WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

AI for Big Data Analytics: Enhancing Decision-Making in Large-Scale Datasets

Avinash Pal [1], Sagar Choudhary [2], Abdul Karim [3]

[1,3] B-Tech Student, Department of Computer Science and Engineering, Quantum University, Roorkee, India.

[2] Assistant Professor, Department of Computer Science and Engineering, Quantum University, Roorkee, India.

Abstract

The present growth in computer architecture has changed the face ofeducation,science and engineering. Technology does not stand still, and today,when assessing the economic development of the organization, university orindustry, the attention is paid to its willingness to use new ideas, especiallyin the field of Artificial Intelligence (AI) and big data analytics. In this research,the topic of AI and Big Data and its integration into decision-making process forcompetitiveness is examined. The emergence of Big Data and related analyticstechnologies led to changes in the business world. In today's business processes,extracting genuine value form AI and Big Data, and integrating it into the coredecision-making strategy plays a vital role. In this paper the necessary steps for thesuccessful integration of AI and Big Data and analytics are covered in Large scale data , by analyzingthe success factors from literature. Finally, its implications in real estate marketand education are discussed with a small-scale cases analysis. In a time when AI-driven decision-making is gaining prominence, the rapid increase in data generation across various industries presents substantial scalability challenges for big data processing. This research explores these issues with the goal of improving our ability to understand and manage vast amounts of data. It begins by emphasizing the vital role scalability plays in big data systems, underlining its importance for the effective operation of today's data-driven applications. The article identifies the primary obstacles to scalability, such as difficulties with data storage, processing efficiency, and resource management. To evaluate how well distributed computing systems like MapReduce, Apache Spark, and Apache Hadoop meet the growing demands of data processing, the study presents a detailed analysis of their performance. Additionally, it expands the discussion to include how modern technologies such as containerization and cloud infrastructure can alleviate scaling limitations. The objective of this study is to deliver a thorough overview of current tools, platforms, and methods to comprehensively address scalability problems in big data environments. Ultimately, it aims to improve data processing efficiency and strengthen AI-based decision-making in an age of overwhelming data proliferation.

Keywords: Artificial Intelligence, Big Data, Big Data analytics, big datatechnology decision-making, firm readiness, education, value-creation, successfactors, interactive decision support, analysis methods, real estate.

Introduction

The generation and accumulation of vast volumes of data in today's digital era have significantly accelerated the development of big data processing technologies. As more sectors and enterprises realize the strategic advantage of data-driven decision-making, the need for robust methods to analyze and process massive datasets has grown exponentially. However, this rapid growth in data volume has introduced a key technical challenge: scalability. Managing the delicate interplay among data expansion, computational efficiency, resource distribution, and system robustness is becoming increasingly complex, making the storage, management, and analysis of large-scale datasets more difficult than ever before.

Scalability stands as a cornerstone of modern data processing architectures [1]. It refers to a system's capability to adapt to rising workloads without compromising responsiveness, availability, or performance. In big data environments, scalability is not just a performance metric but a vital enabler of real-time analytics and insight generation. It provides the structural foundation for organizations to convert raw, unstructured data into actionable insights at a pace that matches the rapid demands of today's commercial and research environments [2].

This research aims to critically examine the scalability challenges in large-scale data processing systems. It seeks to analyze the core issues that arise when attempting to scale operations from gigabytes to petabytes and beyond. Understanding these obstacles is essential to developing new methods and strategies that can effectively reduce scalability bottlenecks and optimize the utilization of vast data repositories.

Emerging Challenges: The exponential growth of data across domains such as social media, e-commerce, scientific computing, and the Internet of Things (IoT) has brought several significant scalability challenges to the forefront. Massive datasets increasingly strain existing infrastructure, making efficient storage a major hurdle. In scenarios where computational resources lag behind the demands of real-time analytics, processing speed becomes a critical bottleneck. Furthermore, performance sustainability requires intelligent load balancing and job distribution across distributed nodes, making resource management a sophisticated task. As systems scale, maintaining data consistency, integrity, and fault tolerance also becomes significantly more complex [3].

This study will explore and evaluate the landscape of current technologies and frameworks that have emerged to address these issues. In particular, it will assess the performance and effectiveness of distributed computing paradigms such as MapReduce, Apache Spark, and Apache Hadoop in meeting the escalating demands of data-intensive applications. Moreover, the study will consider how modern solutions—including cloud computing platforms and containerization technologies like Docker and Kubernetes—can mitigate scalability limitations and foster flexible, resilient architectures for big data processing.

