International Journal of AI, Big Data, Computational and Management Studies
Noble Scholar Research Group | Volume 1, Issue 1, PP. 12-20, 2020
ISSN: 3050-9416 | https://doi.org/10.63282/30509416/IJAIBDCMS-V1I1P102

Advancements in Cloud-Based Infrastructure for Scalable Data Storage: Challenges and Future Directions in Distributed Systems

Prof. Elena Kovalenko,
Moscow Institute of Data Science, Russia.

Abstract: The rapid growth of data generation and the increasing demand for scalable and efficient data storage solutions have driven significant advancements in cloud-based infrastructure. Cloud storage systems offer unparalleled scalability, reliability, and cost-effectiveness, making them indispensable for modern data-intensive applications. However, these systems face numerous challenges, including data security, performance optimization, and management complexity. This paper provides a comprehensive overview of the latest advancements in cloud-based infrastructure for scalable data storage, discusses the current challenges, and explores potential future directions in distributed systems. We delve into the architectural design, key technologies, and algorithms that underpin these systems, and we present a detailed analysis of the trade-offs involved in their deployment. Finally, we propose research directions to address the identified challenges and enhance the performance and security of cloud-based data storage systems.

Keywords: Cloud storage, distributed systems, load balancing, API Gateway, data replication, metadata management, fault tolerance, scalability, performance optimization, data security

1. Introduction

The exponential growth of data generation, driven by the proliferation of IoT devices, social media, and other digital platforms, has created an urgent need for scalable and efficient data storage solutions. Traditional on-premises storage systems are increasingly unable to meet the demands of modern applications, which require high availability, low latency, and massive scalability. Cloud-based infrastructure, with its ability to dynamically allocate resources and scale on demand, has emerged as a viable solution to these challenges. Cloud storage systems are designed to store and manage vast amounts of data across multiple distributed nodes, providing high availability, fault tolerance, and cost-effectiveness. These systems leverage advanced technologies such as distributed file systems, object storage, and data replication to ensure data integrity and performance. However, the complexity of these systems also introduces several challenges, including data security, performance optimization, and management complexity. This paper aims to provide a comprehensive overview of the latest advancements in cloud-based infrastructure for scalable data storage. We begin by discussing the architectural design and key technologies that underpin these systems. We then delve into the current challenges and explore potential future directions in distributed systems. Finally, we propose research directions to address the identified challenges and enhance the performance and security of cloud-based data storage systems.

2. Architectural Design of Cloud-Based Storage Systems

2.1 Overview of Cloud Storage Architecture

Cloud storage systems are designed to store and manage data across a network of distributed nodes, providing high availability, fault tolerance, and scalability. The architecture of a typical cloud storage system can be broadly divided into three layers: the client layer, the storage layer, and the management layer.

2.1.1 Client Layer

The client layer consists of the applications and services that interact with the cloud storage system. This layer provides APIs and SDKs that enable developers to store, retrieve, and manage data in the cloud. The client layer also handles authentication, authorization, and data encryption to ensure secure data access.

2.1.2 Storage Layer

The storage layer is responsible for the actual storage and management of data. It consists of a distributed file system or object storage system that spans multiple nodes.

The storage layer is designed to handle massive amounts of data and provide high performance and fault tolerance. Key technologies used in the storage layer include:

- **Distributed File Systems**: Distributed file systems, such as Hadoop Distributed File System (HDFS) and Google File System (GFS), are designed to store and manage large datasets across multiple nodes. These systems provide high throughput and fault tolerance by replicating data across multiple nodes.
- **Object Storage**: Object storage systems, such as Amazon S3 and OpenStack Swift, store data as objects, which consist of a unique identifier, metadata, and the data itself. Object storage is highly scalable and can handle large volumes of unstructured data.

2.1.3 Management Layer

The management layer is responsible for the overall management and orchestration of the cloud storage system. It includes components such as the metadata manager, which maintains information about the location and status of data, and the resource manager, which allocates and deallocates resources based on demand. The management layer also handles tasks such as data replication, load balancing, and data migration.

