patterns, including sharding, schema versioning, replication models, and the ramifications of the CAP theorem. It underscores utilizing AI-assisted tools, polyglot persistence, and cloud-native technologies for managing contemporary distributed workloads. The paper asserts architects must reconcile technical complexity with strategic design in globally distributed infrastructures.

Kirino (2025) [57] analyzed the impact of distributed, cloud-native, and real-time data systems on the responsibilities of contemporary database architects. The research emphasizes essential design principles, including scalability, high availability, partitioning, and consistency models, while tackling operational challenges such as observability and fault tolerance. It delineates developing responsibilities,

encompassing the management of polyglot persistence, CI/CD integration, and ethical data governance within intricate infrastructures. The paper asserts that architects currently function as strategic system designers, harmonizing performance, compliance, and interdisciplinary collaboration in distributed data environments.

Evelyn (2025)[58] established an adaptive query optimization framework for heterogeneous big data environments, tackling schema diversity and system variability issues. The methodology incorporates metadata abstraction, machine learning-driven cost modelling, and federated execution planning to enhance performance across various platforms. Testing on systems such as PostgreSQL, MongoDB, Hive, and Elasticsearch demonstrated execution times up to 40% faster than conventional optimizers. The research illustrates that integrating learning algorithms with dynamic metadata improves query efficiency in intricate, distributed data systems.

Ailamaki (2025) [59] investigated parallel and distributed query execution as a fundamental approach for managing extensive big data workloads. The research examined systems such as Apache Spark, Hive, Presto, and Dask, emphasizing query planning, partitioning, fault tolerance, and load balancing. Experiments demonstrated that distributed execution markedly enhanced query performance; however, data skew and network bottlenecks constrained scalability beyond a specific threshold. The document underscores the necessity for adaptive, self-optimizing architectures to maintain efficient analytics in intricate distributed settings.

Adeleke (2025)[60] evaluated the efficacy of approximation algorithms for processing big data in distributed database systems. The research assessed sampling, sketching, and hybrid methodologies utilizing Apache Spark and HDFS to analyze performance, accuracy, and scalability. The findings indicated that sketch-based methodologies, such as Count-Min Sketch and HyperLogLog, yielded rapid, memory-efficient estimations with minimal communication overhead. The results validate that approximation methods can enhance query efficiency while preserving acceptable error thresholds in distributed settings.

Abiteboul et al. (2025) [61] examined essential optimization methodologies for distributed query execution in large-scale data systems, encompassing cost-based approaches, adaptive processing, indexing, and parallel execution. Their research demonstrated that these techniques enhance performance by minimizing latency, distributing load, and optimizing resource utilization. The authors investigated nascent methodologies such as machine learning optimization and quantum computing, which exhibit potential yet remain experimental. The study concludes that hybrid, intelligent optimization frameworks are crucial for effective and scalable distributed query execution.

**4. Table 1. Comparison among the reviewed works**

Synthesizing 22 empirical and conceptual studies spanning 2020 to 2025.

Author (Year)	Focus Area	Techniques	Key Findings	Advantages	Dataset	Performance	Limitations
Topcu & Rmis (2020)[40]	Riak KV benchmarking	Read/update workloads	Read fast, updates slow	Threads help scalability	Riak, Basho-bench	Latency, throughput	Update slow
Mosharraf & Adnan (2020)[41]	Filter optimization	Cuckoo, Cassandra	Query gain 100%	Low overhead	Cassandra	Query time	Scale issues
Dioulasso et al. (2020)[42]	DHT storage	Virtual DHT	Balanced access	No single point	Simulated DHT	Access speed	Needs real use
Aswal (2020)[43]	DDBS	SMPC, FL	Real-time support	Scalable	General DDBS	Scalability	Deployment gaps
Jowan et al. (2021)[44]	NoSQL shift	CAP, NoSQL types	Handles big data	Schema-free	NoSQL types	Flexibility, speed	NoSQL tradeoffs
Jinadu et al. (2021)[45]	DSP architecture	RAID, M-TCP	Faster response	Mobile-friendly	M-TCP, WLAN	Response time	Coordination load
Hongwei & Ligetu (2021)[46]	Cloud storage	Object store	Low-cost storage	Adaptable	Cloud, object store	Cost, latency	Integration limits
Chang & Cui (2021)[47]	Economic data	Geo+Hilbert	Efficient storage	Cross-modal	Cassandra, Hilbert	Storage use	Model tuning
Thamar (2023)[48]	Data pipelines	Modern tools	Real-time boost	Streamlined	Kafka, Spark	Real-time ops	Tool mix
Zhang et al. (2024)[49]	Anomaly detect	LSTM + MultiLog	F1 12%	Multi-node accurate	900M log entries	F1 score	Single-node limits
Olusegun et al. (2024)[50]	Secure analytics	SMPC, FL	Safe collaboration	Cloud-ready	Multi-cloud	Security	Overhead
Munawar et al. (2024)[51]	Smart apps	7Vs, IoT	Helps cities/disasters	Flexible	IoT, social media	Process time	Real-time limits
Ibrahim (2024)[52]	Data sync	Dsync logs	No timestamp needed	Scalable	Dsync logs	Sync speed	Metadata sync
Gadde (2024)[53]	AI in DB	ML + anomaly	Violations 80%	AI boosts integrity	Cloud sim.	Throughput	ML overhead
Aryan et al. (2024)[54]	DADBS	Rust + PoW	3000 TPS	Resilient	Rust, 500 nodes	TPS, consistency	Latency, PoW
Zhu et al. (2025)[55]	Redis optimize	RAPO, LBI	Latency 30%	Balanced reads	Redis cluster	Latency drop	Write scaling
Sato (2025)[56]	DB roles	Cloud/micro	Modern DB shift	Architect role	Architectural review	Design agility	Skill gaps
Kirino (2025)[57]	DB design	Cross-domain	Governance aware	Balanced	Design concepts	Observability	Tool balance
Evelyn (2025)[58]	Query opt.	ML + metadata	Query time 40%	Adaptive	PostgreSQL, Hive	Execution time	Source diversity
Ailamaki (2025)[59]	Query exec	Spark, Dask	Latency improved	Self-tuning	Spark, Dask	Latency, fault tol.	Cost model gaps
Adeleke (2025)[60]	Approx. algo	Sampling, sketching	Fast queries	Efficient	Spark, HDFS	Response, error	Coordination
Abiteboul et al. (2025)[61]	Query optimization	Cost, ML, quantum	Performance	Hybrid models	Simulated env.	Query time	Quantum/ML