This study contributes to the ongoing discourse surrounding the scalability challenges, methodologies, and future directions in large-scale data processing. The primary contributions of this research can be summarized as follows:

1. A comprehensive analysis of the critical challenges associated with scalability.
2. A detailed evaluation of existing distributed computing models.
3. An extensive overview of current technologies and strategic approaches employed in big data environments.
4. Significant insights into enhancing AI-powered decision-making systems through improved scalability mechanisms.

The structure of the remainder of this paper is organized as follows: Section II provides a thorough literature review; Section III outlines the system architecture adopted in this study**;** Section IV discusses

Literature Review

The rapid growth of data across various industries has necessitated the evolution of data processing paradigms, compelling researchers and practitioners to confront the inherent challenges of scalability. This section synthesizes prior research efforts that have contributed to the development of methods and technologies designed to overcome these challenges, offering a critical perspective on their significance in the context of big data processing [4].One prominent approach, Dominant Resource Fairness (DRF), offers a max-min fairness model that ensures equitable distribution of computational resources. DRF enhances throughput while maintaining properties such as strategy-proofness, envy-freeness, and Pareto efficiency, thereby supporting more balanced and fair resource utilization [5].

Foundational work by Chen et al. [1] underscored the essential role of scalability in unlocking the full potential of large-scale datasets. They argued that, in the face of growing data volumes, ensuring scalability is even more critical than simply optimizing processing speed. However, their analysis would be strengthened by exploring specific scalability tactics and their sector-specific implications.The introduction of distributed processing frameworks such as MapReduce and Apache Spark by Dean and Ghemawat [3] and Zaharia et al. [2], respectively, marked a significant milestone in large-scale data analytics. These frameworks significantly advanced parallel data processing capabilities. Despite their impact, a more comprehensive evaluation of their limitations in handling heterogeneous workloads and diverse data types would present a more complete picture of their scalability potential [6].

Armbrust et al. [6] emphasized the transformative role of cloud computing in facilitating scalable resource provisioning. Their work introduced Resilient Distributed Datasets (RDDs), which enable fault-tolerant, in-memory computation across large clusters—an advancement particularly beneficial for iterative processing and data mining tasks. RDDs, implemented within Spark, support a wide range of computational models.Moreover, recent surveys [7–9] explore the scalability and limitations of serverless computing across various architectural layers. These analyses highlight its promise for elastic resource management, though they would benefit from deeper investigation into integration challenges and performance bottlenecks, especially in hybrid and edge computing environments.

This study builds upon prior work by not only synthesizing a broad array of scalability strategies but also by evaluating their applicability and constraints in contemporary big data scenarios. It addresses a key gap in the literature by focusing on the interplay between scalability and real-time data processing, offering a fresh lens through which to assess adaptability in evolving systems. By examining current innovations and proposing future research directions, this work contributes to advancing the discourse on AI-driven decision-making amid exponential data growth.Further insights from Ousterhout et al. and Ghodsi et al. [4, 5] explore fault-tolerant architectures and abstract resource management systems, allowing developers to concentrate on application logic rather than infrastructure complexity. In addition, edge computing technologies are being deployed to shift processing closer to the data source, thereby reducing latency and alleviating network congestion [9]. Integrating these distributed paradigms necessitates rethinking architectural design and communication frameworks.

Finally, monitoring and performance optimization emerge as crucial components in maintaining system health and ensuring scalability. Advanced monitoring tools offer visibility into resource consumption, execution timelines, and performance bottlenecks, laying the foundation for adaptive system tuning and efficient resource allocation.The rapid growth of data across various industries has necessitated the evolution of data processing paradigms, compelling researchers and practitioners to confront the inherent challenges of scalability. This section synthesizes prior research efforts that have contributed to the development of methods and technologies designed to overcome these challenges, offering a critical perspective on their significance in the context of big data processing [4].

One prominent approach, Dominant Resource Fairness (DRF), offers a max-min fairness model that ensures equitable distribution of computational resources. DRF enhances throughput while maintaining properties such as strategy-proofness, envy-freeness, and Pareto efficiency, thereby supporting more balanced and fair resource utilization [5]. Its adaptability across multi-resource environments makes it particularly relevant in heterogeneous systems where competing tasks demand varying types and quantities of resources.

