IRSTI 28.23.29

https://doi.org/10.51889/2959-5894.2026.94.2.019

A.B. Kassymova[1]* , R.K. Uskenbayeva[1] , A. Razaque[2] , S. Aliaskarov[3] , V. Elle[1]

[1] JSC Kazakh National Research Technical University named after K.I. Satbayev, Almaty, Kazakhstan
[2]Arkansas Tech University, Russellville, United States
[3] International IT University, Almaty, Kazakhstan
[*]*e-mail:* a.kassymova@satbayev.university

MODELING AND OPTIMIZATION OF A HYBRID HADOOP–SPARK ARCHITECTURE TO IMPROVE BIG DATA PROCESSING EFFICIENCY

Abstract

With the rapid growth of data volumes, heterogeneity, and intensity, the demands on architectures that provide not only high performance, but also robust scalability, efficient use of computing resources, and fault tolerance are increasing. This article examines a hybrid big data processing architecture that combines the Hadoop Distributed File System and Apache Spark operational processing mechanisms. The goal of the study is to develop a formalized approach to evaluating and optimizing the efficiency of a hybrid environment compared to the standalone use of Hadoop and Spark. The paper proposes a system of analytical models describing the processing speed, scalability, resource utilization, overhead, and overall efficiency of the hybrid architecture. Unlike studies that limit platform comparisons to general characteristics or isolated benchmarks, this article focuses on the relationship between data storage, inter-node communication, computing load, and cluster configuration parameters. It is demonstrated that combining Hadoop's distributed storage mechanisms with Spark's in-memory processing reduces the impact of disk I/O, improves resilience to increasing load, and ensures a more balanced use of memory and CPU resources. These results confirm that the hybrid architecture is a promising solution for building scalable analytics platforms designed to process heterogeneous data under variable and intensive workloads. The practical significance of this study lies in the potential use of the proposed models in the design and configuration of regional and enterprise big data analytics systems.

Keywords: big data, hybrid architecture, Hadoop, Spark, distributed computing, scalability, optimization, data processing efficiency, fault tolerance, computing resources.

А.Б. Касымова[1*], Р.К. Өскенбаева[1], А. Разак[2], С. Алиаскаров[3], В. Элле[1]
[1] Қ.И. Сәтбаев атындағы Қазақ Ұлттық техникалық зерттеу университеті, Алматы қ., Қазақстан
[2] Арканзас техникалық университеті, Расселвилл қ., АҚШ
[3] Халықаралық ақпараттық технологиялар университеті, Алматы қ., Қазақстан

ҮЛКЕН ДЕРЕКТЕРДІ ӨҢДЕУ ТИІМДІЛІГІН АРТТЫРУ ҮШІН HADOOP-SPARK ГИБРИДТІ АРХИТЕКТУРАСЫН МОДЕЛЬДЕУ ЖӘНЕ ОҢТАЙЛАНДЫРУ

Аңдатпа

Деректер көлемінің, гетерогенділіктің және қарқындылықтың тез өсуімен жоғары өнімділікті ғана емес, сонымен қатар сенімді масштабталуды, есептеу ресурстарын тиімді пайдалануды және ақауларға төзімділікті қамтамасыз ететін архитектураларға қойылатын талаптар артып келеді. Бұл мақалада Hadoop таратылған файлдық жүйесі мен Apache Spark операциялық өңдеу механизмдерін біріктіретін гибридті үлкен деректерді өңдеу архитектурасы қарастырылады. Зерттеудің мақсаты - Hadoop және Spark-ты дербес пайдаланумен салыстырғанда гибридті ортаның тиімділігін бағалау және оңтайландырудың формальды тәсілін әзірлеу. Мақалада өңдеу жылдамдығын, масштабталуын, ресурстарды пайдалануын, үстеме шығындарын және гибридті архитектураның жалпы тиімділігін сипаттайтын аналитикалық модельдер жүйесі ұсынылады. Платформаларды салыстыруды жалпы сипаттамалармен немесе оқшауланған эталондармен шектейтін зерттеулерден айырмашылығы, бұл мақала деректерді сақтау, түйіндер арасындағы байланыс, есептеу жүктемесі және кластер конфигурациясы параметрлері арасындағы байланысқа бағытталған. Hadoop таратылған сақтау