## 5. DISCUSSIONS & COMPARISON

Table 1 provides a detailed summary of the previous publications. It presents the key indicators used for assessment and emphasizes the significant findings from these studies, highlighting the strengths and innovative concepts that emerged from the research.

### Synthesis of reviewed paper

The examined literature illustrates a dynamic and evolving domain characterized by recurring themes of performance enhancement, scalability, and system adaptability. A thorough analysis indicates that although basic distributed architectures are firmly established, research is increasingly focused on enhancing efficiency, integrity, and intelligent automation.

Performance optimization continues to be a primary emphasis.[40] illustrated that Riak KV performs exceptionally well in read-dominant workloads but falters in update-intensive scenarios, particularly with substantial datasets, highlighting the constraints of linear scalability. [45] and [55] proposed DSP and RAPO architectures that markedly enhance throughput and latency, highlighting performance optimization via virtualization and node load balancing. Incorporating AI and machine learning in query optimization signifies a pivotal transition towards adaptive and autonomous systems. [58] utilized machine learning-based cost models, resulting in a 40% enhancement in query time. [53]developed an AI-driven anomaly detection model that decreased transaction violations by 80%, demonstrating the efficacy of predictive analytics in improving transactional integrity. Scalability and fault tolerance are paramount. [61]identified the shortcomings of traditional MapReduce. They proposed a Distributed Hash Table (DHT)-based system utilizing Poincaré disk geometry, thereby improving decentralized coordination and mitigating single points of failure. [54]further developed a Rust-based decentralized database that can manage 3,000 transactions per second (TPS) and exhibits near-linear scalability up to 500 nodes. Security and data governance, while less commonly examined, are becoming vital issues.[50]examined Secure Multi-Party Computation (SMPC) in big data analytics, demonstrating that privacy-preserving protocols are viable,

though accompanied by communication overhead.[52]addressed synchronization in heterogeneous environments by introducing "Dsync logs," which obviate the need for timestamps in real-time, bi-directional data updates. The literature reveals an imbalance: Although system performance is extensively studied, topics like data ethics, secure collaboration, and governance receive significantly less empirical focus despite their growing importance in distributed, multi-cloud environments. This view indicates a necessity for more integrative strategies that harmonize technical efficiency with ethical and operational resilience [56],[57] In this manner, the trajectory of distributed database research demonstrates a comprehensive understanding of performance mechanics yet exposes significant deficiencies in aspects such as cross-platform interoperability, secure collaboration, and long-term governance. Future initiatives must progress beyond optimization to adopt resilient, ethical, and intelligent architectures that satisfy the requirements of modern data ecosystems.