Foundational work by Chen et al. [1] underscored the essential role of scalability in unlocking the full potential of large-scale datasets. They argued that, in the face of growing data volumes, ensuring scalability is even more critical than simply optimizing processing speed. However, their analysis would be strengthened by exploring specific scalability tactics and their sector-specific implications. For instance, incorporating domain-aware partitioning techniques or leveraging adaptive sampling methods could enhance performance in domains like bioinformatics or financial modeling.

The introduction of distributed processing frameworks such as MapReduce and Apache Spark by Dean and Ghemawat [3] and Zaharia et al. [2], respectively, marked a significant milestone in large-scale data analytics. These frameworks significantly advanced parallel data processing capabilities, enabling efficient execution of batch and iterative workloads across massive clusters. Despite their impact, a more comprehensive evaluation of their

limitations in handling heterogeneous workloads and diverse data types would present a more complete picture of their scalability potential [6]. Specifically, Spark's reliance on lineage-based fault recovery, while efficient for certain workloads, can lead to latency spikes during recomputation under frequent node failures.Armbrust et al. [6] emphasized the transformative role of cloud computing in facilitating scalable resource provisioning. Their work introduced Resilient Distributed Datasets (RDDs), which enable fault-tolerant, in-memory computation across large clusters—an advancement particularly beneficial for iterative processing and data mining tasks. RDDs, implemented within Spark, support a wide range of computational models, including SQL queries, streaming analytics, and machine learning pipelines. The modularity of this ecosystem has spurred extensive adoption in both academic and industrial settings.

Moreover, recent surveys [7–9] explore the scalability and limitations of serverless computing across various architectural layers. These analyses highlight its promise for elastic resource management, though they would benefit from deeper investigation into integration challenges and performance bottlenecks, especially in hybrid and edge computing environments. For example, cold start latency and stateless execution models often hinder real-time data stream processing. Addressing these challenges requires intelligent orchestration mechanisms and deeper integration with containerized backends.This study builds upon prior work by not only synthesizing a broad array of scalability strategies but also by evaluating their applicability and constraints in contemporary big data scenarios. It addresses a key gap in the literature by focusing on the interplay between scalability and real-time data processing, offering a fresh lens through which to assess adaptability in evolving systems. By examining current innovations and proposing future research directions, this work contributes to advancing the discourse on AI-driven decision-making amid exponential data growth.

The exponential growth of data across sectors—ranging from smart manufacturing and digital healthcare to finance and autonomous systems—has propelled the evolution of data processing paradigms. This transformation has intensified the focus on scalability, prompting researchers and practitioners alike to develop robust methodologies for managing increasing data volumes, diversity, and velocity. This section synthesizes key academic and technical contributions that have shaped the discourse on scalability, providing a critical lens through which to evaluate their relevance and limitations in modern big data ecosystems[4].

One influential contribution is the Dominant Resource Fairness (DRF) model, which addresses the challenge of fair and efficient resource allocation in multi-resource environments. Introduced by Ghodsi et al. [5], DRF builds on max-min fairness principles to balance computational loads while upholding strategy-proofness, envy-freeness, and Pareto efficiency. These properties make DRF particularly suitable for heterogeneous clusters, where workloads often contend for diverse resources such as CPU, memory, and disk I/O. Its widespread integration into resource schedulers like Apache Mesos underscores its practical significance.

Methodology

Figure 1 illustrates the architectural model for scalable big data processing. A core element of such architectures is their capacity to effectively address the ever-increasing volume of data while maintaining high performance. A robust and adaptable system architecture not only supports data growth but also ensures efficient computation, intelligent resource management, fault tolerance, and system responsiveness.

Key architectural components that underpin scalability in big data processing systems include the following:

A. Distributed Data Storage

To manage expanding data volumes, scalable architectures employ distributed storage techniques. File systems such as the Hadoop Distributed File System (HDFS) divide data into blocks and distribute them across multiple nodes. This approach supports both parallel processing and fault tolerance while optimizing storage efficiency. Moreover, contemporary architectures incorporate cloud-based storage platforms, which facilitate seamless scalability by provisioning additional storage on demand [7].

B. Data Processing Layer

The data processing layer is critical for executing large-scale computations. Frameworks such as MapReduce, introduced by Dean and Ghemawat [3], distribute computational tasks across nodes and consolidate the outcomes. Apache Spark, with its in-memory computation capabilities, significantly improves processing speed by reducing reliance on disk operations. This layer ensures task orchestration, load balancing, and parallelism, all of which are vital for scalable performance.