механизмдерін Spark жадындағы өңдеумен біріктіру дискінің енгізу/шығару әсерін азайтатыны, жүктеменің артуына төзімділікті жақсартатыны және жад пен CPU ресурстарын теңгерімді пайдалануды қамтамасыз ететіні көрсетілген. Бұл нәтижелер гибридті архитектураның айнымалы және қарқынды жұмыс жүктемелері кезінде гетерогенді деректерді өңдеуге арналған масштабталатын аналитикалық платформаларды құру үшін перспективалы шешім екенін растайды. Бұл зерттеудің практикалық маңыздылығы ұсынылған модельдерді аймақтық және кәсіпорындық үлкен деректерді талдау жүйелерін жобалау мен конфигурациялауда әлеуетті пайдалануда жатыр.

Түйін сөздер: үлкен деректер, гибридті архитектура, Hadoop, Spark, таратылған есептеулер, масштабталу, оңтайландыру, деректерді өңдеу тиімділігі, ақауларға төзімділік, есептеу ресурстары.

А.Б. Касымова[1*], Р.К. Ускенбаева[1], А. Разак[2], С. Алиаскаров[3], В. Элле[1]
[1] Казахский Национальный исследовательский технический университет им. К.И. Сатпаева, г. Алматы, Казахстан
[2] Арканзасский технологический университет, г. Расселвилл, США
[3] Международный университет информационных технологий, г. Алматы, Казахстан

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ ГИБРИДНОЙ АРХИТЕКТУРЫ HADOOP–SPARK ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ

Аннотация

В условиях стремительного роста объёмов, разнородности и интенсивности поступления данных повышаются требования к архитектурам, обеспечивающим не только высокую производительность, но и устойчивую масштабируемость, рациональное использование вычислительных ресурсов и отказоустойчивость. В статье рассматривается гибридная архитектура обработки больших данных, объединяющая распределённую файловую систему Hadoop Distributed File System и механизмы оперативной обработки Apache Spark. Цель исследования заключается в разработке формализованного подхода к оценке и оптимизации эффективности гибридной среды по сравнению с автономным использованием Hadoop и Spark. В работе предложена система аналитических моделей, описывающих скорость обработки, масштабируемость, использование ресурсов, накладные издержки и интегральную эффективность гибридной архитектуры. В отличие от исследований, ограничивающихся сравнением платформ на уровне общих характеристик или изолированных бенчмарков, в данной статье внимание сосредоточено на взаимосвязи между хранением данных, межузловым обменом, вычислительной нагрузкой и параметрами конфигурации кластера. Показано, что объединение механизмов распределённого хранения Hadoop с in-memory-обработкой Spark позволяет снизить влияние дискового ввода-вывода, повысить устойчивость к росту нагрузки и обеспечить более сбалансированное использование памяти и процессорных ресурсов. Полученные результаты подтверждают, что гибридная архитектура является перспективным решением для построения масштабируемых аналитических платформ, ориентированных на обработку гетерогенных данных в условиях переменных и интенсивных нагрузок. Практическая значимость исследования состоит в возможности использования предложенных моделей при проектировании и настройке региональных и корпоративных систем аналитики больших данных.

Ключевые слова: большие данные, гибридная архитектура, Hadoop, Spark, распределённые вычисления, масштабируемость, оптимизация, эффективность обработки данных, отказоустойчивость, вычислительные ресурсы.

Introduction

Main provisions. The study examines the effectiveness of a hybrid Hadoop–Spark architecture for distributed big data processing under conditions of growing volumes and heterogeneous data flows. A formalized architecture evaluation approach is proposed based on a joint analysis of processing time, scalability, resource utilization, and integrated efficiency. Comparative analysis results show that the hybrid architecture delivers higher execution speed compared to Hadoop and more robust scalability compared to standalone Spark as data volumes increase. It is established that combining HDFS distributed storage and Spark computing engines enables a more balanced load distribution and improves overall system efficiency. The practical significance of this work lies in the potential

application of the proposed approach to the design and optimization of analytical platforms for regional and corporate information systems.

