Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2020, 7(12):258-263**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Comparative Analysis of Big Data Storage Strategies: Hadoop Distributed File System (HDFS) vs. Cloud-based Solutions

Ravi Shankar Koppula

Satsyil Corp, Herndon, VA, USA
Ravikoppula100@gmail.com

Abstract: As the volume and complexity of data continue to grow, selecting an efficient and scalable storage strategy becomes imperative for organizations leveraging big data technologies. This paper presents a comparative analysis of two prominent big data storage strategies: Hadoop Distributed File System (HDFS) and cloud-based solutions. Through an in-depth examination of their architecture, scalability, cost-effectiveness, performance, and security features, this study aims to highlight the strengths and limitations of each approach. By analyzing real-world case studies and performance benchmarks, we provide insights into the optimal scenarios for deploying HDFS versus cloud-based storage. This comparative analysis is intended to guide data architects and IT professionals in making informed decisions about the most suitable storage solutions for their specific data management needs.

Keywords: Big data storage, Hadoop Distributed File System, HDFS, cloud-based storage, data management, scalability, cost-effectiveness, performance, security, comparative analysis.

Introduction

Big data-enabled intelligent applications have significantly changed the way that programmers build data-driven applications and that researchers and practitioners approach various problems in applied sciences and business. The ability to store, manage, manipulate, and analyze massive data now provides numerous values for individuals who collect data about their environments, industries who mine knowledge from their historical data, and social scientists who leverage new and untapped digital data to answer empirical questions. There has been explosive growth in every aspect of the data landscape, including increasing volume, variety, and velocity of data that organizations have embraced through state-of-the-art big data systems like the Hadoop ecosystem. One of the most widely adopted tools in this ecosystem is Hadoop Distributed File System (HDFS), a highly distributed, replicated, and scalable file system designed to run on a set of ordinary hardware.

However, as the volume and value of big data grow, public cloud storage infrastructure has started to gain attention for scalable, effective, and efficient storage of millions of petabytes of data. Indeed, various cloud vendors now offer big data solutions that promise modern businesses sustainable data lakes or data warehouses to store all their raw data and that utilize modern cloud deployment architectures and best practices for scalable, performant, and reliable processing of this data. Given that big data storage is the first cornerstone of scalable, effective, and efficient big data processing and that it is not easy to migrate between different storage solutions due to high data egress costs, it is important for data-center/supercomputing-oriented businesses to understand the trade-offs among different big data storage strategies, e.g., HDFS vs. various cloud storage solutions, and to discuss when a given storage strategy should be chosen, combined, or compromised for optimal performance. We endeavor to answer the root question of whether HDFS is still needed in the cloud era and, if it is, how we could use HDFS most effectively in Amazon Web Services.[1][2]

Overview Of Big Data Storage Strategies

The continuous increase in the volume of data generated due to large deployments of distributed systems motivates the development of scalable storage technologies. These big data storage technologies differ in their interfaces, data models, availability or partition tolerance guarantees, etc., and hence favor different workloads. Since a growing body of research is advocating for a hybrid approach that combines multiple storage services to exploit their best features, it is important to compare their properties to optimize the performance of real big data workloads. Specifically, and because of the popularity of Hadoop Distributed File System (HDFS) and cloud storage services, we want to characterize how they can complement each other. For this purpose, we have built an analytical model that we have experimentally validated and used to compare the performance of Hadoop and Amazon S3/EMR data storage strategies when executing typical MapReduce workloads that read full files. After discussing the results, we conclude on the properties of both storage strategies and characterize common workload conditions that preconize a combination of both data storage services.

