REVIEW ARTICLE

An Overview of Data Persistence Approaches for Enterprise Web Applications

Zhiwei Xu[1,*] **and Ming Lei**[2]

[1] Department of of Railway Transportation, Shaanxi Railway Institute, Weinan 714000, China
[2] School of Computer Science, Shaanxi Normal University, Xi'an 710119, China

Abstract

In the era of digital transformation, enterprise web applications have become indispensable tools for business operations, necessitating the efficient and reliable management of vast amounts of data. Data persistence is critical to ensure consistency, security, and scalability, especially in complex environments involving high concurrency and sensitive information. This paper reviews the key requirements for data persistence in enterprise-level web applications, such as reliability, security, scalability, and high availability, while addressing the challenges posed by modern business needs. Various persistence solutions, including relational databases, NoSQL databases, and distributed storage systems, are examined with respect to their performance in these critical areas. By providing a comprehensive analysis of these solutions, this paper aims to guide enterprises in selecting the most suitable data persistence approach to ensure long-term stability and regulatory compliance.

Academic Editor:
Jinchao Chen

Submitted: 11 October 2024
Accepted: 18 November 2024
Published: 07 December 2024

Vol. 2, **No.** 1, 2025.
10.62762/TCS.2024.529749

***Corresponding author:**
✉ Zhiwei Xu
yiqingyi@163.com

Keywords: data persistence, enterprise web applications, relational databases, distributed databases.

1 Introduction

In today's wave of digital transformation, Web applications have become a core component of enterprise business operations and management. From e-commerce platforms, Enterprise Resource Planning (ERP) systems, to Customer Relationship Management (CRM) systems, etc., these enterprise-level Web applications need to process massive amounts of data every day [1]. This data not only drives operational decisions, but also directly affects customer experience and business success or failure. Therefore, how to efficiently, securely, and reliably store these data in a persistent manner has become a key issue in the design of enterprise information systems.

Data persistence for enterprise-level Web applications involves multiple complex requirements. First of all, enterprises must ensure the reliability of data when dealing with massive amounts of data. Whether it is a system crash, hardware failure or network interruption, it should not affect the integrity of the data, and must ensure that the data can be recovered in a timely manner and ensure its consistency. Secondly, security is also one of the core requirements of enterprise data persistence, especially in scenarios involving sensitive data (e.g., financial data, user privacy), the data must be encrypted and access rights

Citation
Xu, Z., & Lei, M. (2025). An Overview of Data Persistence Approaches for Enterprise Web Applications. *IECE Transactions on Computer Science*, 2(1), 10–17.

© 2024 IECE (Institute of Emerging and Computer Engineers)

should be strictly controlled. In addition, as enterprise business continues to expand, scalability becomes an essential requirement, and the data storage system should be able to flexibly cope with scenarios where data volume grows and concurrent accesses surge, without significantly impacting performance.

However, enterprise-class data persistence faces many challenges in meeting these demands. For example, in highly concurrent scenarios, how to ensure high system performance while maintaining data consistency? For unstructured or semi-structured data (e.g., files, images, logs, etc.), are traditional relational database solutions still applicable? Meanwhile, in distributed architectures and cloud computing environments, how to achieve reliable transaction management and data synchronization among multiple data nodes?

To address these challenges, organizations can choose from a variety of different data persistence solutions, as shown in Figure 1, including traditional relational databases (e.g., MySQL, PostgreSQL), emerging NoSQL databases (e.g., MongoDB, Cassandra), distributed databases (e.g., CockroachDB, TiDB), and object storage (e.g., Amazon S3, Azure Blob Storage), and so on [2]. Each technology has its own unique advantages and application scenarios, and enterprises need to make reasonable choices based on their own business needs.