In recent years, the development of digital services, sensor networks, electronic collaboration platforms, and enterprise information systems has led to a dramatic increase in the volume, speed, and diversity of data. Under these conditions, traditional centralized processing tools are proving insufficiently flexible to support high-load analytics, especially when scaling, processing streaming data, and integrating heterogeneous sources are required. Recent research emphasizes that big data architectures are increasingly focused on distributed processing, modularity, and adaptation to various types of computing workloads. Furthermore, the transition from classic Hadoop-based solutions to more flexible and cloud-compatible architectures is considered an important trend in the development of the entire big data ecosystem [1-3].

Among the most widely used big data processing platforms, Hadoop and Spark occupy a special place. According to modern comparative studies, the choice between them cannot be determined solely by overall performance, as efficiency depends on the nature of the tasks, fault-tolerance requirements, scheduling features, the computing model used, and infrastructure parameters. Hadoop retains its value as a reliable platform for distributed storage and batch processing, while Spark is particularly effective in scenarios requiring iterative computing, operational analytics, and intensive memory usage. At the same time, review studies show that neither platform is a universal solution for all classes of workloads in standalone mode, and architectural tradeoffs between speed, resilience, and resource efficiency remain the subject of active research [4-6].

For this reason, interest in hybrid architectures that combine the advantages of various data processing stacks is growing in the modern literature. Research in recent years has demonstrated a shift in focus from isolated analysis of individual platforms to a comparison of holistic architectural stacks, including Hadoop-based, modern, and cloud-based approaches. Such studies demonstrate that when designing an analytical environment, it is necessary to consider not only the computing core but also the entire data lifecycle: collection, preprocessing, storage, transmission, analytical processing, and presentation of results. Moreover, for distributed systems, data locality, inter-node exchange, transmission costs, and coordination between storage mechanisms and computational logic are essential [2, 7, 8].

The issue of optimizing task execution in a distributed environment adds further relevance to the problem. Recent studies show that the overall efficiency of big data platforms depends significantly on the quality of load balancing, scheduling strategies, and mechanisms that reduce data transfer losses and uneven distribution of computing resources. Intelligent scheduling methods are actively developing for Spark-based environments, while approaches aimed at improving data locality and reducing task execution overhead are being developed for Hadoop-like systems. This suggests that research into architectural efficiency should not be limited to comparing the execution times of individual tests; a more formal consideration of the relationship between storage, processing, network exchange, and resource utilization is required [9-10].

An analysis of modern publications reveals that, despite a significant number of review and empirical studies, the literature remains insufficiently developed to formally describe the hybrid Hadoop-Spark architecture as a single optimized system. Many studies either limit themselves to comparing platforms under specific benchmark conditions or examine architectural stacks at the macro level, without integrating performance, scalability, resource utilization, and overhead into a single model. However, this formulation of the problem is particularly important for systems operating with heterogeneous data streams and variable loads, where the practical value of the architecture is determined not by a single metric, but by the consistency of several characteristics simultaneously [1, 4, 7, 8].

Therefore, the objective of this study is to develop an approach to modeling and optimizing a hybrid Hadoop-Spark architecture to improve the efficiency of big data processing in a distributed environment. The scientific novelty of this work lies in its approach to the hybrid architecture, which is considered not only as a set of technological components but also as a formalized system for which

metrics for processing speed, scalability, resource utilization, and overall efficiency can be defined and agreed upon. The practical significance of this study lies in the potential application of the proposed approach in the design and configuration of analytical platforms for regional and corporate information systems that require robust computations on large and heterogeneous data sets.

Research methods

This study utilized a comparative analytical approach to evaluate the effectiveness of a hybrid Hadoop–Spark architecture relative to standalone Hadoop and Spark implementations. The research methodology included developing an architectural model, selecting a metric system, constructing formalized dependencies for quantitative performance evaluation, and conducting a series of computational experiments in a controlled cluster environment. The object of study was a distributed big data processing system combining the long-term storage capabilities of the Hadoop Distributed File System and the high-speed processing mechanisms of Apache Spark.