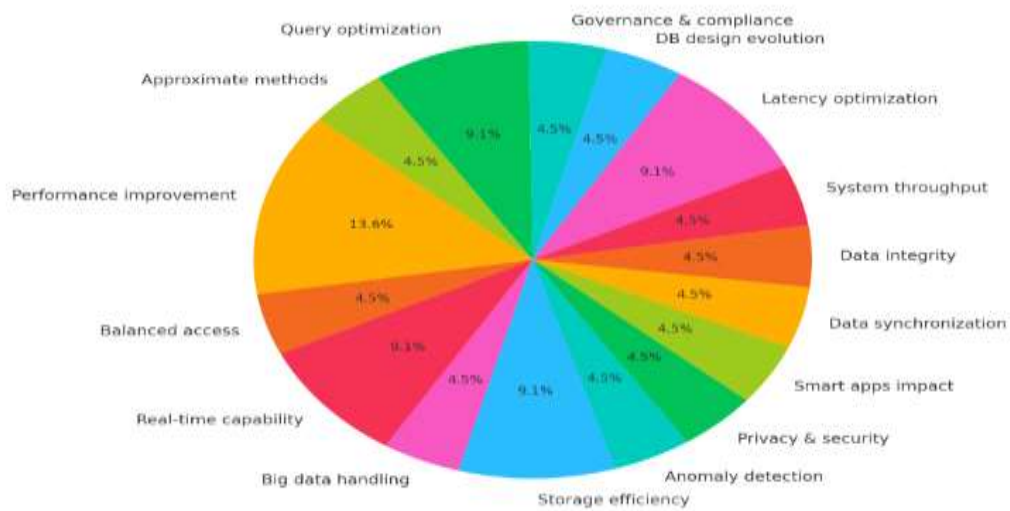
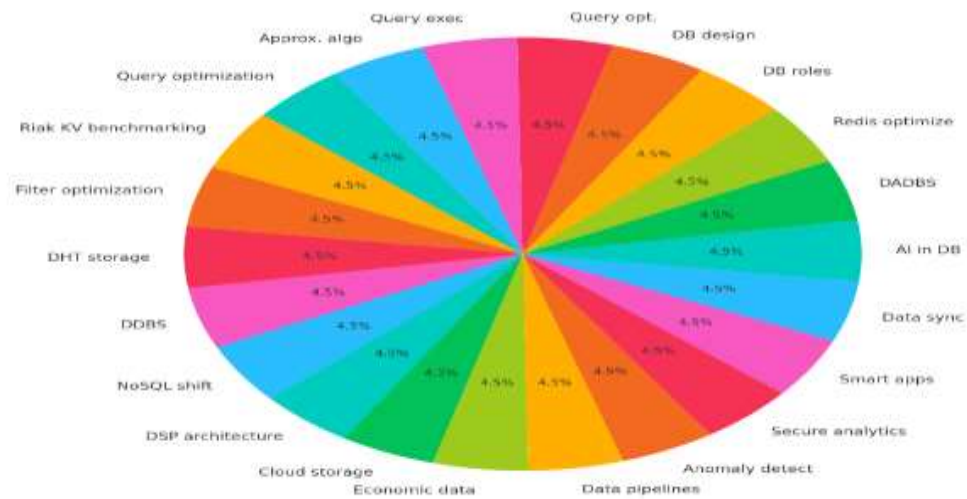
## 6. EXTRACTED STATISTICS

Figure (1) depicts a multifaceted research landscape in distributed databases and big data systems, emphasizing diverse focal points without a predominant theme. The persistent interest in query optimization, AI integration, and NoSQL significantly indicates a collective emphasis on improving performance and scalability. As demonstrated, the focus on machine learning-based optimization signifies a transition towards intelligent, self-adaptive data systems. Simultaneously, underrepresented domains such as secure analytics and data synchronization present significant privacy, consistency, and system interoperability issues, highlighting these subjects' complexity and evolving nature. The insufficient focus on intelligent applications and economic data storage indicates a disparity between research and sector-specific requirements. The distribution indicates a domain that harmonizes innovation with execution. Future research must integrate theoretical insights with practical limitations, ensuring that distributed systems are efficient, ethical, secure, and contextually aware.

**Figure 1 Statistical representation of Big Data papers (2020 – 2025) based on the focus area**

Figure (2) represents the thematic allocation of principal findings from the analyzed research on distributed databases and big data systems. Most research focuses on enhancing performance, indicating the academic community's emphasis on speed, throughput, and system responsiveness in progressively intricate data environments. The significant emphasis on query optimization and latency reduction reflects continuous endeavors to enhance data access efficiency and timeliness. Simultaneously, real-time functionality and storage optimization demonstrate the

demand for scalability and agility in dynamic environments such as IoT and cloud platforms. Infrequent yet equally significant are themes such as privacy and security, data synchronization, and governance, which evoke ethical and technical issues frequently neglected in performance-focused discussions. The chart highlights the necessity for a comprehensive research methodology that improves system performance while tackling integrity, coordination, and ethical data utilization in distributed architectures.



**Figure 2 Statistical representation of Big Data papers (2020 – 2025) based on the key findings**

Figure (3) illustrates a thematic analysis of the benefits identified in research on distributed databases and big data systems. Scalability is highlighted as the paramount advantage, signifying the urgent need to manage increasing data volumes and effectively dispersed workloads. The subsequent themes pertain to efficiency and performance optimization, illustrating continuous endeavors to refine system operations, minimize latency, and improving responsiveness. Flexibility underscores the significance of adaptable systems that accommodate various data types and

changing requirements. Infrequent but essential benefits, including security, accuracy, and governance, indicate an increasing recognition of ethical and operational intricacies. The distribution indicates a research landscape predominantly focused on technical performance objectives while becoming progressively mindful of adaptability and trust. In practical settings, future development must prioritize the equilibrium between system optimization and overarching issues such as user autonomy, data ethics, and cross-platform resilience.