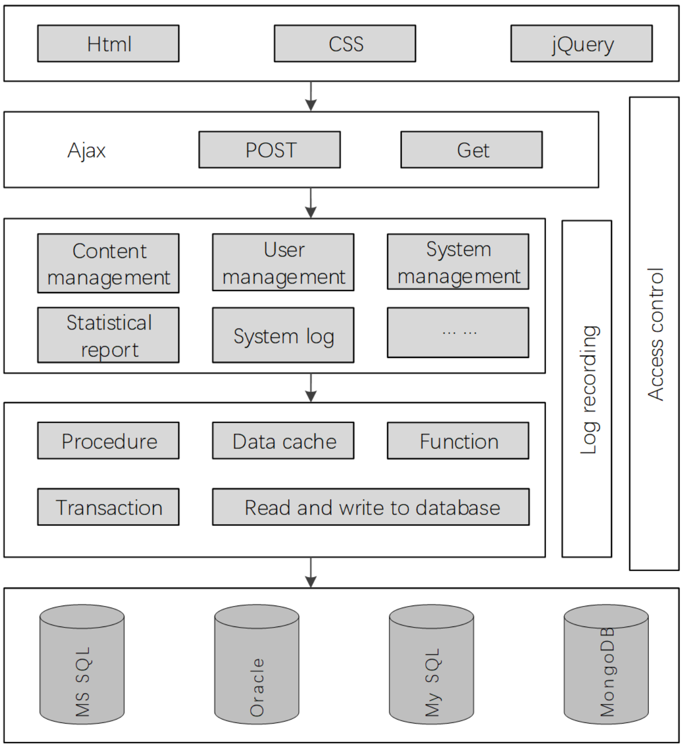

Figure 1. Enterprise WEB Application Storage Model.

In this paper, we will analyze the typical scenarios of enterprise-level web applications, discuss the core requirements and major problems of data persistence, review the existing solutions, and focus on evaluating their performance in terms of reliability, security, scalability, etc. We hope that this paper will help enterprises to choose and design a suitable solution for their business needs. We hope to provide reference and reference for enterprises in selecting and designing data persistence systems suitable for their own business.

2 Data Persistence Requirements in Enterprise Web Applications

In enterprise-level Web applications, data persistence is the basis for ensuring the normal operation of business. With the digital transformation of enterprise business, the amount of data in Web applications has shown exponential growth, and the business scenarios involved have become more and more complex. Data persistence is not only about storing data in persistent media, but also about solving key issues such as high concurrency processing, data consistency, security and scalability. For enterprises, the requirements of data persistence cover reliable data storage, efficient data access, strong data security and flexible scalability.

First of all, data consistency and reliability is one of the most basic and important requirements in enterprise-level Web applications. In many core business systems, such as finance, e-commerce, supply chain management, etc., the accuracy of data directly affects the correctness of business decisions and operations. Any data loss, damage or inconsistency will bring serious economic losses to the enterprise. Therefore, the data persistence system must support transaction processing to ensure the atomicity, consistency, isolation and durability (ACID) of the operation, especially when dealing with multi-step complex business logic, to ensure that all operations are either all successful or rolled back to the initial state in case of error to ensure data integrity.

Second, with the expansion of enterprise business and the growth of the number of users, high concurrency processing capability puts high demands on data persistence in Web applications. Modern Web applications usually need to support a large number of users to access and manipulate data at the same time, which requires the persistence system to have a fast response and high throughput capabilities to ensure system availability without sacrificing data consistency and security. Traditional relational databases are more difficult to achieve horizontal scaling in a single-computer environment, while new persistence solutions such as NoSQL and distributed databases support highly concurrent access through data sharding, distributed storage and replica mechanisms to meet the needs of enterprises to serve users on a global scale.

In addition, data security is one of the core requirements for data persistence in enterprise-level Web applications [3]. Enterprise data not only includes users' personal information, but also involves transaction data, contracts, trade secrets and other sensitive information. Data leakage or unauthorized access can lead to serious legal and economic consequences. Therefore, enterprises must adopt effective security measures, such as data encryption, access control, backup and recovery mechanisms, to ensure the safe storage and transmission of data. Especially when dealing with distributed systems and cloud computing environments, data encryption and transport protocols (e.g. SSL/TLS) are more important to prevent man-in-the-middle attacks and data tampering.

For modern web applications, scalability and flexibility are also essential requirements for data persistence. As business changes and data volumes continue to grow, systems must be able to flexibly scale to address new business needs without significantly impacting the existing system architecture. Traditional vertical scaling (boosting the hardware performance of individual servers) can no longer meet the demands of large-scale data storage and processing, so modern enterprises prefer distributed storage solutions with horizontal scaling capabilities. By adding nodes to expand storage capacity and computing power, organizations can more flexibly respond to business peaks and avoid performance bottlenecks.

Data persistence and high availability requirements are also critical. Enterprise Web applications must ensure that data is continuously accessible under all circumstances, whether it is lost due to server downtime, natural disasters, or other unforeseen circumstances, and must have reliable backup and recovery mechanisms in place. High-availability architecture synchronizes data between different geographic locations through data redundancy and multi-copy mechanisms to ensure that in the event of a server failure in one location, the system can be immediately switched to another copy, thus avoiding service interruptions and guaranteeing business continuity.