C. Resource Management and Allocation

Efficient resource allocation is essential for handling growing workloads. Resource managers such as Apache Hadoop YARN and Kubernetes dynamically assign compute resources based on task requirements. These platforms enable fine-grained resource allocation and isolation, which helps to prevent contention and reduce bottlenecks, thereby improving performance and scalability.

D. Fault Tolerance Mechanisms

In large-scale distributed systems, hardware or software failures are inevitable. Scalable architectures incorporate redundant data storage, node replication, and automatic failover mechanisms to preserve system reliability. Studies such as Ghodsi et al. [5] and others [8] highlight the necessity of designing systems that can maintain data integrity and recover gracefully in the face of failures.

E. Parallel Processing Frameworks

Parallel task execution lies at the heart of scalable architectures. Effective parallelism involves partitioning data and computational tasks, scheduling across distributed nodes, and synchronizing task outputs. Frameworks like MapReduce and Apache Spark exemplify this approach by facilitating high-speed, scalable computations that meet the demands of large datasets.

F. Elastic Scalability

Elastic scaling, enabled through cloud platforms, allows systems to dynamically provision or release resources in response to workload fluctuations. Serverless architectures further enhance this by abstracting infrastructure concerns and providing on-demand computation, thereby reducing latency and improving scalability. This paradigm plays a key role in optimizing performance and cost-efficiency in modern big data systems.

While these architectural strategies address many scalability challenges, limitations persist—particularly in handling real-time data processing, maintaining data consistency, and managing the rising cost of scaling operations. Serverless computing mitigates some of these issues by offering automated resource scaling, while edge computing processes data at or near the source, lowering latency and bandwidth usage. However, integrating such paradigms necessitates careful reengineering of system components and communication protocols.

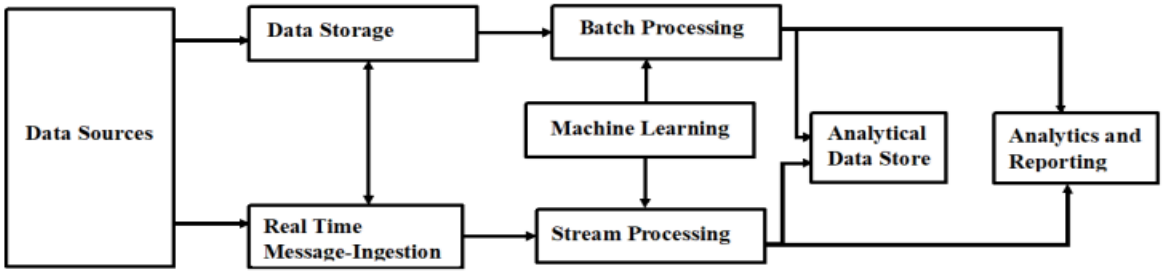

Figure 1. Big data architecture

To manage expanding data volumes, scalable architectures employ distributed storage techniques that support high availability and durability. File systems such as the Hadoop Distributed File System (HDFS) divide data into uniformly sized blocks and distribute them across multiple nodes within a cluster. This approach not only supports parallel processing and fault tolerance but also improves data locality, which enhances overall processing efficiency. Furthermore, modern architectures increasingly integrate cloud-based storage platforms like Amazon S3, Google Cloud Storage, or Microsoft Azure Blob Storage, which offer seamless elasticity, data redundancy, and geographic distribution for disaster recovery and compliance [7].
The data processing layer is central to performing complex computations on large datasets. Frameworks such as MapReduce, introduced by Dean and Ghemawat [3], distribute computational tasks across numerous nodes and aggregate the results to derive meaningful insights. Apache Spark further advances this model by enabling in-memory data processing, which reduces disk I/O and accelerates iterative algorithms common in machine learning and graph processing. This layer incorporates workload schedulers, task monitors, and execution engines to ensure optimal task orchestration, dynamic load balancing, and efficient utilization of computational resources. Efficient resource allocation plays a crucial role in sustaining performance across varying workloads. Tools such as Apache Hadoop YARN and Kubernetes provide robust frameworks for dynamically allocating compute, memory, and I/O resources based on real-time task requirements. These platforms support containerization, multitenancy, and auto-scaling, which prevent resource contention and system overload. Additionally, resource managers monitor system health and resource utilization to preemptively detect anomalies and ensure service continuity. Parallel execution of computational tasks is a cornerstone of big data scalability. This involves dividing workloads into independent subtasks that can be executed simultaneously across distributed processing units. Frameworks like MapReduce and Apache Spark implement this paradigm using partitioned datasets (RDDs in Spark) and parallel execution engines. These frameworks optimize task scheduling, manage inter-task dependencies, and aggregate results with minimal overhead. This parallelism significantly reduces processing time and enables handling of petabyte-scale datasets with high efficiency. Elastic scalability allows big data systems to adapt to changing workloads by automatically provisioning or decommissioning resources. Cloud platforms such as AWS, Azure, and Google Cloud provide autoscaling features that adjust computational power based on demand. Serverless computing, such as AWS Lambda and Google Cloud Functions, abstracts infrastructure management and allows developers to focus solely on business logic. This approach minimizes idle resources, lowers operational costs, and accelerates deployment cycles. Elastic scalability is particularly beneficial in environments with variable data ingress rates or batch workloads.