Big data storage systems have been designed to scale as the volume of data to be processed witnesses a vast increase. Distinct recognition and features make them suitable for various applications. Interestingly, we showed in a previous work that understanding the unique features of different storage systems and making use of hybrid solutions can optimize MapReduce applications. More recently, Sun et al. showed that updating the same dataset stored in different capacities on different systems (or cloud providers) may serve different performance requirements. This suggests that combining storage systems is not only used for redundancy or fault-tolerant requirements, but also to leverage the unique properties of different systems when data access times or consistency models are required. In this work, we are specifically interested in the combination of HDFS and a commercial cloud-based storage service, named Amazon S3 in the Amazon hosting ecosystem, whose performance is experimentally assessed in this work as part of Amazon EMR.[3]

Hadoop Distributed File System (HDFS)

Hadoop project offers a software library for distributed parallel computation. And this library provides a clean implementation of Google's file system - Hadoop Distributed File System (HDFS). HDFS is designed for distributed and scalable storage. To achieve these features, it divides data into blocks, replicates every piece of data several times, and distributes these data and blocks across multiple machines. HDFS is fault-tolerant, easy to administer, can sustain large-scale data sets, and handle server under-performance. It is a self-healing file system that doesn't require a lot of manual maintenance and is capable of quickly recovering from machine failures. Hadoop is designed to handle cluster machines and network equipment failures that will occur in large-scale deployments, and it utilizes replication and a MapReduce work placement strategy for data durability. It's clear that HDFS has similar characteristics and will be a viable option to store and handle large-scale data sets.

In this study, we have multiple goals with respect to HDFS. First, we want to perform many I/O and storage operations with large-scale data sets to observe the throughput and latency of HDFS and to investigate performance scalability with respect to node counts and per-node data capacity. Second, we want to walk through different replication and data placement strategies to understand the impact on HDFS functions and usage patterns. Third, we want to compare HDFS storage overhead with traditional file systems. Fourth, we want to compare HDFS performance and storage overhead with third-party cloud storage service providers that host large-scale data. Based on our empirical observations and analysis, we hope to provide some guidelines for HDFS performance and resource budget determination for achieving good read and write performance with large-scale data sets. Moreover, investigation on data center balance and scaling behaviors of large-scale storage systems can lead to generic disk storage design principles.

Architecture and Components

Hadoop Distributed File System (HDFS) - Architecture and Components

HDFS is inspired by Google File System (GFS) and the functionalities of HDFS are competitive with respect to the description of Google File System. To take into account the specifics of data processing with MapReduce, HDFS gives priority to aspects of data streaming rather than the speed of individual processing operations. The system was formally described in a 2003 scientific article released by Google. The GFS specification focuses on the software layer between applications and storage, which enables very high-speed transmission of data

between nodes and their reliability on cheaper and less reliable hardware. In contrast to standard NFS, GFS is implemented in a single logical node providing persistent data stored on multiple physical storage devices.
Google's GFS paper in 2003 had a significant impact on HDFS design and use cases of Hadoop, as described in various scientific papers and discussed in agreement. Thanks to the predetermined objectives of distributed data processing with Hadoop, HDFS maintains the following aspects from GFS: the block storage requirements, the data replication system, and Azure's block storage management. The main idea of HDFS architecture is simple. It provides very fast reads and writes by streaming the data from the client to the system, while not requiring support for operations such as check-out, updates, and random file access.[1][2]