Finally, with the rise of data compliance requirements, especially the implementation of GDPR, CCPA and other privacy bills, enterprises also need to consider the issue of data compliance and traceability. Organizations must not only ensure that data is stored legally and compliantly, but also that it is saved, archived, or deleted within a specified time frame. The data persistence system must support fine-grained rights management, operation logging, and other functions for auditing and regulatory needs.

In summary, the data persistence requirements in enterprise-level Web applications cover multiple aspects such as data consistency and reliability, concurrent processing capability, data security, scalability, high availability, and compliance. With the increase of business complexity and the development of technology, enterprises need to choose a persistence solution that suits their business characteristics to cope with these diverse needs and ensure the long-term stable operation of the system.

3 Common Problems and Challenges of Data Persistence

In enterprise-level web applications, data persistence is a key component to ensure data availability and system stability. However, in practice, organizations face a series of complex issues and challenges that not only affect the effective management of data, but also may bring significant risks and costs to business operations. The following are common issues and challenges in the data persistence process.

First, data consistency is one of the most significant issues facing data persistence [4]. As system architectures become more distributed, data synchronization and consistency maintenance across multiple nodes becomes increasingly complex. Although many systems adopt distributed consistency protocols (e.g., Paxos and Raft) to ensure data consistency across multiple replicas, the implementation of these protocols may lead to performance bottlenecks or even data inconsistency under high concurrency. Especially when using NoSQL databases, the eventual consistency model improves the availability and scalability of the system, but it also brings the problem that data consistency is difficult to guarantee in some cases.

Second, performance and scalability are important factors to be considered in the process of data persistence. With the dramatic increase in the number of users and the amount of data, the persistence system must have the ability to handle highly concurrent requests. If not chosen properly, it may cause the system to experience delays under high load and affect the user experience. Therefore, when designing a data persistence solution, enterprises need to consider storage architecture, query optimization, and indexing strategies to ensure that the system can operate efficiently in highly concurrent scenarios.

Another prevalent challenge is data security. Security risks during data persistence mainly include data leakage, unauthorized access and data tampering. Enterprises need to protect sensitive information through data encryption, access control, and audit logs. However, while implementing security measures, the performance of the system needs to be balanced, as too many security checks may introduce additional latency. In addition, data backup and recovery mechanisms need to be well-designed to ensure quick recovery in case of data corruption or loss.

Complexity and management is another challenge that organizations face during data persistence. As technology evolves and data architectures diversify, organizations often need to manage multiple different data storage solutions (e.g., relational databases, NoSQL databases, object stores, etc.). This diversity can lead to increased complexity in system architecture and higher O&M costs. In particular, integration and data migration between different data sources often requires significant time and resources. The lack of a unified management platform may lead to data silos, affecting the efficiency of data utilization.

In addition, the issue of data compliance is also becoming more prominent. As countries tighten data privacy regulations (e.g., GDPR and CCPA), organizations need to ensure that data is stored and processed in compliance with the regulations. This not only requires enterprises to achieve legal compliance in data persistence, but also to ensure that users' privacy rights are respected. The establishment of data auditing and compliance checking mechanisms will undoubtedly bring additional complexity and cost to data management.

Finally, technology selection and implementation are also challenges that enterprises must face in the process of data persistence. Among the many data persistence technologies, enterprises need to make reasonable choices based on their business needs, technology stack and budget. Each technology option has its advantages and disadvantages, and enterprises must consider factors such as future scalability, maintenance costs, and community support when evaluating them. Mistakes in technology selection may lead to subsequent system architecture reconfiguration, increasing the enterprise's development and operation and maintenance costs.

In summary, enterprises face multiple challenges in data consistency, performance and scalability, data security, complexity and management, compliance, and technology selection and implementation when conducting data persistence. Successfully addressing these challenges requires enterprises to carry out comprehensive planning and design from the strategic level, and choose the appropriate persistence solution in combination with actual business needs to ensure the long-term reliability and availability of data.

4 Data Persistence Solutions and Key Technologies

In enterprise-level Web applications, data persistence is one of the core technologies to ensure the efficient operation of business systems. With the expansion of application scale and the increase of data complexity, the existing data persistence solutions have gradually diversified, and different technologies and architectures can meet the different needs of enterprises for reliability, performance and scalability. Existing solutions mainly include relational databases, NoSQL databases, NewSQL databases, object storage and distributed file systems. These solutions adopt key technologies such as transaction management, distributed consistency, data replication, encryption and access control in the specific implementation of data persistence, providing solid technical support for enterprises.