SCALABILITY KEY CHALLENGES

The exponential growth of data across numerous industries introduces several core challenges to scalable big data processing systems:

1. Data Volume and Storage: The ever-increasing data volume strains traditional storage systems. These systems lack the scalability and efficiency required to handle vast datasets. Implementing distributed storage solutions and scalable data retrieval methods is essential to prevent data bottlenecks.
2. Processing Speed: Timely processing of large datasets remains a significant hurdle. As datasets grow, processing times increase, potentially delaying actionable insights. Efficient solutions, such as parallel computing, optimized processing frameworks, and in-memory computations, are necessary to ensure timely data handling [11].
3. Resource Management: Allocating computational resources across multiple distributed nodes is inherently complex. Inefficient allocation leads to system lags and underutilized resources. Adaptive and dynamic resource management platforms are essential to balance loads and maximize resource usage.
4. Fault Tolerance: Failures in large-scale environments are unavoidable. Systems must incorporate redundancy mechanisms and error recovery processes to maintain integrity and availability, as emphasized in recent studies [5, 8].
5. Communication Overhead: Inter-node communication in distributed environments introduces latency and can become a bottleneck. Minimizing data transfer and enhancing data locality are critical to reducing overhead.
6. Complexity vs. Scalability: As systems expand, architectural and operational complexity also increases. Designing systems that scale effectively while remaining manageable and maintainable is crucial.
7. Cost Efficiency: Scalability often leads to increased operational costs, especially in cloud infrastructures. Balancing performance with financial sustainability through efficient provisioning is essential [10].

8. Data Consistency: Ensuring consistent data across distributed environments is a continuous challenge, especially during concurrent access or real-time processing.
9. Workload Adaptability: Big data workloads are often unpredictable. Scalable systems must be responsive and adaptable to sudden spikes or shifts in data processing demands.
10. Security and Privacy: With more data and nodes, safeguarding sensitive information becomes more complex. Ensuring compliance and security at scale is paramount.
11. Resource Contention: In multi-tenant environments, competing processes can degrade performance. Effective resource isolation and scheduling mechanisms are vital.
12. Maintainability: As systems grow in complexity, maintaining and debugging them becomes harder. Balancing scalability with ease of maintenance is a key design consideration [12].
13. Real-Time Processing Requirements: Many applications demand real-time data analysis. Achieving low latency and high throughput under such requirements is a scalability challenge.
14. Cross-Cluster Communication: Global systems face delays and synchronization issues across regions. Optimized communication protocols are needed to ensure system-wide efficiency.
15. Data Skew: Uneven data distribution among nodes can lead to imbalanced workloads and reduced performance. Mechanisms for detecting and correcting skew are essential [13].

Strategic Methodology for Tackling Scalability

A structured methodology is essential to effectively analyze and overcome these scalability issues:

- Problem Definition: Identify specific scalability pain points and define the research scope, including target technologies and application contexts.
- Literature Review: Analyze past studies and advancements in big data scalability. Assess existing frameworks and strategies from references [1-23].
- Data Analysis: Use real or synthetic datasets that reflect large-scale challenges. Understand key data dimensions (volume, velocity, variety, veracity).
- Technology Evaluation: Examine current solutions such as MapReduce, Apache Spark, serverless and edge computing, cloud services, and evaluate their scalability features.
- Experimentation: Develop test scenarios to simulate scalability stress points. Evaluate solutions using metrics like processing time, fault tolerance, and resource efficiency.
- Solution Design: Based on evaluation results, propose improvements or innovations—new architectural designs, optimized algorithms, or adaptive strategies.
- Optimization and Validation: Refine solutions by considering trade-offs between performance, resource use, and system complexity. Validate improvements through comparative analysis.
- Discussion and Future Scope: Contextualize findings in the broader landscape of scalable data processing. Suggest future research directions, including AI integration and hybrid computing strategies.