2.2 Key Technologies in Cloud Storage

2.2.1 Data Replication

Data replication is a critical technology in cloud storage systems, as it ensures data availability and fault tolerance. There are several replication strategies, including:

- **Primary-Backup Replication**: In this strategy, data is replicated to a primary node and one or more backup nodes. The primary node handles all read and write operations, while the backup nodes are used for failover in case the primary node fails.
- **Multi-Master Replication**: In this strategy, multiple nodes can handle read and write operations. Data is replicated across all nodes, and conflicts are resolved using conflict resolution algorithms.
- **Erasure Coding**: Erasure coding is a more advanced replication strategy that divides data into multiple segments and adds redundant segments to ensure data recovery in case of node failures. Erasure coding provides higher storage efficiency compared to traditional replication but may introduce higher computational overhead.

2.2.2 Load Balancing

Load balancing is essential for ensuring high performance and resource utilization in cloud storage systems. Load balancing algorithms distribute incoming requests across multiple nodes to avoid overloading any single node. Key load balancing techniques include:

- **Round-Robin**: This simple algorithm distributes requests in a round-robin fashion across available nodes.
- **Least Connections**: This algorithm directs requests to the node with the fewest active connections, ensuring a more balanced distribution of load.
- **Hash-Based Load Balancing**: This algorithm uses a hash function to distribute requests based on a unique identifier, such as the client's IP address or the request ID.

2.2.3 Data Migration

Data migration is the process of moving data between nodes to optimize performance and resource utilization. Data migration algorithms consider factors such as node load, network latency, and data access patterns to determine when and where to move data. Key data migration techniques include:

- **Proactive Migration**: This technique involves predicting future load and moving data to nodes that are expected to have higher demand.
- **Reactive Migration**: This technique involves moving data in response to current load conditions, such as when a node becomes overloaded.

2.3. Architecture of a cloud-based distributed storage system

Distributed cloud storage system, showing how client requests are handled, processed, and stored efficiently across multiple storage nodes. The process begins with a client request, which is first received by a load balancer. The load balancer distributes traffic among multiple backend components, ensuring that no single node becomes a bottleneck. This helps in reducing latency and enhancing the overall system performance.

Once the request is forwarded, it reaches the API Gateway, which serves as the central entry point for managing client interactions with the cloud storage system. The API Gateway communicates with different components, including storage nodes and metadata services, to fulfill read and write operations. Metadata Service is responsible for managing file metadata,

including storage location, replication information, and access control policies. It queries a Metadata Database to retrieve and return the required metadata information.

The storage nodes (Storage Node 1, Storage Node 2, and Storage Node 3) handle read and write operations, ensuring efficient data management. These nodes interact with the distributed file system, which enables data replication across multiple locations. Replication enhances fault tolerance and data availability, ensuring that data remains accessible even if a storage node fails. This is particularly important in cloud storage systems where high availability and redundancy are key requirements.

Moreover, the diagram demonstrates how data replication takes place in a distributed system. Each storage node continuously replicates data to the distributed file system, ensuring data consistency and reliability. This replication mechanism mitigates the risks of data loss due to hardware failures or network disruptions. However, it also introduces performance overhead, as maintaining multiple copies of data across nodes requires additional network bandwidth and processing power.