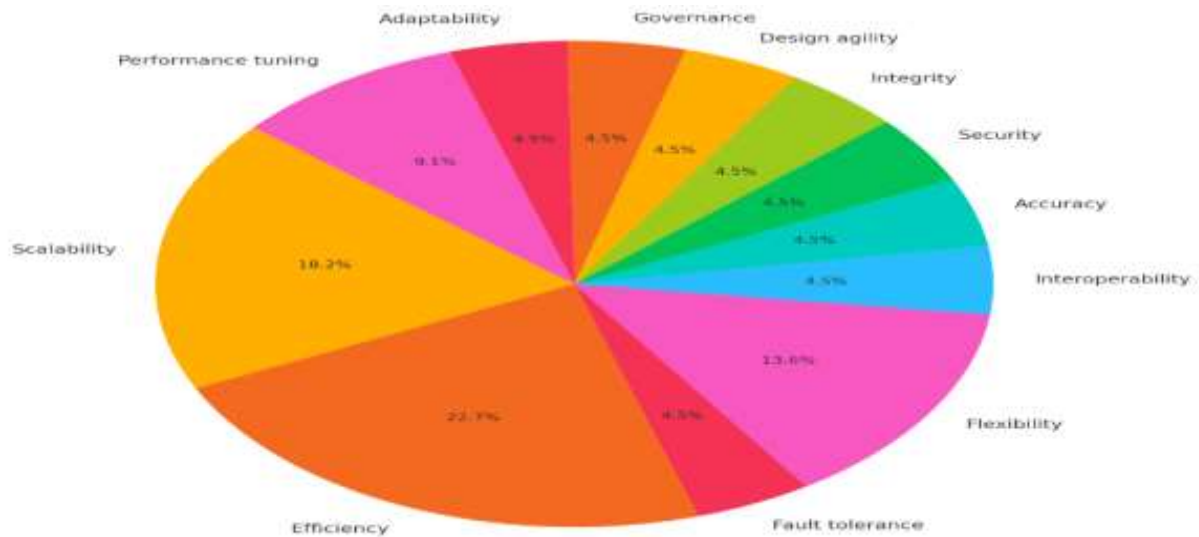


Figure 3 Statistical representation of Big Data papers (2020 – 2025) based on the Advantages

Figure (4) displays the varied performance metrics highlighted in the literature examined on distributed databases and big data systems. Latency and throughput are the most addressed metrics, highlighting the field's persistent focus on speed and responsiveness. Query time is significant, indicating the necessity for efficient data retrieval in real-time and extensive environments. Efficiency, real-time operations, and system throughput exemplify the diverse objectives of optimizing computational resources and user experience. Thus, underrepresented yet vital aspects such as

synchronization, security, and monitoring highlight systemic issues that, while frequently subordinate to performance, are crucial for enduring stability and trust. This distribution indicates a performance-oriented paradigm in contemporary research; however, it also prompts contemplation: as systems become increasingly interconnected and data-intensive, there is an escalating necessity to reconcile speed with reliability, transparency, and adaptability in distributed data architectures.