In the first stage, an architectural diagram of the environment was created. In its basic configuration, Hadoop was used as a distributed storage and batch processing platform using HDFS, MapReduce, and YARN. Spark was considered as a computational processing environment focused on in-memory tasks using Spark Core, Spark SQL, Spark Streaming, MLlib, and GraphX. The hybrid architecture was built as an integrated environment, with data storage in HDFS and computational processing performed by Spark with direct access to the distributed storage. This arrangement combined the fault tolerance and scalability of Hadoop with the advantages of Spark's in-memory approach.

In the second stage, the experimental environment was defined (Table 1). The study was conducted on a cluster of 6-10 computing nodes, each equipped with Intel Xeon E5-2630 v4 processors, 64 GB of RAM, and a 10 Gbps network connection. Hadoop 3.2 and Spark 3.1 were used for the software, with an HDFS replication factor of three. Datasets ranging from 500 GB to 3 TB in size, including structured, semi-structured, and unstructured data, were used to simulate typical distributed analytics scenarios. Streaming data loading was simulated using Apache Flume and Apache Kafka, while batch integration was simulated via HDFS. Ganglia and Apache Ambari were used to monitor cluster status and record metrics.

Table 1. Experimental environment and benchmark configuration

Cluster size	*6–10 nodes*
CPU	*Intel Xeon E5-2630 v4*
RAM per node	*64 GB*
Network	*10 Gbps*
Hadoop version	*3.2*
Spark version	*3.1*
HDFS replication factor	*3*
Dataset size	*500 GB – 3 TB*
Data types	*Structured, semi-structured, unstructured*
Streaming tools	*Apache Kafka, Apache Flume*
Monitoring tools	*Ganglia, Apache Ambari*
Benchmarks	*TPC-H, HiBench*

In the third stage, a test workload system was established. Standardized benchmarks, TPC-H and HiBench, were used to ensure reproducibility and comparability of results. TPC-H was used to evaluate the performance of analytical queries on structured data, while HiBench was used to model a wide range of operations: batch processing, streaming analytics, machine learning tasks, and graph processing. This combination allowed us to study the behavior of the architectures under different workload profiles and varying intensities of inter-node communication.

In the fourth stage, a system of quantitative metrics was developed. The key indicators were processing time, scalability, computational resource utilization, and the overall efficiency of the hybrid architecture. Processing speed was defined as the ratio of the number of completed operations to the total task execution time:

$$P = \frac{\sum O_j}{T} \tag{1}$$

where O_j – the number of operations completed within task j, T – the total computation time.

The scalability indicator was defined as the ratio of performance gain to data volume gain:

$$S = \frac{P_{new}/P_{base}}{D_{new}/D_{base}} \tag{2}$$

where P_{new} and P_{base} – the system performance under the new and base load volumes, D_{new} and D_{base} – the corresponding data volumes. The integrated resource utilization score was calculated as an average function of the processor, memory, and input-output subsystem load:

$$R = \frac{1}{p}\sum_{k=1}^{p}(CPU_k + MEM_k + IO_k) \tag{3}$$

where p – number of active processes, CPU_k, MEM_k, IO_k – resource utilization indicators for the process k. For the hybrid architecture, an additional efficiency metric was introduced, considering not only the volume of operations performed and execution time, but also the size of the data processed, inter-node transfers, and system overhead:

$$E_h = \frac{\sum O_i}{T+\alpha\sum Tr_m+\beta\sum Oh_l} \tag{4}$$

where O_i – the number of executed operations, T – the total processing time, Tr_m – the cost of transferring data segments between nodes, Oh_l – the system overhead, and α and β are the coefficients that measure the impact of data transfer and service costs on the overall efficiency. Unlike a simple measurement of execution time, this metric allows one to evaluate the architecture as a holistic distributed system, in which computational performance depends on the consistency of storage, processing, and communications.