Alternative Approaches for Managing Large Data

Not all scenarios require full-scale big data solutions. In some cases, alternatives are more practical:

- Traditional Tools: For smaller datasets, tools like Excel or SQL databases suffice.
- Sampling: Analyze a subset of data to reduce resource needs.
- Data Aggregation: Simplify data by summarizing it into categories.
- Batch and Parallel Processing: Process data in manageable portions or distribute tasks across machines.
- Stream Processing: Real-time systems analyze data as it arrives.
- Data Warehousing: Specialized systems optimized for large queries help with data accessibility.
- Cloud Solutions: Even for smaller tasks, cloud platforms offer flexible scalability.
- Statistical Techniques: For limited data, infer patterns using statistical methods.

Each method suits different scenarios depending on data size, speed requirements, resources, and desired insights. Balancing trade-offs is key to efficient data strategy planning [14-23].

Figure 2. Big data scalability challenges

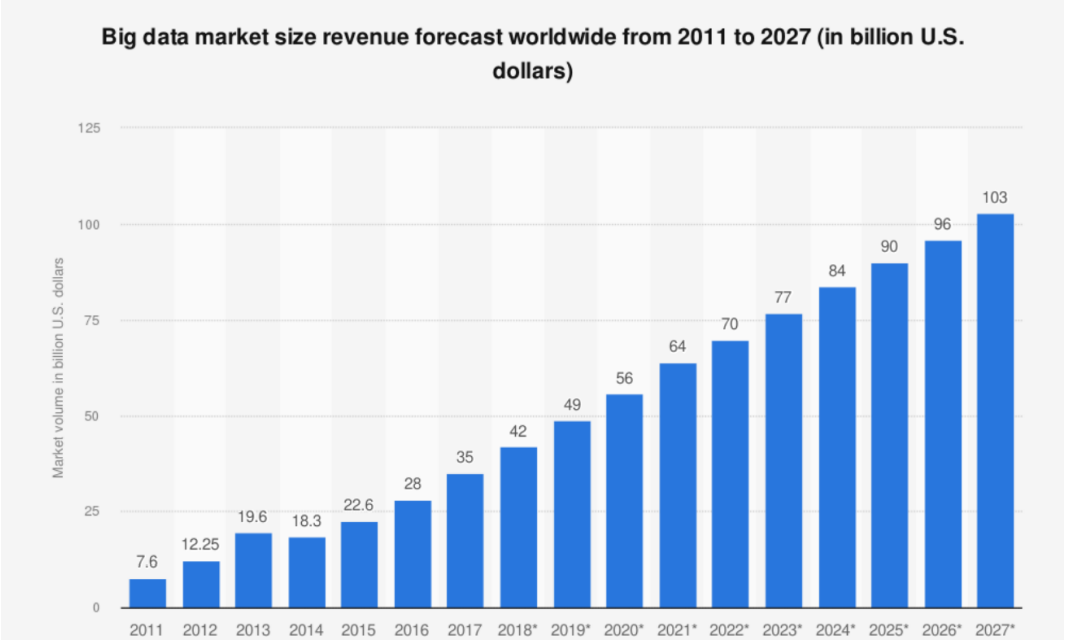

Figure 3. Big data growth projection

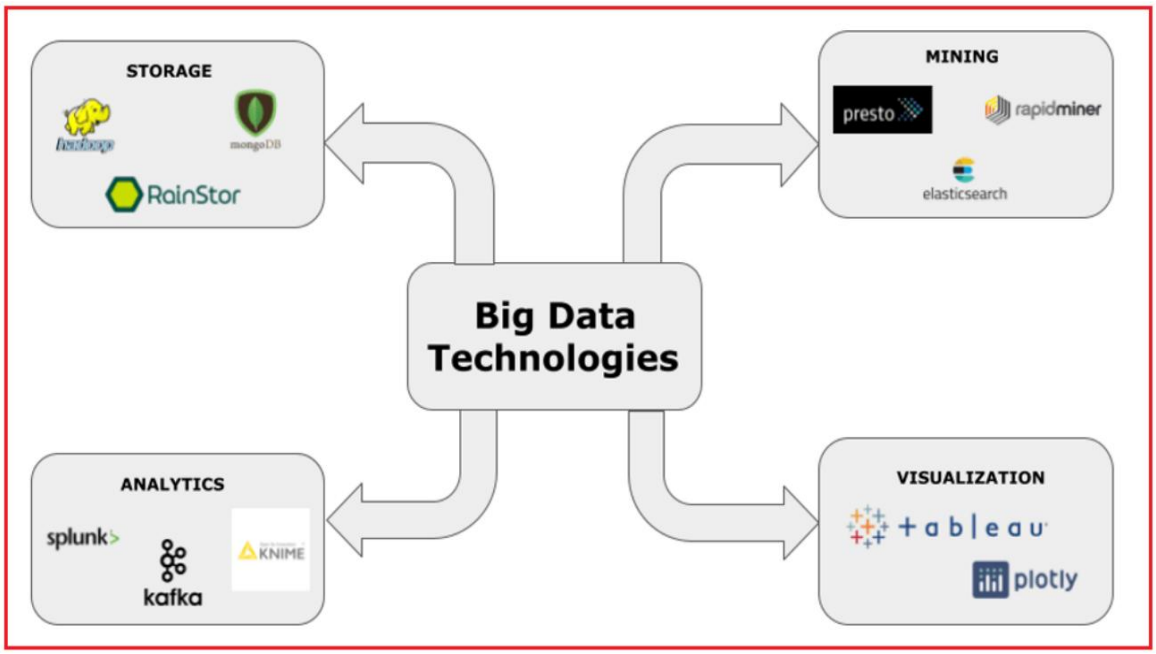

Figure 4. Big dataset technologies

Not all scenarios require full-scale distributed systems. Depending on workload characteristics and constraints, alternative strategies can be more pragmatic:

- Traditional Relational Tools (e.g., SQL, PostgreSQL) for structured, manageable datasets
- Sampling and Statistical Inference to extract meaningful trends without full data traversal
- Data Aggregation and Summarization to simplify complex datasets into actionable insights
- Batch and Micro-Batch Processing for handling periodic workloads efficiently
- Stream Processing Tools (e.g., Apache Kafka, Apache Storm) for real-time analytics
- Data Warehouses like Snowflake or Google BigQuery, optimized for analytical queries
- Cloud-Based On-Demand Resources for elasticity without upfront hardware investment
- Edge and Federated Processing to reduce data transmission and preserve privacy
- Compression and Encoding Techniques to reduce storage and network costs

Balancing complexity, performance, and cost requires a nuanced understanding of the problem context. A hybrid approach often yields the best results [14–23].

CONCLUSION