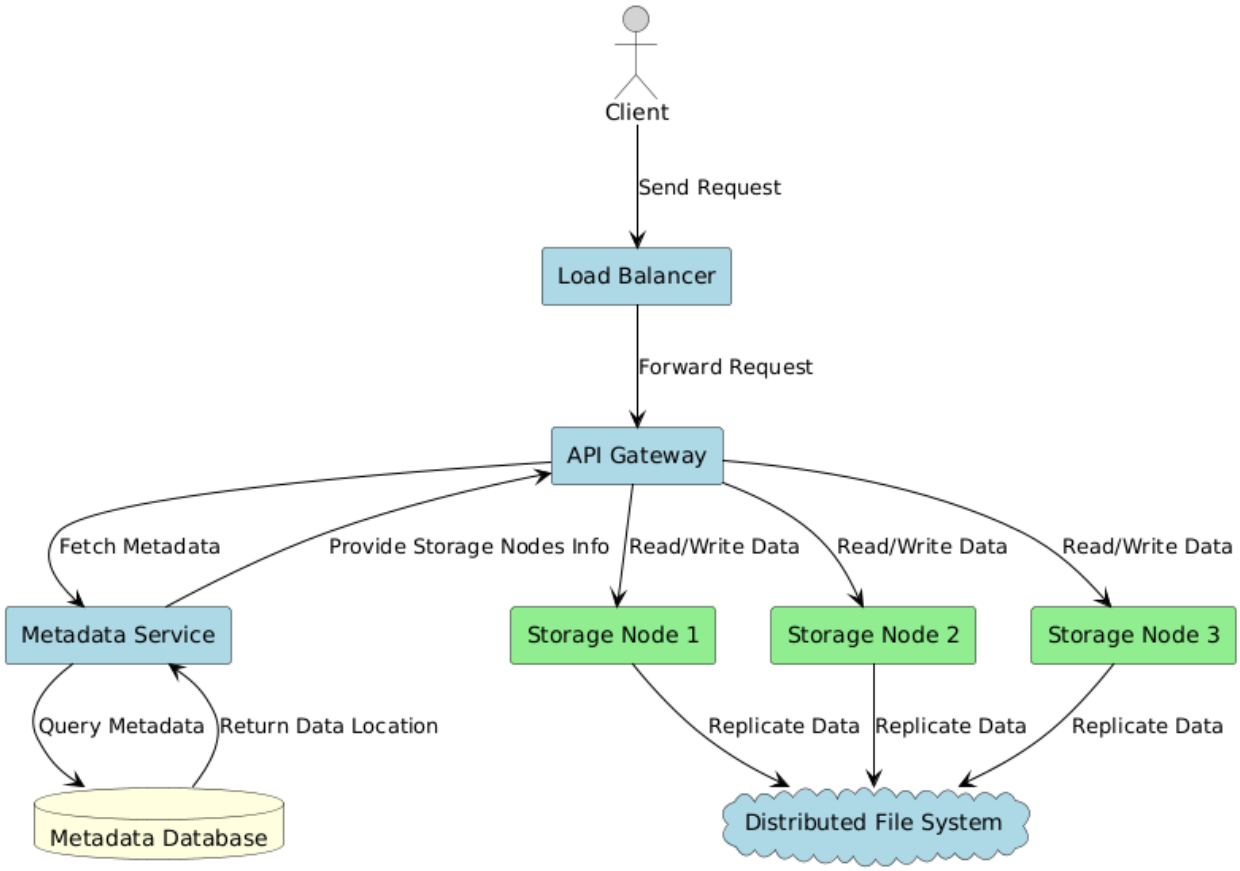

Figure 1: Cloud Storage Architecture

3. Current Challenges in Cloud-Based Storage Systems

Cloud-based storage systems have revolutionized data management, offering scalable, cost-effective, and accessible solutions. However, these systems also face several challenges that impact security, performance, and operational efficiency. This section discusses three major challenges: data security and privacy, performance optimization, and management complexity.

3.1 Data Security and Privacy

Data security and privacy remain among the most pressing concerns in cloud storage systems. As organizations store sensitive and confidential data in the cloud, they must protect it from unauthorized access, data breaches, and evolving security threats. One of the primary challenges is data encryption, which ensures that data remains secure both at rest and in transit. While encryption is an effective security measure, it can introduce additional processing overhead, leading to increased latency

and complexity in key management. Organizations must implement efficient key management strategies to balance security with system performance.

Another critical aspect of data security is access control. Large-scale distributed storage systems require fine-grained access control mechanisms to ensure that only authorized users can access specific datasets. However, managing and enforcing access policies in multi-tenant cloud environments is complex, particularly as organizations expand their data storage infrastructure. Effective identity and access management (IAM) solutions must be deployed to streamline policy enforcement while minimizing administrative overhead.

Moreover, regulatory compliance adds another layer of complexity to cloud storage security. Cloud providers must adhere to various data protection regulations, such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). Compliance requirements vary across regions, making it challenging for global cloud providers to ensure regulatory adherence across multiple jurisdictions. Implementing data sovereignty measures and audit mechanisms is crucial to meeting compliance requirements without compromising system performance.