In the fifth stage, the three architectures were compared using a single procedure. All computational scenarios were run under identical hardware conditions and with comparable data volumes. For each configuration, the average execution time, result variability, the nature of performance changes with data growth, and the level of computational resource utilization were recorded. This minimized the influence of external factors and ensured the validity of the comparative analysis. Attention was paid to cases in which Spark demonstrated acceleration due to in-memory processing but began to lose stability with increasing data volume, while Hadoop, conversely, maintained storage stability but suffered from execution speed. The hybrid architecture was analyzed as a compromise model capable of providing a more balanced performance under variable load conditions. The proposed methodology combines architectural analysis, experimental comparison, and a formalized metrics system. This allows us to move beyond a descriptive comparison of Hadoop, Spark, and their hybrid integrations to a more rigorous assessment of the conditions under which a hybrid architecture truly delivers increased efficiency in big data processing.

To ensure methodological consistency, the study was organized as a sequential workflow including architecture design, experimental setup, workload generation, metric definition, comparative evaluation, and interpretation of results. This workflow provided a structured basis for the formal and experimental assessment of the Hybrid Hadoop–Spark architecture. The main stages of the research process are presented in Figure 1.

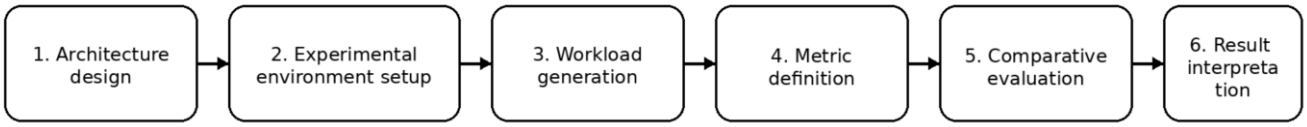

Figure 1. Research workflow for evaluating the Hybrid Hadoop–Spark architecture

As shown in Figure 1, the proposed methodology combines architectural modeling with benchmark-based evaluation and analytical interpretation. Such a sequence makes it possible to move from a conceptual description of the Hybrid architecture to a quantitative comparison with standalone Hadoop and Spark under controlled workload conditions.

Results of the study

A comparative analysis showed that the hybrid architecture delivers superior execution time and overall efficiency while maintaining robust scalability. The most pronounced advantage is observed in mixed-workload scenarios requiring a combination of long-term storage, intensive analytical processing, and adaptability to data growth. To quantify the performance differences between the evaluated architectures, the average execution time and the variability of results were measured under identical workload conditions. Table 2 summarizes the comparative results for Hadoop, Spark, and the Hybrid architecture.

Table 2. Comparison of execution time and stability of results

System	*Average execution time, s*	*Standard deviation, s*	*Reduced time relative to Hadoop, %*
Hadoop	*1200*	*45*	*0,0*
Spark	*740*	*38*	*38,3*
Hybrid	*690*	*32*	*42,5*

Note: Reduced time relative to Hadoop was calculated as the percentage decrease in average execution time with Hadoop taken as the baseline.

From the data in Table 2 it follows that the hybrid architecture provides the minimum average execution time and the least variability of results, which indicates more stable operation of the system.

The lower standard deviation observed for the Hybrid architecture indicates not only improved execution speed but also more consistent behavior across repeated runs. This suggests that the integration of distributed storage and accelerated processing reduces sensitivity to workload fluctuations and improves execution stability. To complement the quantitative results, a qualitative comparison was performed across the main architectural criteria relevant to big data processing. The results of this comparison are presented in Table 3.

Table 3. Qualitative comparative assessment of architectures based on key criteria

Criteria	*Hadoop*	*Spark*	*Hybrid*
Batch processing of big data	*High*	*Medium*	*High*
Streaming analytics	*Low*	*High*	*High*
Iterative calculations	*Low*	*High*	*High*
Scalability with data growth	*High*	*Medium*	*High*
Robustness under limited memory	*High*	*Low*	*Medium/High*
Integral efficiency	*Medium*	*High*	*Highest*

Note: High, Medium, and Low indicate qualitative comparative assessments derived from the experimental results and analytical interpretation of the considered architectures.

The results in Table 3 show that the standalone platforms have a pronounced specialization, while the hybrid architecture exhibits more general characteristics and is better adapted to heterogeneous processing scenarios. To assess the performance of the considered architectures, the average processing time was measured under identical workload conditions. The comparison included standalone Hadoop, standalone Spark, and the proposed Hybrid architecture. The obtained results are presented in Figure 2.