First of all, relational database (RDBMS) is the most traditional and widely used persistence solution, especially in finance, banking, telecommunication and other industries that require high data consistency and complex transaction processing to dominate. Relational databases use Structured Query Language (SQL) to manage data, following the ACID characteristics (Atomicity, Consistency, Isolation, and Durability) to ensure the accuracy and reliability of the data. open-source databases such as MySQL, PostgreSQL, and commercial databases such as Oracle and SQL Server are widely used RDBMS solutions. They are widely used RDBMS solutions. They are able to optimize data access efficiency through transaction support and rich indexing mechanisms, and especially excel in scenarios with data integrity constraints and complex queries. However, with the growth of data volume and concurrency, relational databases have limited vertical scalability, which makes it difficult to cope with the high concurrency requirements of some large-scale distributed systems.

Facing the demands of big data and unstructured data processing, NoSQL databases are gradually becoming an important choice for data persistence in enterprise-level applications. Unlike relational databases, NoSQL databases use a flexible storage model that can handle both semi-structured and unstructured data, and support a variety of data formats such as key-value, document, column family, and graph. Common NoSQL solutions such as MongoDB, Cassandra, and Redis specialize in handling document-based data, highly scalable columnar storage, and caching, respectively [5, 6]. NoSQL databases are highly scalable horizontally through a distributed architecture and can handle massive concurrent requests and large-scale data storage. They use an eventual consistency model that allows for transient data inconsistencies in highly concurrent scenarios in exchange for high system availability and performance. This architecture is well suited for highly dynamic and concurrent application scenarios such as social networking, real-time data analytics, and the Internet of Things.

However, NewSQL databases are a compromise for organizations seeking to combine the transactional nature of relational databases with the high scalability of NoSQL, which can achieve NoSQL-like horizontal scalability while maintaining ACID characteristics through distributed transaction management, as demonstrated by Google Spanner, CockroachDB, and others. NewSQL databases ensure data consistency and high availability in distributed environments through innovative transaction consistency protocols. They are particularly well suited for applications that require worldwide transaction processing and data synchronization, such as financial systems and global supply chain management. The most important feature of NewSQL databases is that they retain support for the SQL query language, enabling organizations to smoothly migrate existing RDBMS systems to a distributed architecture.

Object storage is becoming the mainstream choice for enterprise-class [7] Web applications when dealing with unstructured data. Object storage can manage large-scale files, images, videos, and other data, and is accessible via URL. Unlike traditional file systems, object storage can easily cope with the storage needs of massive amounts of data and has strong scalability. Amazon S3 and Azure Blob Storage are two common object storage solutions that ensure high availability and reliability of data through redundant backups and multi-region disaster recovery mechanisms. The low cost and high efficiency of object storage make it ideal for media platforms, content delivery networks, and data archiving.

Meanwhile, distributed file systems are also widely used in big data scenarios. Distributed file systems provide highly available storage solutions by storing data in slices on multiple nodes. Hadoop HDFS and Ceph are common distributed file systems, especially suitable for application scenarios that require large-scale file processing and complex data analysis. The HDFS storage Architecture model is shown in Figure 2. HDFS is often used in conjunction with big data analytics frameworks such as Hadoop to provide efficient file storage and management capabilities in distributed computing environments. HDFS is often used in conjunction with big data analytics frameworks such as Hadoop to provide efficient file storage and management in a distributed computing environment.

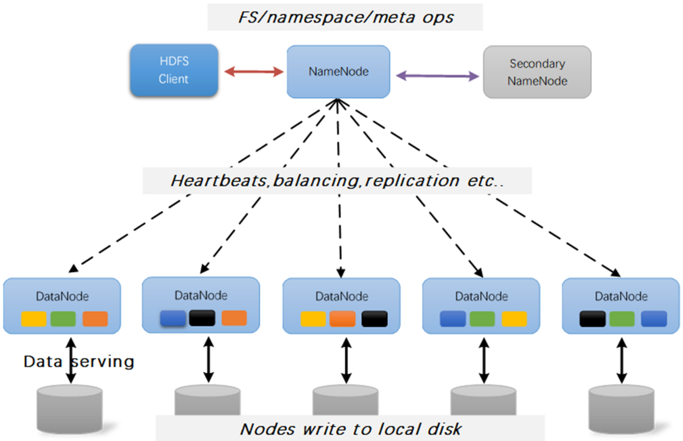

Figure 2. HDFS Storage Architecture Model.