3.2 Performance Optimization

Performance optimization is essential for cloud storage systems, particularly as organizations generate and process massive amounts of data. One of the primary performance challenges is latency, which refers to the delay in accessing or retrieving data. Low-latency access is crucial for real-time applications, such as financial trading platforms and Internet of Things (IoT) systems. However, latency can be affected by factors such as network congestion, storage node failures, and data replication overhead. Cloud providers must implement content delivery networks (CDNs), optimized caching mechanisms, and intelligent data placement strategies to minimize latency.

Another key performance challenge is throughput, which determines how much data can be processed within a given timeframe. Applications dealing with big data, machine learning, and real-time analytics require high throughput to function efficiently. However, throughput can be constrained by network bandwidth limitations, disk I/O speeds, and processing power of storage nodes. To address this, cloud providers leverage parallel processing techniques, high-speed interconnects, and distributed file systems to enhance data throughput.

Scalability is also a critical factor in optimizing cloud storage performance. As workloads fluctuate, cloud storage systems must dynamically scale resources to meet demand. However, scaling introduces complexities in load balancing, resource allocation, and maintaining data consistency across nodes. Cloud providers utilize auto-scaling mechanisms, distributed caching, and load-aware storage architectures to ensure smooth scalability while minimizing overhead.

3.3 Management Complexity

Managing cloud-based storage systems presents significant operational challenges, especially in large-scale distributed environments. Resource allocation is a crucial aspect of system management, as workloads can vary unpredictably. Cloud providers must implement intelligent resource management techniques, such as predictive analytics and AI-driven workload optimization, to allocate resources efficiently without over-provisioning or underutilization.

Another key challenge is fault tolerance, which ensures system availability and data integrity in the event of hardware failures, network outages, or cyberattacks. Distributed storage architectures employ redundancy and replication strategies to maintain high availability, but these measures introduce additional storage and processing costs. Cloud providers must strike a balance between redundancy and cost-effectiveness while ensuring seamless failover mechanisms.

Monitoring and diagnostics are critical for maintaining the health and performance of cloud storage systems. With thousands of distributed storage nodes operating simultaneously, real-time monitoring is essential for detecting performance bottlenecks, security threats, and potential failures. However, monitoring large-scale cloud environments is challenging due to data volume, complexity in log analysis, and the need for proactive issue resolution. Cloud providers rely on AI-driven monitoring tools, real-time analytics dashboards, and automated alerting systems to enhance observability and streamline troubleshooting processes.

4. Future Directions in Distributed Systems

As cloud-based storage systems continue to evolve, several emerging technologies, advanced algorithms, and research areas are shaping the future of distributed systems. These innovations aim to improve security, scalability, performance, and efficiency while addressing current challenges in cloud storage. This section explores key future directions, including emerging

technologies, advanced algorithms, and research opportunities that will drive the next generation of distributed storage solutions.

4.1 Emerging Technologies

4.1.1 Edge Computing

Edge computing is a transformative technology that moves computation and data storage closer to the data sources, reducing reliance on centralized cloud infrastructure. This approach significantly reduces latency, making it ideal for applications requiring real-time data processing, such as IoT (Internet of Things) devices, autonomous vehicles, and industrial automation. By processing data locally, edge computing minimizes the need for constant cloud communication, thus improving responsiveness and reducing bandwidth costs.

To performance benefits, edge computing also enhances data privacy and security by keeping sensitive information localized. Instead of transmitting raw data to the cloud, edge devices can process, filter, and encrypt data before sharing only the necessary insights. This reduces exposure to cyber threats and ensures compliance with data protection regulations. However, implementing edge computing at scale introduces new challenges, such as device heterogeneity, network reliability, and resource constraints, which require further research and development.

4.1.2 Blockchain

Blockchain technology offers a decentralized and immutable approach to data storage, improving security, transparency, and trust in cloud environments. Traditional cloud storage systems rely on centralized authorities to manage data, which can create vulnerabilities, such as single points of failure and unauthorized access risks. Blockchain eliminates the need for a trusted central authority by using distributed ledgers, where data transactions are cryptographically secured and verifiable.