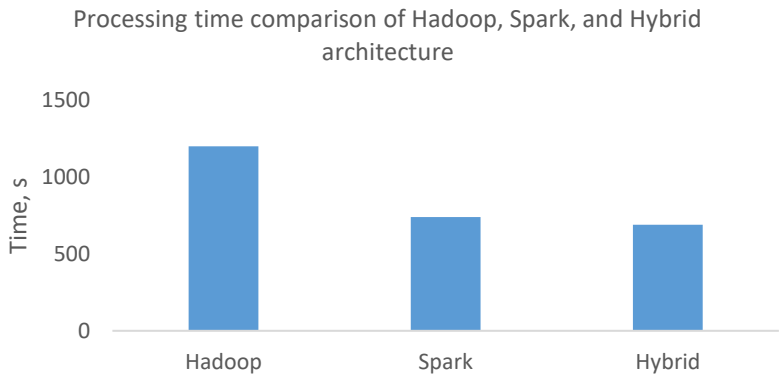

Figure 2. Processing time comparison of Hadoop, Spark, and Hybrid architecture

As shown in Figure 2, the Hybrid architecture achieved the lowest processing time among the evaluated systems. Hadoop demonstrated the highest execution time due to its dependence on disk-based operations, whereas Spark significantly improved performance through in-memory processing. At the same time, the Hybrid architecture combined the advantages of distributed storage and accelerated computation, which resulted in the best overall performance.

To further interpret the scalability behavior of the evaluated architectures, a trend-based comparative visualization was constructed. Figure 3 does not present raw benchmark measurements for each individual point; rather, it reflects the observed scalability trend derived from the experimental results and their analytical generalization under increasing data volumes. This representation is used to illustrate the relative behavior of Hadoop, Spark, and the Hybrid architecture as workload size grows.

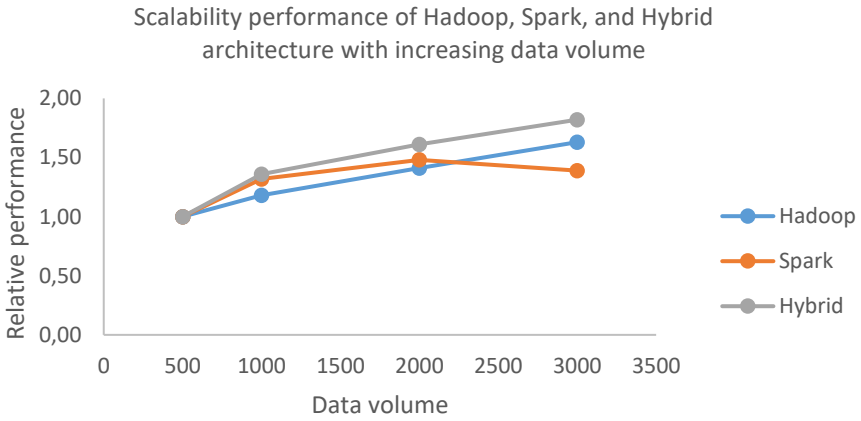

Figure 3. Scalability performance of Hadoop, Spark, and Hybrid architecture with increasing data volume

As illustrated in Figure 3, Hadoop maintains a relatively stable scalability profile, although its performance growth remains moderate. Spark demonstrates stronger initial acceleration, but its relative efficiency declines when the workload approaches memory limitations. The Hybrid

architecture preserves a more consistent upward trend, indicating a better balance between storage scalability and computational speed under increasing data volumes.

To complement the performance analysis, resource utilization profiles were compared across the evaluated architectures. Figure 4 summarizes the relative CPU load, memory consumption, and I/O overhead observed during the experimental runs.