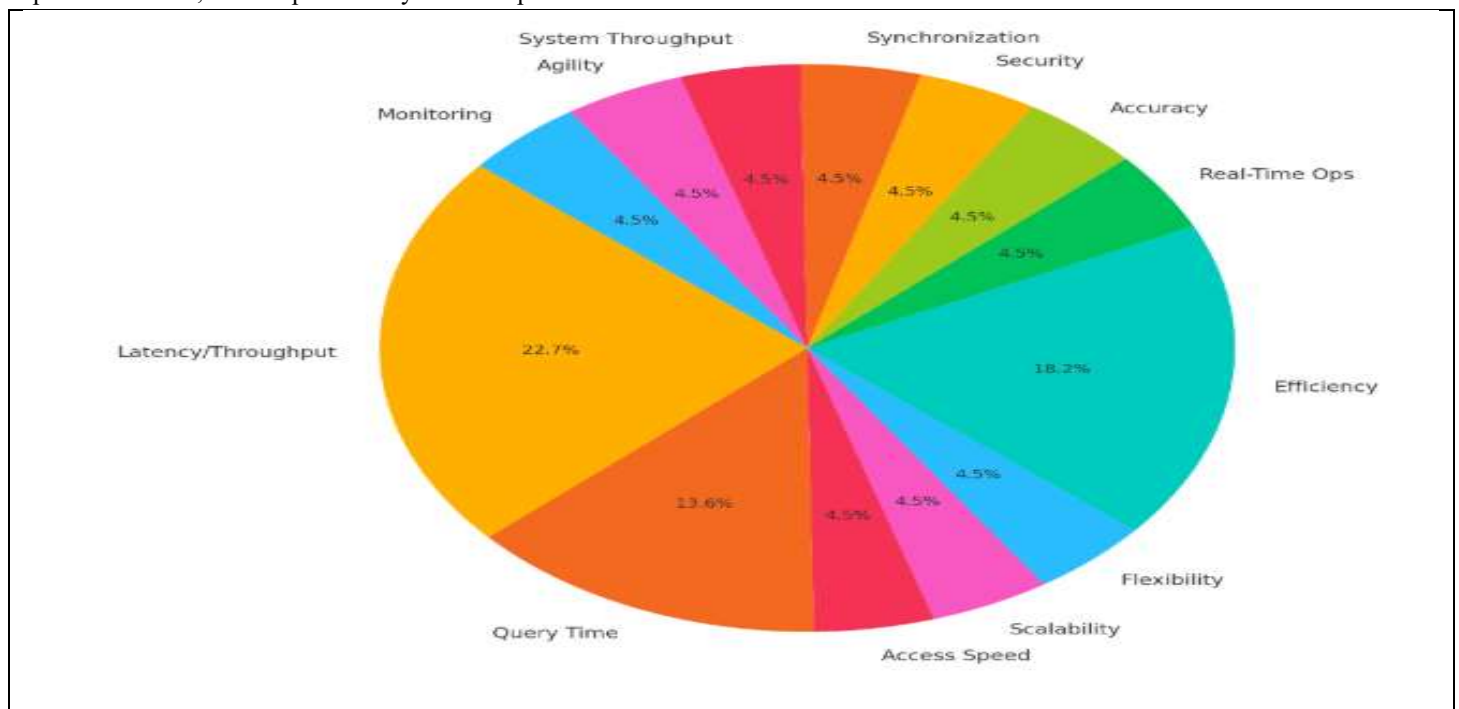


Figure 4 Statistical representation of Big Data papers (2020 – 2025) based on the Performance

## 7. RECOMMENDATIONS & FUTURE AREA

A fundamental recommendation is to incorporate ethics by design into their architecture to promote the responsible

development of distributed database systems to promote the responsible development of distributed database systems. Contemporary development practices frequently emphasize



technical performance while neglecting inherent ethical considerations, including fairness, transparency, and user autonomy. Future systems, especially those utilizing AI for autonomous decision-making, must incorporate ethical considerations from the initial phases of system modelling and design [57],[53]. Moreover, performance should not be sought in isolation. Distributed databases must adequately facilitate governance, auditability, and interoperability, especially within multi-tenant or federated frameworks. This view involves implementing synchronization mechanisms, data lineage tracking, and compliance-ready features as standard components [52],[50].

Furthermore, an increasing dependence on databases in essential services necessitates that resilience and decentralization be considered fundamental design principles. Decentralized Autonomous Databases (DADBS) and Distributed Hash Table (DHT) architectures present effective frameworks for fault tolerance and operational continuity in adverse conditions [61]&[55],[54]. Finally, creating multi-faceted evaluation frameworks that extend beyond conventional latency or throughput metrics is essential. These must encompass energy consumption metrics, explainability, user autonomy, and regulatory compliance—ensuring that distributed systems are efficient, sustainable, and socially responsible.

Ultimately, distributed databases' future must be engineered and ethically designed. As these systems increasingly support essential societal infrastructures, from healthcare and finance to public administration, the stakes have transcended mere technical considerations. Researchers and practitioners must adopt multi-dimensional design methodologies integrating performance, ethical foresight, regulatory compliance, and civic responsibility. This outlook involves transitioning from discrete technological enhancement to system thinking, where databases are more efficient and intelligent but equitable, secure, and transparent. This transition is not merely advantageous, it is essential. Future research should primarily focus on enhancing the scalability of privacy-preserving computation. With the increasing rigor of data sovereignty and cross-border regulations, it is imperative to adapt mechanisms such as Secure Multi-Party Computation (SMPC) and Federated Learning for large-scale, low-latency environments, particularly in multi-cloud deployments [50]. A promising avenue is the advancement of explainable AI for database optimization. As AI-driven systems progressively execute autonomous decisions regarding query planning, anomaly detection, and integrity verification, researchers must guarantee that these decisions are interpretable and auditable for technical users and governance entities [58],[53]. Hence, cross-domain applications of distributed databases are still inadequately investigated. The practical application in healthcare, public administration, and smart city infrastructures would evaluate the proposed models' resilience and underscore the conflicts between technical

scalability and ethical accountability. Research must examine these intersections to guarantee that distributed systems are adaptable to socially sensitive contexts. The environmental impact of distributed architectures must be prioritized. The energy intensity of replication protocols, consensus mechanisms, and continuous synchronization operations must be examined in the context of overarching objectives of green computing and sustainable data infrastructure.

Overrepresentation of performance-centric studies is a major limitation. These contributions improve our understanding of scalability and efficiency but often sacrifice socio-technical dimensions. Thus, algorithmic bias, data ethics, and end-user empowerment are understudied. Many proposed frameworks, such as RAPO, DSP, and DADBS, are validated only in simulation or controlled environments, raising questions about their real-world viability[55], [45]. Geographic and infrastructural bias in much of the reviewed literature is another limitation. Few studies examine how distributed databases work in low-resource or Global South contexts, where bandwidth, electricity, and regulations vary. Existing solutions are less global and inclusive due to this neglect. User-centered evaluation is scarce in the field. What humans do with distributed systems, how they interpret outputs, how much they trust automated processes, and how system opacity affects decision-making have received little attention. Addressing these limitations is crucial to creating technically robust and socially legitimate distributed databases.

## 8. CONCLUSIONS

In the big data age, distributed database systems have rapidly evolved with a primary focus on improving performance measures, such as query response, latency, and throughput. Even if these advancements are essential for mission-critical and high-volume applications, placing too much focus on computing speed runs the danger of encouraging a technocentric viewpoint that ignores crucial elements like system transparency, architectural complexity, and environmental impact. Incorporating economic, environmental, and human-centered measures into performance reviews would be a more equitable strategy.