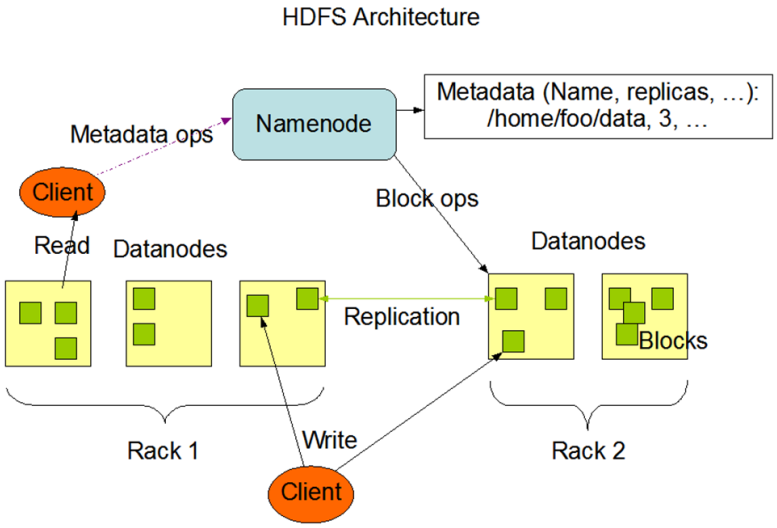


Fig. *1 [7]*

Advantages and Limitations

Advantages and Limitations: HDFS offers the following advantages: 1. Batch data processing, a unique ability to use Hadoop's MapReduce, especially for complex analytical tasks. 2. Data redundancy. Data saved in HDFS is available regardless of hardware failures (redundancy is provided by generating the specified number of copies). 3. Horizontal scalability. HDFS increases the storage capacity of a computing system in which it is used using architecture that includes data skidding. 4. Compatibility with popular analytical systems. 5. High speed - HDFS supports high-speed data analysis. There are other file systems designed for processing large datasets based on DHT technology (e.g. ADAM from DAFS, Kinetic by Seagate), but HDFS has broader functionality.
Nevertheless, HDFS has a number of limitations that push users to modify the functionality of the entire Hadoop system or use cloud-based storage systems. Some of the limitations of Hadoop (as a result of the impact of HDFS, or rather, the initial view of Hadoop on the organization of data storage) are well known and have been addressed. However, over time, the generated datasets have become even greater, so the limitations of HDFS are still sharp and a number of current solutions and mechanisms for bypassing these limitations are directly dependent on a certain minimum volume of the researchers' dataset. Little attention has been paid to the creation of easy-to-use and easy-to-organize file systems for integrating source data (i.e. small preliminary datasets).

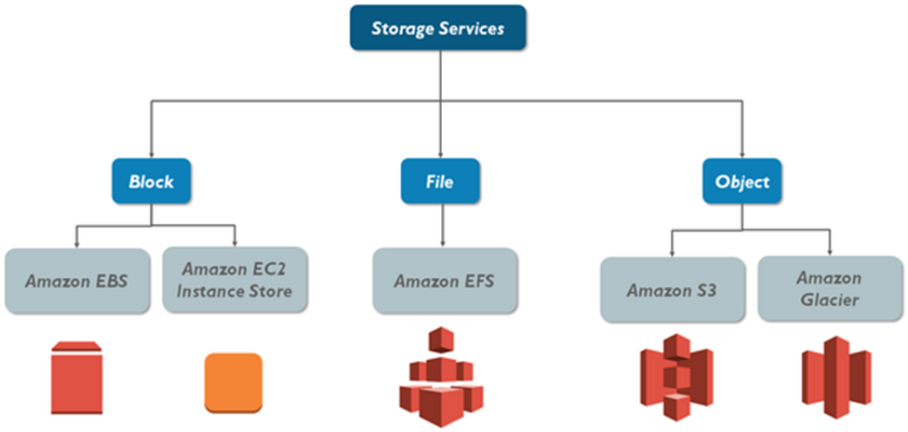
Cloud-Based Solutions

Cloud-based solutions include file systems, which offer their users high scalability, easy access to data at high performance, and single-image access to the data that is suitable for HPC-style data access patterns, usually with high bandwidth and sometimes with high latency. The cloud-based solutions include Amazon Simple Storage Service (S3) and Google Cloud Storage (GCS) and offer a good alternative, very competitive to Hadoop Distributed File System (HDFS). In this chapter, we provide a comparative analysis of these three cloud-based storage systems, present results of the performance of the word count application executed across Hadoop installations, which are running in four different environments [Cluster with different numbers of machines; Hadoop Amazon Elastic MapReduce (EMR) using Amazon Web Services (AWS); Hadoop Google Compute Engine (GCE) using Google Cloud Platform; Hadoop Distributed Cluster (HDC) running on the university cluster]. The performance analysis is based on different numbers of input files and lengths. Moreover, the

performance of the word count application called through the separate application and map-reduce mode are compared. Results of the performance indicate that execution of an application on a small number of processors allows to check correctness of the application operation and establish the workload and the application scalability.[3][5]