In the process of realizing data persistence, several key technologies provide technical support for these solutions. The first is transaction management and ACID characteristics, which is the core technology in relational databases and NewSQL databases to ensure the consistency and atomicity of data operations. Distributed consistency protocols in distributed systems, such as Paxos and Raft, are used to ensure data consistency across multiple nodes, especially in scenarios with high availability and fault tolerance requirements.

In addition, data replication and high availability technologies improve the disaster tolerance of a system by storing copies of data on multiple nodes to prevent a single point of failure. Whether it is master-slave or multi-master replication, this technology plays an important role in improving system availability. In addition, indexing and query optimization technology effectively improves the response speed of the system by optimizing the query performance when the data volume increases.

Data encryption and access control ensure the security of enterprise data. Whether encrypting data at rest (encryption at storage) or in transit (encryption via SSL/TLS), these technologies prevent data leakage and tampering. Role-Based Access Control (RBAC) further safeguards access to data, ensuring that only authorized users can access sensitive data.

The above analysis shows that data persistence for enterprise-level Web applications is not just a process of storing and retrieving data, but relies on a series of key technologies and architectural designs to ensure that the system can achieve efficient, reliable, and secure data management in the face of complex business scenarios. Different application scenarios determine different persistence solutions. Enterprises should reasonably choose a suitable persistence solution based on their own business needs, data types and scalability requirements to ensure long-term data availability and continuous business development.

5 Future Trends and Technology Outlook

With the rapid development of information technology, data persistence for enterprise-level Web applications is facing unprecedented opportunities and challenges. In the future, the field of data persistence will continue to evolve to adapt to changing business requirements and technical environments. The following trends will have a profound impact on data persistence for enterprise Web applications.

First, the convergence of cloud and edge computing will reshape the architecture of data persistence. Cloud computing provides strong elasticity and scalability, enabling organizations to flexibly manage and store data across geographic locations. Edge computing, on the other hand, reduces latency and bandwidth pressures by pushing compute and storage capabilities to the edge of data generation. In the future, organizations will increasingly adopt a hybrid cloud and edge computing architecture for real-time data processing and analytics. This architecture not only improves data access speed, but also enables more efficient resource utilization while ensuring data security [8].

Second, the application of intelligent technology will enhance the automation and intelligence of the data persistence system. The introduction of artificial intelligence and machine learning will make data storage and management more intelligent. For example, through predictive analytics, the system can dynamically adjust the storage strategy based on historical data and access patterns to optimize data layout and query performance [9]. Meanwhile, automated operation and maintenance tools can help enterprises monitor the health status of data persistence systems in real time and identify potential problems in advance, thus reducing operation and maintenance costs and the need for manual intervention.

With increasingly stringent data privacy and security regulations, data security and compliance management will become the core focus of data persistence. The persistence system of the future will not only have to meet basic security requirements, but also need to be equipped with more robust compliance features, such as automated compliance audits, real-time data privacy protection and access control policies. These features will ensure that organizations can quickly respond to legal and regulatory requirements when handling sensitive data and reduce legal risks arising from compliance issues.

In addition, the rise of serverless architectures will further drive innovation in data persistence. The serverless computing model allows organizations to use computing resources only when they are needed, which greatly improves the efficiency of resource utilization. In terms of data persistence, serverless architectures allow developers to focus on business logic without having to concern themselves with the management of the underlying infrastructure. This trend will enable enterprises to achieve faster development and deployment of data persistence solutions while reducing costs [10].

Finally, the emergence of new storage technologies will bring more options for data persistence. With the popularity of flash memory, solid state drives (SSDs) and non-volatile storage (e.g., NVMe), data access speeds and storage efficiencies will increase dramatically [11]. At the same time, the continued development of distributed storage technologies and big data processing frameworks will enable enterprises to process and analyze massive amounts of data more flexibly. The integration of these new technologies will promote the development of data persistence systems in the direction of higher performance and lower latency to meet the growing business needs.

In summary, data persistence for enterprise Web applications will usher in richer technology choices and innovation opportunities in the future. By taking full advantage of the combination of cloud computing and edge computing, the introduction of intelligent technologies, the emphasis on security compliance, the promotion of serverless architecture, and the development of new storage technologies, enterprises will be able to build more efficient, secure and flexible data persistence solutions to gain a sustainable advantage in the fierce market competition. In the face of these trends, enterprises need to proactively explore and adapt to new data persistence technologies to ensure the success of their business in the future digital transformation.