A major change has been made to database systems with the incorporation of artificial intelligence (AI) and machine learning (ML), which allows for adaptive query optimization and real-time anomaly identification. However, this advancement raises issues with possible bias, ethical governance, and algorithmic transparency. Addressing concerns of accountability and interpretability becomes essential as these systems transform from passive instruments to proactive decision-makers.

Despite progress, there is still a significant gap in the areas of governance-focused design, data synchronization across heterogeneous systems, and privacy-preserving computing. Especially in multi-cloud and global data settings, these elements are essential for guaranteeing trust, interoperability,

and compliance. To satisfy the demands of contemporary data governance, solutions such as secure multi-party computation (SMPC) need to be scaled and verified.

This research emphasizes the necessity of a paradigm change away from designs that are just focused on performance and toward distributed database systems that are more inclusive, robust, and morally sound. Future initiatives should aim to strike a balance between social responsibility and technological innovation, making sure that these systems are not only quick and scalable but also transparent, equitable, and long-lasting.

## REFERENCES

1. S. A. Bhat and N. F. Huang, “Big Data and AI Revolution in Precision Agriculture: Survey and Challenges,” *IEEE Access*, vol. 9, pp. 110209–110222, 2021, doi: 10.1109/ACCESS.2021.3102227.
2. Md. T. Islam and B. U. Khan, “Big Data and Analytics,” 2024, pp. 1–30. doi: 10.4018/978-1-6684-7366-5.ch048.
3. R. L. de C. Costa, J. Moreira, P. Pintor, V. dos Santos, and S. Lifschitz, “A Survey on Data-driven Performance Tuning for Big Data Analytics Platforms,” *Big Data Research*, vol. 25, Jul. 2021, doi: 10.1016/j.bdr.2021.100206.
4. R. K. Vankayalapati, “Zero-Trust Security Models for Cloud Data Analytics: Enhancing Privacy in Distributed Systems,” *Journal of Artificial Intelligence & Cloud Computing*, vol. 4, no. 1, pp. 1–8, Feb. 2025, doi: 10.47363/JAICC/2025(4)415.
5. N. Shakhovska, N. Boyko, Y. Zasoba, and E. Benova, “Big data processing technologies in distributed information systems,” in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 561–566. doi: 10.1016/j.procs.2019.11.047.
6. N. Deepa et al., “A Survey on Blockchain for Big Data: Approaches, Opportunities, and Future Directions,” Sep. 2020, [Online]. Available: <http://arxiv.org/abs/2009.00858>
7. O. T. Jinadu, O. V. Johnson, and M. Ganiyu, “Distributed Database System Optimization for Improved Service Delivery in Mobile and Cloud BigData Applications,” *International Journal of Computer Science and Mobile Computing*, vol. 10, no. 9, pp. 38–45, Sep. 2021, doi: 10.47760/ijcsmc.2021.v10i09.004.
8. A. / Ml et al., “Optimizing Production Efficiency in Manufacturing using Big.” [Online]. Available: <https://ssrn.com/abstract=5080585>
9. J. Wang, Y. Yang, T. Wang, R. Simon Sherratt, and J. Zhang, “Big data service architecture: A survey,” 2020, Taiwan Academic Network Management Committee. doi: 10.3966/160792642020032102008.
10. N. Aisyah and B. Hassan, “Orient Journal of Emerging Paradigms in Artificial Intelligence and Autonomous Systems Managing Data Dependencies in Cloud-Based Big Data Pipelines: Challenges, Solutions, and Performance Optimization Strategies.”
11. R. L. de C. Costa, J. Moreira, P. Pintor, V. dos Santos, and S. Lifschitz, “A Survey on Data-driven Performance Tuning for Big Data Analytics Platforms,” *Big Data Research*, vol. 25, Jul. 2021, doi: 10.1016/j.bdr.2021.100206.
12. H. Tang, “Intelligent Processing and Classification of Multisource Health Big Data from the Perspective of Physical and Medical Integration,” *Sci Program*, vol. 2022, 2022, doi: 10.1155/2022/5799354.
13. L. Wu, L. Yuan, and J. You, “Survey of Large-Scale Data Management Systems for Big Data Applications,” *J Comput Sci Technol*, vol. 30, no. 1, pp. 163–183, Jan. 2015, doi: 10.1007/s11390-015-1511-8.
14. A. A. Salih et al., “Deep Learning Approaches for Intrusion Detection,” *Asian Journal of Research in Computer Science*, pp. 50–64, Jun. 2021, doi: 10.9734/ajrcos/2021/v9i430229.
15. R. Avdal Saleh and H. Maseeh Yasin, “Comparative Analysis of AI and Machine Learning Applications in Modern Database,” *Engineering and Technology Journal*, vol. 10, no. 03, Oct. 2025, doi: 10.47191/etj/v10i03.21.
16. H. Gadde, “Optimizing Transactional Integrity with AI in Distributed Database Systems,” 2024.
17. H. B. Abdalla, “A brief survey on big data: technologies, terminologies and data-intensive applications,” Dec. 01, 2022, Springer Science and Business Media Deutschland GmbH. doi: 10.1186/s40537-022-00659-3.
18. P. Li and L. Zhang, “Application of big data technology in enterprise information security management,” *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-85403-6.
19. M. Shahnawaz and M. Kumar, “A Comprehensive Survey on Big Data Analytics: Characteristics, Tools and Techniques,” Mar. 05, 2025, Association for Computing Machinery. doi: 10.1145/3718364.
20. J. J. Pan, J. Wang, and G. Li, “Survey of Vector Database Management Systems,” Oct. 2023, [Online]. Available: <http://arxiv.org/abs/2310.14021>
21. L. M. Peltonen, H. von Gerich, E. Myllymäki, J. Walsh, and M. Medvecky, “Exploring Delays in Cardiac Care Processes Through Electronic Health Records,” *Stud Health Technol Inform*, vol. 316, pp. 1866–1870, Aug. 2024, doi: 10.3233/SHTI240795.