Types of Cloud Storage Services

Some of the key families of cloud storage services are the following: 1) Object Storage Services: Object storage services organize data into objects (files), where each object has a unique identifier, and there are various access levels - to read, write, or delete those storage elements. Examples of object storage services are Amazon S3, Microsoft Azure Blob Storage, Google Cloud Storage, and others. 2) Block Storage Services: With block storage services, users can manage storage volumes, allocate and assign the amount of storage that they need and have been granted by their configuration management software, format and mount those volumes to their virtual (or physical) devices, and finally and mainly use all attached volumes as locally installed hard drives of their virtual machines. Examples of block storage services are AWS Elastic Block Store, Azure Disk Storage, Google Persistent Disk, Rackspace Block Storage Volume, IBM Cloud Block Storage, and others. 3) File Storage Services: Cloud file storage services provide organized file systems and allow users to store, access, share, and synchronize files on a server. Examples of file storage services are AWS Elastic File System, Azure File Storage, Google Filestore, and others.[4]



Key Providers and Offerings

In this section, we discuss the key cloud-based economic solutions focusing on the storage services. Using big data in the cloud requires tasks such as encoding, ingestion, and storage. Below, we discuss the major cloud-based storage providers such as Amazon Web Services, Microsoft Azure, Google Cloud Services, and others. Note that our choice is limited to the major cloud infrastructure providers. This is because there are many smaller providers and installations that are built on open-source software, OpenStack for instance. Similarly, services from the giant providers are built around such packages. An exception to the open-source software is Google BigQuery, which is recognized to be built in its entirety by Google itself.

Large cloud infrastructure providers offer a wide range of services. They include computation services like virtual machines, solutions to control how workloads are distributed among the virtual machines, storage services, networking services, database services, but also artificial intelligence workloads such as machine learning, big data analytics, deployments of blockchain applications, etc. In this work, we are only interested in storage services for big data. Nonetheless, it is important to consider that many of the advanced processing services such as deep learning need to work with large data, potentially stored on cloud infrastructures.

Comparative Analysis Framework

The advent of the Big Data era has advocated the necessity of the development of new storage systems that can overcome the limitations of traditional storage systems. In this study, we have surveyed some of the Big Data storage systems which are based on distributed cloud storage infrastructures and compared Hadoop Distributed File System (HDFS) and cloud-based storage systems for adoption of the most suitable one for data storage needs by defining a novel, comprehensive and revealing criteria set. The findings revealed that HDFS is more

applicable due to its availability of a scalable storage system inspired from the original Google File System (GFS) for managing big volumes of data. However, due to its limitations, cloud-based storage systems are selected for data storage needs for some criteria. This study will support the efforts to evaluate HDFS and cloud-based storage systems not only for researchers and academicians but also for entrepreneurs and industry professionals who are engaged in developing data storage applications.

This study mainly aims to guide users in the selection of distributed storage systems with an extensive approach. For this purpose, we defined a comprehensive and illustrative criteria set, enumerated the attributes of the proven Apache HDFS and described several examples of the commonly preferred cloud storage services/applications such as Amazon S3, Windows Azure Storage, Google Cloud Storage and Apache HBase. Our analysis presents that HDFS is a more reasonable alternative for data storage needs, since several cloud-based storage properties may not be compatible with the requirements of the users. The research also sustains an application field by providing fruitful contributions to researchers which target performance efficiency.[1][2][3][5]

Performance Metrics and Evaluation Criteria

The growth of data generated annually at an exponential speed is making the design and management of distributed file systems an ever-challenging task. Various distributed file storage techniques, suitable for the flow of Big Data, have been proposed in academia and implemented in the industry to handle these challenges.