The rapid expansion of data generation across multiple industries has positioned big data processing as a cornerstone of modern technological progress. However, with this surge in data comes a critical need to address scalability, which remains a central challenge in realizing the full potential of large-scale data analytics. This study explored the intricate nature of scalability within big data systems, presenting approaches and techniques designed to mitigate these difficulties. Scalability is no longer optional—it is fundamental for extracting timely and actionable insights from massive datasets, especially in today's data-driven landscape. This paper examined key hurdles such as the escalating volume of data, the demand for faster processing, efficient distribution of computational resources, and the need for robust fault-tolerant systems. Achieving scalable data processing requires careful orchestration of these components. A thorough evaluation of current technologies and architectures was conducted, including the roles of distributed computing models like MapReduce and Apache Spark. The analysis also considered how cloud platforms and serverless computing models have reshaped scalability by enabling flexible resource management. Furthermore, edge computing was identified as a promising approach for minimizing latency and improving performance by bringing data processing closer to its source. Ultimately, this paper emphasizes that future advancements must prioritize flexible architectures, resilient fault-handling mechanisms, intelligent resource allocation, and efficient parallel processing. These elements will be key in crafting scalable solutions that can keep pace with the growing demands of big data. The unprecedented acceleration in data generation across diverse domains such as healthcare, finance, e-commerce, and social media has placed big data processing at the forefront of technological evolution. As organizations increasingly rely on data-driven decision-making, the need to efficiently manage, process, and extract insights from enormous datasets has become paramount. However, the inherent complexity of handling such vast amounts of information introduces significant scalability challenges that must be carefully addressed to ensure operational efficiency and strategic advantage. This paper underscores the importance of designing systems that are not only scalable but also adaptable, resilient, and cost-effective. Future research should continue to explore hybrid

architectures that combine centralized and decentralized models, optimize algorithmic efficiency, and enhance system monitoring for predictive scalability management. In conclusion, the pursuit of scalable big data solutions must focus on a holistic integration of distributed computing, cloud-native technologies, and intelligent automation. By aligning technical innovations with practical demands, organizations can unlock the full potential of their data assets, driving informed decisions and sustained innovation in an increasingly complex digital world.