22. M. Del Giudice, R. Chierici, A. Mazzucchelli, and F. Fiano, “Supply chain management in the era of circular economy: the moderating effect of big data,” *International Journal of Logistics Management*, vol. 32, no. 2, pp. 337–356, 2020, doi: 10.1108/IJLM-03-2020-0119.
23. A. K. Sandhu, “Big Data with Cloud Computing: Discussions and Challenges,” *Big Data Mining and Analytics*, vol. 5, no. 1, Mar. 2022, doi: 10.26599/BDMA.2021.9020016.
24. A. Adadi, “A survey on data-efficient algorithms in big data era,” *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00419-9.
25. F. Berloco, V. Bevilacqua, and S. Colucci, “Distributed Analytics For Big Data: A Survey,” *Neurocomputing*, vol. 574, Mar. 2024, doi: 10.1016/j.neucom.2024.127258.
26. M. S. Mahmud, J. Z. Huang, S. Salloum, T. Z. Emara, and K. Sadatdiynov, “A survey of data partitioning and sampling methods to support big data analysis,” Jun. 01, 2020, Tsinghua University Press. doi: 10.26599/BDMA.2019.9020015.
27. Y. Himeur et al., “AI-big data analytics for building automation and management systems: a survey, actual challenges and future perspectives,” *Artif Intell Rev*, vol. 56, no. 6, pp. 4929–5021, Jun. 2023, doi: 10.1007/s10462-022-10286-2.
28. Z. Zhang, A. Megargel, and L. Jiang, “Performance Evaluation of NewSQL Databases in a Distributed Architecture,” *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3529740.
29. M. Poudel, R. P. Sarode, Y. Watanobe, M. Mozgovoy, and S. Bhalla, “Processing Analytical Queries over Polystore System for a Large Astronomy Data Repository,” *Applied Sciences (Switzerland)*, vol. 12, no. 5, Mar. 2022, doi: 10.3390/app12052663.
30. S. Ferreira, J. Mendonça, B. Nogueira, W. Tiengo, and E. Andrade, “Impacts of data consistency levels in cloud-based NoSQL for data-intensive applications,” *Journal of Cloud Computing*, vol. 13, no. 1, Dec. 2024, doi: 10.1186/s13677-024-00716-7.
31. M. Armbrust, A. Ghodsi, R. Xin, M. Zaharia, and U. Berkeley, “Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics.”
32. A. Ali, S. Naeem, S. Anam, and M. M. Ahmed, “A State of Art Survey for Big Data Processing and NoSQL Database Architecture,” *International Journal of Computing and Digital Systems*, vol. 14, no. 1, pp. 297–309, 2023, doi: 10.12785/ijcds/140124.
33. F. Q. Kareem et al., “SQL Injection Attacks Prevention System Technology: Review,” *Asian Journal of Research in Computer Science*, pp. 13–32, Jul. 2021, doi: 10.9734/ajrcos/2021/v10i330242.
34. Z. S. Ageed et al., “A State of Art Survey for Intelligent Energy Monitoring Systems,” *Asian Journal of Research in Computer Science*, pp. 46–61, Apr. 2021, doi: 10.9734/ajrcos/2021/v8i130192.
35. Oluwafemi Oloruntoba, “AI-Driven autonomous database management: Self-tuning, predictive query optimization, and intelligent indexing in enterprise it environments,” *World Journal of Advanced Research and Reviews*, vol. 25, no. 2, pp. 1558–1580, Feb. 2025, doi: 10.30574/wjarr.2025.25.2.0534.
36. T. Phiri, “Adaptive and Autonomous Systems in Advanced Computing A Future of Self-Optimizing Technologies,” *Journal of Advanced Computing Systems (JACS)* [www.scipublication.com](http://www.scipublication.com), vol. 3, no. 5, pp. 1–8, 2023, doi: 10.69987/JACS.2023.30501.
37. S. Chinamanagonda, “Cloud-native Databases: Performance and Scalability-Adoption of cloud-native databases for improved performance,” 2023.
38. G. Nookala Jp, “Journal of Computational Innovation Adaptive Data Governance Frameworks for Data-Driven Digital Transformations.” [Online]. Available: <https://researchworkx.com/index.php/jciVo14>
39. M. Asch et al., “Big data and extreme-scale computing: Pathways to Convergence-Toward a shaping strategy for a future software and data ecosystem for scientific inquiry,” Jul. 01, 2018, SAGE Publications Inc. doi: 10.1177/1094342018778123.
40. A. E. Topcu and A. M. Rmis, “Analysis and evaluation of the riak cluster environment in distributed databases,” *Comput Stand Interfaces*, vol. 72, Oct. 2020, doi: 10.1016/j.csi.2020.103452.
41. S. I. M. Mosharraf and M. A. Adnan, “Improving lookup and query execution performance in distributed Big Data systems using Cuckoo Filter,” *J Big Data*, vol. 9, no. 1, Dec. 2022, doi: 10.1186/s40537-022-00563-w.
42. T. Tiendrebeogo and M. Diarra, “Big Data Storage System Based on a Distributed Hash Tables System,” *International Journal of Database Management Systems*, vol. 12, no. 5, pp. 1–9, Oct. 2020, doi: 10.5121/ijdms.2020.12501.
43. “Distributed Database Systems for Large-Scale Data Management,” *Turkish Online Journal of Qualitative Inquiry*, 2023, doi: 10.52783/tojq.v11i4.10020.