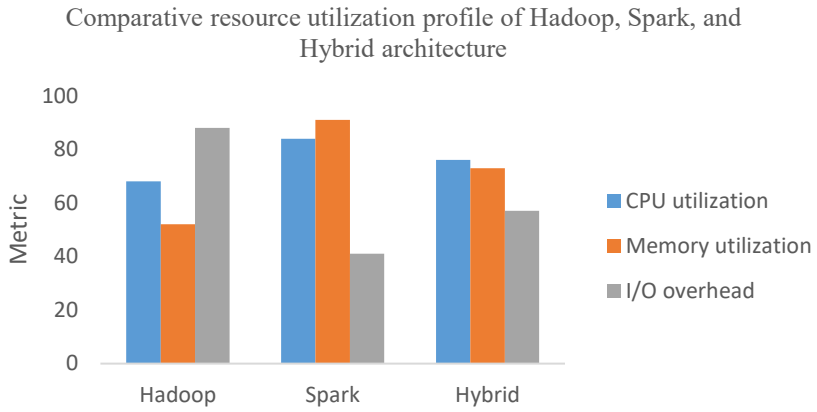

Figure 4. Comparative resource utilization profile of Hadoop, Spark, and Hybrid architecture

As shown in Figure 4, Hadoop is characterized by relatively high I/O overhead, while Spark demonstrates the highest memory consumption. The Hybrid architecture provides a more balanced resource profile, reducing I/O intensity compared to Hadoop and avoiding the excessive memory dependence observed in standalone Spark.

The values shown in Figure 4 reflect the comparative resource utilization profile observed in the experimental environment and are used to illustrate the relative balance of CPU, memory, and I/O demands across the evaluated architectures.

The graphical presentation of the results confirms that the hybrid architecture combines high Spark processing speed with greater resilience to load growth than the standalone Spark configuration, while outperforming Hadoop in execution time in most computing scenarios.

To complement the comparative performance analysis, the evaluated architectures were also interpreted in terms of their practical suitability for different workload scenarios. This makes it possible to relate the obtained results not only to benchmark performance, but also to the most appropriate application domains of each platform. The corresponding recommendations are summarized in Table 4.

Table 4. Recommended application scenarios for the evaluated architectures

Architecture	*Most suitable scenario*
Hadoop	*Large-scale batch processing and long-term storage*
Spark	*Iterative analytics and real-time data processing*
Hybrid	*Mixed workloads with variable data volume and heterogeneous sources*

Table 4 shows that each architecture has its own area of strongest applicability. Hadoop is more appropriate for storage-oriented and batch scenarios, Spark is preferable for fast iterative and streaming analytics, whereas the Hybrid architecture is the most suitable option for environments where storage reliability, computational speed, and adaptability to heterogeneous workloads are required simultaneously.

Discussion

The obtained results show that the differences between Hadoop, Spark, and the Hybrid architecture are determined not only by computational performance, but also by the way storage and processing are organized. Hadoop remains effective for large-scale batch workloads due to its distributed storage and fault tolerance, but its reliance on disk-based operations limits responsiveness in low-latency scenarios. Spark provides higher performance in iterative and real-time analytics tasks, although its efficiency is more sensitive to memory availability and cluster configuration.

Against this background, the Hybrid architecture demonstrates a more balanced operational profile. Its advantage lies in the functional separation between storage and computation: HDFS provides durable distributed storage, while Spark accelerates computationally intensive processing steps. This combination makes it possible to preserve high processing speed while maintaining stability under increasing and heterogeneous workloads.

An important implication of the study is that the assessment of big data architectures should not be limited to execution time alone. A more informative comparison requires the joint consideration of scalability, resource utilization, and overall efficiency. From this perspective, the Hybrid architecture can be viewed as an optimizable system in which performance depends on the coordination of storage, computation, and data exchange mechanisms.

At the same time, the results should be interpreted with some caution. The experiments were conducted for a specific cluster configuration and a selected set of workloads, so the observed effects may vary under different infrastructure settings. Further research may therefore focus on adaptive tuning of the Hybrid environment and on extending the analysis to broader workload and deployment scenarios.

The study has several limitations. The experiments were performed on a fixed cluster configuration and a selected set of workloads, which may not fully represent all deployment conditions. In addition, the scalability visualization reflects the observed experimental trend and its analytical generalization rather than a full raw benchmark series for every workload scenario.