REFERENCES

1. Chen, M., Mao, S., Liu, Y. (2014). Big data: A survey. Mobile Networks and Applications, 19(2): 171-209. https://doi.org/10.1007/s11036-013-0489-0
2. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I. (2010). Spark: Cluster computing with working sets. In 2nd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 10).
3. Dean, J., Ghemawat, S. (2004). MapReduce: Simplified data processing on large clusters. Communications of the ACM, 51(1): 107-113. https://doi.org/10.1145/1327452.1327492
4. Ousterhout, J.K., Gopalan, A., Rosenblum, M., Zhuang, H. (2015). The case for tiny tasks in computing. Communications of the ACM, 58(9): 45-53.
5. Ghodsi, A., Zaharia, M., Hindman, B., Konwinski, A., Shenker, S., Stoica, I. (2011). Dominant resource fairness: Fair allocation of multiple resource types. In 8th USENIX Symposium on Networked Systems Design and Implementation (NSDI 11).
6. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M. (2010). A view of cloud computing. Communications of the ACM, 53(4): 50-58. http://doi.acm.org/10.1145/1721654.1721672
7. Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N. (2009). The case for VM-based cloudlets in mobile computing. IEEE Pervasive Computing, 8(4): 14-23. https://doi.org/10.1109/MPRV.2009.82
8. Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauly, M., Franklin, M.J., Shenker, S., Stoica, I. (2012). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12), pp. 15-28.
9. Sivaraj, R., Ali, S.H., Buyya, R. (2019). The emergence of serverless computing: A survey. ACM Computing Surveys (CSUR), 52(6): 1-35.
10. Zaharia, M., Xin, R.S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M.J., Ghodsi, A., Gonzalez, J.E., Shenker, S.J., Stoica, I. (2016). Apache spark: A unified engine for big data processing. Communications of the ACM, 59(11):
11. White, T. (20152). Hadoop: The definitive guide (4 th. ed.). O'Reilly Media, Inc.
12. Shvachko, K., Kuang, H., Radia, S., Chansler, R. (2010). The hadoop distributed file system. In 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), Incline Village, NV, USA, pp. 1-10. https://doi.org/10.1109/MSST.2010.5496972
13. Zaharia, M., Borthakur, D., Sen Sarma, J., Elmeleegy, K., Shenker, S., Stoica, I. (2010). Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling. In Proceedings of the 5th European Conference on Computer Systems, pp. 265-278. https://doi.org/10.1145/1755913.1755940
14. Koh, K., Kim, K., Jeon, S., Huh, J. (2019). Disaggregated cloud memory with elastic block management. IEEE Transactions on Computers, 68(1): 39-52. https://doi.org/10.1109/TC.2018.2851565
15. Zaharia, M., Konwinski, A., Joseph, A.D., Katz, R.H., Stoica, I. (2008). Improving MapReduce performance in heterogeneous environments. In Osdi, 8(4): 7.
16. Zhang, X., Qi, L., Dou, W., He, Q., Leckie, C., Kotagiri, R., Salcic, Z. (2017). MRMondrian: Scalable multidimensional anonymisation for big data privacy preservation. IEEE Transactions on Big Data, 8(1): 125- 139. https://doi.org/10.1109/TBDATA.2017.2787661
17. Yang, C., Xu, X., Ramamohanarao, K., Chen, J. (2019). A scalable multi-data sources based recursive approximation approach for fast error recovery in big sensing data on cloud. IEEE Transactions on Knowledge and Data Engineering, 32(5): 841-854. https://doi.org/10.1109/TKDE.2019.2895612
18. Fawzy, D., Moussa, S.M., Badr, N.L. (2022). The internet of things and architectures of big data analytics: Challenges of intersection at different domains. IEEE Access, 10: 4969-4992. https://doi.org/10.1109/ACCESS.2022.3140409
19. Dean, J., & Ghemawat, S. (2008). *MapReduce: Simplified data processing on large clusters*. Communications of the ACM, 51(1), 107-113.
20. Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). *Spark: Cluster computing with working sets*. Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing.
21. Grolinger, K., Higashino, W. A., Tiwari, A., & Capretz, M. A. M. (2014). *Data management in cloud environments: NoSQL and NewSQL data stores*. Journal of Cloud Computing: Advances, Systems and Applications, 3(1), 1-24.