44. S. A. Jowan, R. Faraj Swese, A. Yousf Aldabrzi, and M. Saad Shertil, “TRADITIONAL RDBMS TO NOSQL DATABASE: NEW ERA OF DATABASES FOR BIG DATA,” 2016. [Online]. Available: <https://www.researchgate.net/publication/355165835>
45. O. T. Jinadu, O. V. Johnson, and M. Ganiyu, “Distributed Database System Optimization for Improved Service Delivery in Mobile and Cloud BigData Applications,” *International Journal of Computer Science and Mobile Computing*, vol. 10, no. 9, pp. 38–45, Sep. 2021, doi: 10.47760/ijcsmc.2021.v10i09.004.
46. D. Hongwei and B. Ligetu, “Research on Distributed Storage Technology of Database Big Data Based on Cloud Computing,” in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Aug. 2021. doi: 10.1088/1742-6596/1982/1/012195.
47. X. Chang and H. Cui, “Distributed Storage Strategy and Visual Analysis for Economic Big Data,” *Journal of Mathematics*, vol. 2021, 2021, doi: 10.1155/2021/3224190.
48. N. Thamar, “Big data engineering and distributed systems Integration Perspective Big data engineering and distributed systems Integration Perspective NIYOMUKIZA Thamar”, doi: 10.13140/RG.2.2.28602.98245.
49. L. Zhang, T. Jia, M. Jia, Y. Li, Y. Yang, and Z. Wu, “Multivariate Log-based Anomaly Detection for Distributed Database,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, Aug. 2024, pp. 4256–4267. doi: 10.1145/3637528.3671725.
50. J. Olusegun and S. Brightwood, “Distributed Secure Multi-Party Computation (SMPC) for Cloud-Based Big Data Analytics,” 2024. [Online]. Available: <https://www.researchgate.net/publication/384885080>
51. H. S. Munawar, S. Qayyum, F. Ullah, and S. Sepasgozar, “Big data and its applications in smart real estate and the disaster management life cycle: A systematic analysis,” Jun. 01, 2020, MDPI. doi: 10.3390/bdcc4020004.
52. “Data Synchronization for Distributed Heterogeneous Database,” 2024.
53. H. Gadde, “Optimizing Transactional Integrity with AI in Distributed Database Systems,” 2024.
54. P. Aryan, R. Khatri, and V. Balakrishnan, “An Experimental Framework for Implementing Decentralized Autonomous Database Systems in Rust,” Dec. 2024, [Online]. Available: <http://arxiv.org/abs/2412.05078>
55. Y. Zhu, T. Xia, T. Zhu, Z. Zhao, K. Li, and X. Hu, “RAPO: An Automated Performance Optimization Tool for Redis Clusters in Distributed Storage Metadata Management,” *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3556240.
56. Y. Sato, “Database Architects in the Age of Distributed Systems: Principles, Patterns, and Practice.”
57. M. K. Kirino, “Design Principles and Evolving Roles of the Modern Database Architect in Scalable and Distributed Data Systems.” [Online]. Available: <https://ijarc.com/chiefeditor.ijarc@gmail.comhttps://ijarc.com/>
58. D. Esther and O. Evelyn, “Query Optimization Strategies for Heterogeneous Big Data Environments,” 2025. [Online]. Available: <https://www.researchgate.net/publication/390544586>
59. J. Andrew and A. Ailamaki, “Parallel and Distributed Query Execution for Big Data Workloads.” [Online]. Available: <https://www.researchgate.net/publication/390286214>
60. J. B. Adelusi and A. Adeleke, “Approximation Algorithms for Big Data in Distributed Databases,” 2025. [Online]. Available: <https://www.researchgate.net/publication/390544396>
61. M. Muniswamaiah, T. Agerwala, and C. Tappert, “Big Data in Cloud Computing Review and Opportunities,” *International Journal of Computer Science and Information Technology*, vol. 11, no. 4, pp. 43–57, Aug. 2019, doi: 10.5121/ijcsit.2019.11404.