Conclusion

This article examines an approach to modeling and evaluating the effectiveness of a hybrid Hadoop-Spark architecture for distributed big data processing. The analysis revealed that standalone Hadoop and Spark platforms offer distinct advantages for certain classes of tasks; however, their independent use does not always provide the optimal balance of performance, scalability, and resource efficiency.

The study's results confirmed that a hybrid architecture combining Hadoop's distributed storage and Spark's computational engines enables more robust and balanced processing performance. Compared to Hadoop, it delivers higher task execution speeds, and compared to Spark, it demonstrates greater resilience to growing data volumes and increasingly complex computational scenarios. This makes the hybrid approach particularly promising for information systems that combine large data volumes, heterogeneous source structures, and variable workloads.

The scientific significance of this work lies in the fact that the hybrid architecture is considered as a formalizable system for which metrics for processing speed, scalability, resource utilization, and overall efficiency can be defined. The practical significance lies in the potential use of the proposed approach in designing and configuring analytical platforms for regional and corporate information systems. Prospects for further research include deepening adaptive task distribution models, taking network latency into account, and developing methods for intelligently optimizing hybrid environment configurations based on data characteristics and computational load.

Acknowledgements

This research is funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (grant AP23489233 - “SmartBuy Connect: AI-based intelligent group purchasing system”).

References

[1] Badshah A., Daud A., Alharbey R., Banjar A., Bukhari A., Alshemaimri B. Big data applications: overview, challenges and future. Artificial Intelligence Review. – 2024. – Vol. 57. – Article 290. – DOI: 10.1007/s10462-024-10938-5.

[2] Elouataoui W., Beni-Hssane A., El Fissaoui M., Ouhbi B. Empirical Evaluation of Big Data Stacks: Performance and Design Analysis of Hadoop, Modern, and Cloud Architectures. Big Data and Cognitive Computing. – 2025. – Vol. 10. – No. 1. – Article 7. – DOI: 10.3390/bdcc10010007

[3] Plazotta M., Zwettler M., Vossen G. Data Architectures in Cloud Environments. Datenbank-Spektrum. – 2024. – Vol. 24. – P. 243-247. – DOI: 10.1007/s13222-024-00490-5.

[4] Khalid M., Yousaf M.M. A Comparative Analysis of Big Data Frameworks: An Adoption Perspective. Applied Sciences. – 2021. – Vol. 11. – No. 22. – Article 11033. – DOI: 10.3390/app112211033.

[5] Ullah F., Dhingra S., Xia X., Babar M.A. Evaluation of Distributed Data Processing Frameworks in Hybrid Clouds. Journal of Network and Computer Applications. – 2024. – Vol. 224. – Article 103837. – DOI: 10.1016/j.jnca.2024.103837.

[6] Ali H.A.E.-S., Alham M.H., Ibrahim D.K. Big data resolving using Apache Spark for load forecasting and demand response in smart grid: a case study of Low Carbon London Project. Journal of Big Data. – 2024. – Vol. 11. – Article 59. – DOI: 10.1186/s40537-024-00909-6.

[7] Berloco F., Moschetto G., Rundo F., Trenta F., Vitabile S. Distributed Analytics for Big Data: A Survey. Neural Networks. – 2024. – Vol. 175. – P. 106269. – DOI: 10.1016/j.neucom.2024.127258.

[8] Shahnawaz M., Fatima A., Purohit G.N., et al. A Comprehensive Survey on Big Data Analytics. ACM Computing Surveys. – 2025. – Vol. 57. – Article 196. – P. 1-33. – DOI: 10.1145/3718364.

[9] Verma V.P., Abbas H., Alqahtani S., et al. Optimizing Spark Job Scheduling with Distributional Deep Learning in Cloud Environments. Journal of Cloud Computing. – 2025. – Vol. 14. – P. 59. - DOI: 10.1186/s13677-025-00773-6.

[10] Ma C., Zhao M., Zhao Y. An Overview of Hadoop Applications in Transportation Big Data. Journal of Traffic and Transportation Engineering (English Edition). – 2023. – Vol. 10. – No. 5. – P. 900–917. – DOI: 10.1016/j.jtte.2023.05.003.