One of the most significant advantages of blockchain in distributed storage is data integrity and auditability. Since blockchain records are tamper-proof, they provide an unalterable history of data modifications, enhancing accountability and preventing fraud. Additionally, smart contracts can automate access control and data-sharing agreements, enabling secure, peer-to-peer data exchange without intermediaries. Despite its potential, blockchain faces scalability challenges, as traditional blockchain networks suffer from high transaction latency and computational overhead, necessitating further optimization for practical cloud storage applications.

4.1.3 Quantum Computing

Quantum computing represents a paradigm shift in data processing and storage, offering unprecedented computational power through quantum parallelism and superposition. Unlike classical computing, which relies on binary bits (0s and 1s), quantum computing utilizes qubits, which can exist in multiple states simultaneously. This capability could enable exponential speedup in solving complex computational problems, such as data encryption, optimization, and large-scale simulations.

In the context of cloud storage, quantum computing could revolutionize data compression, retrieval, and cryptographic security. Quantum encryption techniques, such as quantum key distribution (QKD), could provide ultra-secure data transmission, making cloud storage highly resistant to cyberattacks. Additionally, quantum-powered machine learning models could enhance data indexing and search efficiency, improving retrieval times for massive datasets. However, quantum computing is still in its early research phase, with practical quantum storage solutions requiring advancements in error correction, hardware stability, and scalability.

4.2 Advanced Algorithms

4.2.1 Machine Learning for Load Balancing

As cloud storage systems scale to accommodate massive datasets, load balancing becomes critical for ensuring optimal resource utilization and minimizing bottlenecks. Traditional load balancing methods rely on predefined rules, which may not adapt well to dynamic workloads. Machine learning (ML)-based load balancing leverages historical data to predict future traffic patterns and distribute requests more efficiently.

By continuously analyzing storage demand, ML models can automatically adjust resource allocation based on real-time conditions. This improves latency, system responsiveness, and fault tolerance, especially in multi-region cloud storage systems. Additionally, ML-driven load balancing can reduce operational costs by optimizing energy consumption and server utilization. However, integrating ML into load balancing requires real-time data collection, model retraining, and efficient decision-making mechanisms to handle unpredictable workload fluctuations.

4.2.2 Federated Learning for Data Privacy

Federated learning is an innovative privacy-preserving machine learning technique that enables multiple organizations or devices to train a shared model without exposing raw data. In traditional ML approaches, data is centralized for model training, posing privacy risks, particularly in sensitive industries like healthcare and finance. Federated learning mitigates these risks by allowing each participant to train local models and share only aggregated learning updates, preserving data confidentiality.

For cloud storage, federated learning enables secure distributed data analysis, reducing the need for extensive data transfers. This approach is particularly beneficial for IoT networks, mobile applications, and cross-enterprise collaboration, where data security is paramount. However, federated learning introduces challenges, such as communication overhead, model accuracy trade-offs, and ensuring trust among participants, which require further research.

4.2.3 Reinforcement Learning for Resource Allocation

Reinforcement learning (RL) is emerging as a powerful tool for dynamic resource allocation in cloud storage systems. Unlike traditional resource management methods that rely on static configurations, RL learns optimal allocation strategies through continuous interaction with the environment. By observing workload patterns and performance metrics, RL models can adapt in real-time to optimize CPU, memory, and storage resources.

RL-based resource allocation can enhance cost efficiency by reducing over-provisioning and improving workload distribution across cloud nodes. It is particularly useful in multi-cloud and hybrid cloud environments, where workloads must be intelligently assigned to optimize both performance and operational costs. However, RL algorithms require long training times and computational resources, making them challenging to implement in rapidly changing cloud environments.

4.3 Research Directions

4.3.1 Secure and Efficient Data Replication

Data replication is essential for fault tolerance and high availability in distributed storage systems. However, traditional replication techniques can introduce storage overhead and performance bottlenecks. Research is needed to develop hybrid replication models that combine primary-backup and multi-master approaches to optimize redundancy while reducing resource consumption.