6 Conclusion

In this paper, we explored data persistence approaches for enterprise-level web applications and the challenges they face. With the deepening of enterprise digital transformation, effective management and persistence of data become more and more important. We analyzed the core requirements of data persistence, including reliability, security, and scalability, and discussed the applicable scenarios and advantages of different data persistence technologies. Traditional relational databases have mature solutions for handling structured data, but their performance and scalability may be limited when facing large-scale data and high concurrency scenarios. Comparatively speaking, NoSQL databases and distributed databases provide higher flexibility and scalability for application scenarios dealing with unstructured or semi-structured data. In addition, object storage, as an emerging data persistence method, can store massive amounts of data at a lower cost, making it suitable for needs such as big data analysis and data backup. However, when choosing the right data persistence solution, enterprises must comprehensively consider their business needs, technical architecture and development goals. With the continuous evolution of technology, emerging data management technologies are also emerging, and enterprises need to keep an eye on the latest technologies in order to adjust and optimize their data management strategies in a timely manner.

In summary, data persistence for enterprise Web applications is not just a technical issue, but also a strategic decision. By choosing the right persistence solution, enterprises can ensure data security and reliability while improving data access efficiency and system performance, providing a solid foundation for sustainable business development.

Conflicts of Interest

The authors declare no conflicts of interest.

Funding

This work was supported by Scientific Research Fund Project of Shaanxi Railway Institute (2014-14) and the Xi'an Science and Technology Plan Project under Grant 22GXFW0023.

References

[1] Guyo, E. D., & Hartmann, T. (2024). Evaluating the efficiency and performance of data persistent systems in managing building and environmental Data: A comparative study. *Advanced Engineering Informatics*, 62, 102582. [CrossRef]

[2] Jambor, D. (2023). *DevOps for databases: A practical guide to applying DevOps best practices to data-persistent technologies*. Packt Publishing.

[3] Wang, X., Hu, X., Fan, W., & Wang, R. (2023). Efficient data persistence and data division for distributed computing in cloud data center networks. *The Journal of Supercomputing*, 79(14), 16300-16327. [CrossRef]

[4] Peng, X., Li, J., & Ren, Y. (2023). Design of Data Persistence for Network Resources Recommendation System Based on Hibernate Architecture. *Procedia Computer Science*, 228, 1143-1151. [CrossRef]

[5] Zdepski, C., Bini, A., & Matos, S. (2020). PDDM: A Database Design Method for Polyglot Persistence. *American Academic Scientific Research Journal for Engineering, Technology, and Sciences*, 71(1), 136-152.

[6] Al-Obeidat, F., Bani-Hani, A., Adedugbe, O., Majdalawieh, M., & Benkhelifa, E. (2021). A microservices persistence technique for cloud-based online social data analysis. *Cluster Computing*, 24(3), 2341-2353.

[7] Rajagopal, K., & Narayanan, K. (2019). Distinctive advancements in Bigdata persistence and processing. *International Journal of Engineering and Advanced Technology*, 9(1), 1442-1444. [CrossRef]

[8] Li, S., Jian, J., Poopal, R. K., Chen, X., He, Y., Xu, H., ... & Ren, Z. (2022). Mathematical modeling in behavior responses: the tendency-prediction based on a persistence model on real-time data. *Ecological Modelling*, 464, 109836. [CrossRef]

[9] Heller, M. (2020, June 22). *Redis 6: A high-speed database, cache, and message broker*. InfoWorld.

https://www.infoworld.com/article/2258622/redis-6-a-high-speed-database-cache-and-message-broker.html

[10] Roy-Hubara, N., Shoval, P., & Sturm, A. (2022). Selecting databases for Polyglot Persistence applications. *Data & Knowledge Engineering*, 137, 101950. [CrossRef]

[11] Wu, S., Zhou, F., Gao, X., Jin, H., & Ren, J. (2019). Dual-page checkpointing: An architectural approach to efficient data persistence for in-memory applications. *ACM Transactions on Architecture and Code Optimization (TACO)*, 15(4), 1-27. [CrossRef]

Zhiwei Xu received a master's degree in computer technology engineering from Xidian University, No. 2 South Taibai Road, Xi'an, CN, in 2014. (Email: yiqingyi@163.com)

Ming Lei received Ph.D. degree in Computer Science & Technology from Xian Jiaotong University, Xi'an, China, in 2019. He is currently a lecturer with the School of Computer Science of Shaanxi Normal University. His research interests include wireless networks, and intelligent optimization. (Email: lei1987920@163.com)