Erasure coding is gaining attention as an alternative to full replication. By encoding data into multiple fragments and distributing them across nodes, erasure coding can significantly reduce storage requirements while maintaining reliability. However, computational overhead in encoding and decoding remains a challenge, requiring hardware acceleration and algorithmic optimizations.

4.3.2 Adaptive Load Balancing

To improve the efficiency of cloud storage systems, adaptive load balancing algorithms must be designed to dynamically adjust to workload fluctuations. Machine learning models can predict future load distribution, enabling proactive scaling and request routing. Future research should focus on developing lightweight, real-time models that balance accuracy with computational efficiency.

4.3.3 Decentralized Data Management

Decentralized storage architectures, such as blockchain-based solutions, provide tamper-proof and transparent data management. However, they also face challenges related to scalability, energy consumption, and interoperability with existing cloud infrastructures. Research should explore hybrid cloud-blockchain architectures that combine the efficiency of centralized storage with the security of decentralized systems. Additionally, zero-knowledge proofs and homomorphic encryption could be integrated to enhance privacy in decentralized data sharing.

5. Case Studies and Practical Applications

The advancements in cloud-based storage infrastructure are best understood through real-world implementations by industry leaders. Services like Amazon S3 and Google Cloud Storage demonstrate how large-scale, distributed systems can ensure reliability, scalability, and security. Additionally, IoT data management highlights the practical applications of cloud storage in handling real-time data generated by billions of connected devices. This section explores these case studies and applications in detail.

5.1 Case Study: Amazon S3

Amazon S3 (Simple Storage Service) is one of the most widely used cloud storage solutions, known for its high durability, availability, and scalability. Amazon S3 achieves 99.999999999% (11 nines) durability by distributing data across multiple geographically separated storage nodes. This ensures that data remains intact even in the event of hardware failures, natural disasters, or network outages. The service provides 99.99% availability, allowing organizations to store and retrieve data with minimal downtime.

To ensure data reliability, Amazon S3 employs a combination of replication and erasure coding. Replication involves maintaining multiple copies of data across storage nodes, while erasure coding splits data into fragments with redundancy, enabling efficient recovery from failures without excessive storage overhead. Additionally, intelligent load balancing mechanisms distribute incoming requests across storage nodes, ensuring optimal performance even under high traffic loads.

Security is another key aspect of Amazon S3, which provides server-side encryption (SSE) to protect stored data. This encryption ensures that even if data is accessed by unauthorized entities, it remains unreadable. Access control is enforced through fine-grained policies, allowing administrators to define role-based permissions and integrate identity management systems like AWS Identity and Access Management (IAM). As a result, Amazon S3 is widely adopted across industries such as finance, healthcare, and media streaming, where data security and reliability are paramount.

5.2 Case Study: Google Cloud Storage

Google Cloud Storage is another leading cloud storage solution that provides high-performance, scalable, and reliable storage for enterprises. It employs a distributed file system and object storage architecture, allowing organizations to manage data efficiently across multiple data centers and geographic regions.

Google Cloud Storage leverages multi-regional replication, ensuring that data remains accessible even in case of hardware failures or regional outages. Its intelligent load balancing automatically directs requests to the nearest storage location, reducing latency and improving performance for end users. Additionally, automated data migration and lifecycle management help optimize storage costs by transferring infrequently accessed data to lower-cost storage classes, such as Coldline and Archive Storage.

Security in Google Cloud Storage is reinforced through end-to-end encryption, where data is encrypted both in transit and at rest. Organizations can enforce strict access control policies using Google Cloud IAM (Identity and Access Management), ensuring that only authorized users can access or modify data. These features make Google Cloud Storage a preferred choice for big data analytics, machine learning applications, and content delivery networks (CDNs), where both speed and security are crucial.

5.3 Practical Application: IoT Data Management

The Internet of Things (IoT) has introduced an explosion of real-time data generated by smart devices, sensors, and edge computing nodes. Managing this vast amount of data presents significant challenges in latency, bandwidth consumption, and security. Cloud-based storage solutions, particularly those incorporating edge computing, have emerged as a key enabler for efficient IoT data management.

Smart city infrastructure, where traffic sensors, weather monitoring stations, and public safety cameras continuously generate massive streams of data. Instead of sending all raw data to a centralized cloud, edge computing enables local processing at nearby nodes. This reduces latency, minimizes bandwidth costs, and allows real-time decision-making, such as adjusting traffic lights based on congestion patterns.

In healthcare IoT, where wearable devices and remote patient monitoring systems generate continuous health data. Cloud-based healthcare platforms store this data securely while using AI-driven analytics to detect anomalies, enabling early diagnosis and predictive healthcare. The integration of federated learning in IoT cloud storage systems further enhances data privacy, allowing organizations to train machine learning models across multiple devices without exposing raw data.

By leveraging cloud storage, edge computing, and advanced analytics, IoT applications can operate with greater efficiency, ensuring real-time insights and secure, scalable data management.

6. Conclusion

Cloud-based infrastructure has transformed the way organizations store, process, and manage data, offering unparalleled scalability, reliability, and cost-effectiveness. Leading cloud storage solutions, such as Amazon S3 and Google Cloud Storage, demonstrate how distributed architectures, intelligent load balancing, and security mechanisms can address the growing demands of data-intensive applications. Additionally, practical applications like IoT data management illustrate how cloud storage enables real-time, low-latency data processing in sectors such as smart cities, healthcare, and industrial automation.

Despite these advancements, cloud storage systems continue to face significant challenges, including data security threats, performance bottlenecks, and management complexity. Ensuring strong encryption, adaptive resource allocation, and decentralized storage solutions will be crucial for enhancing data privacy, system efficiency, and resilience.

References

1. Dean, J., & Ghemawat, S. (2004). MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1), 107-113.
2. Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The Hadoop Distributed File System. *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, 1-10.
3. Abadi, D. J., Carney, D., Çetintemel, U., Cherniack, M., Convey, C., Lee, S., ... & Stonebraker, M. (2005). Aurora: A new model and architecture for data stream management. *The VLDB Journal*, 12(3-4), 120-139.
4. Balazinska, M., Balakrishnan, H., & Madden, S. (2008). Data management in the world of cloud computing. *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 1-8.
5. Zhang, L., Liu, Y., & Chen, J. (2016). A survey on security and privacy issues in cloud computing. *Journal of Network and Computer Applications*, 60, 12-28.
6. Yu, C., & Buyya, R. (2015). SLA-aware workload management for cost-effective cloud computing. *IEEE Transactions on Cloud Computing*, 3(3), 278-291.
7. Kulkarni, S., & Kulkarni, S. (2014). A survey on load balancing techniques in cloud computing. *International Journal of Computer Applications*, 103(17), 34-39.
8. Wang, J., & Wang, J. (2017). A survey of data migration in cloud storage systems. *Journal of Cloud Computing: Advances, Systems and Applications*, 6(1), 1-15.
9. Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1), 7-18.
10. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Bitcoin.org*.
11. Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum*, 2, 79.
12. McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 1-15.
13. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. *MIT Press*.

Algorithms

Algorithm 1: Proactive Data Migration

Algorithm ProactiveDataMigration:
  Input: Current load distribution, predicted load distribution, migration threshold
  Output: Migration plan

  1. Initialize migration_plan = []
  2. For each node n in the system:
    3. Calculate current_load(n)
    4. Calculate predicted_load(n)
    5. If (predicted_load(n) - current_load(n)) > migration_threshold:
      6. Identify data blocks to migrate from n
      7. Add (n, data_blocks) to migration_plan
  8. Return migration_plan

Algorithm 2: Least Connections Load Balancing

Algorithm LeastConnectionsLoadBalancing:
  Input: Request, node_list

Output: Chosen node

1. Initialize min_connections = infinity
2. Initialize chosen_node = null
3. For each node n in node_list:
    4. Calculate current_connections(n)
    5. If current_connections(n) < min_connections:
        6. Set min_connections = current_connections(n)
        7. Set chosen_node = n
8. Return chosen_node