This manuscript specifically conducts a comparative evaluation of the prevalent Hadoop Distributed File System, serving as the Big Data storage solution with characteristics like high throughput and fault tolerance, and two commercial cloud-based solutions—Microsoft Azure Blob and the WebHDFS service for Amazon S3. Our investigation is based on the premise that the direct adaptation of Hadoop Distributed File System for non-MapReduce based data handling tasks could offer significantly improved experimental completion times for a wide variety of data handling tasks suitable for the Map—Shuffle—Process, the ETL—Transform and Load, and the Extract—Analyze Big-Data pattern applications.

Our empirical experiments offer first-hand insights to cloud engineers on how the most popular of the scalable Big Data ecosystem file systems can perform outside its original paradigm, and on which of three currently prevalent scalable cloud storage services are best suitable for a suite of currently prevalent distributed file handling tasks. Some of the performance metrics included in our evaluations and the derived comparative analysis to rank candidates are presented, including a practical example from the financial services sector.

Case Studies and Real-World Applications

Question: What are the successful applications and use cases of HDFS and Clouds? Which of them can benefit from the deployment of the unified virtual Big Data storage infrastructure? HDFS proved to be a solid solution to manage very large amounts of data in the area of (but not limited to) enterprise data analysis, social networks and media, clusters, supercomputers, and traffic control systems. The particular HDFS distributions are known and revised, and they include Apache Hadoop 2.7.7+ ones. Hadoop Ecosystem is known, wide, and very powerful. It includes Hadoop MapReduce to process distributed data and deliver the results into a Hadoop cluster, Flume, Sqoop, Apache Spark, Hadoop Streaming, the Apache Pig Latin language, the Hive data warehouse, HBase, Tez, ZooKeeper, Oozie, Apache Mahout – a library that helps organize machine learning workflows, and others. Hadoop supports a well-distinguished VMware virtualization technology. The software is versatile, and it may run on different operating systems and Cloud environment configurations. It really pushes the scalability limits. With HDFS you may lose 1/2 massive data amount in comparison to Amazon S3 at the lowest price.

Conclusion and Future Directions

In the context of big data, the development of modern solutions to big data storage problems is critical. Today, it is important to compare existing big data storage systems in order to rank and select the most effective solution. In this study, we compared Hadoop Distributed File System (HDFS) and cloud-based solutions in terms of availability, durability, scalability, and fault tolerance.

After analyzing the storage methods, we have concluded that cloud-based systems have a number of advantages over Hadoop Distributed File System. First, they are easy to use. Second, they allow up to 11 nines of durability (cloud) without any coding complexities. Third, we can use cloud storage with any environment we wish. However, Hadoop is not so user friendly. It does not provide advantages for moving large datasets out of many environments, and it can provide up to 5-7 nines of durability using the -X option in HDFS (Hadoop)—a lower durability rate than most cloud-based solutions.

In general, Hadoop Distributed File System and cloud-based big data storage solutions demonstrate a high level of implementation in all storage matters (availability, durability, scalability, fault tolerance). We also calculated and compared the total cost of ownership for the storage solutions. The cost analysis revealed that, whereas the file system was sustainable in spreading data for fault tolerance due to racks, low bandwidth prevented it from being accessed concurrently.

Overall, our comparisons of these technologies suggest that the choice of the solution can improve big data storage facilities for all parties involved—companies, researchers, and developers. It is better to use cloud instead of, or in addition to, using HDFS for higher durability and availability.[1][2][3][5][6]

References

[1]. Borthakur, D., "The Hadoop Distributed File System: Architecture and Design," Hadoop Project Website, 2007.

[2]. Shvachko, K., Hairong, K., Radia, S., and Chansler, R., "The Hadoop Distributed File System," in 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), Incline Village, NV, USA, 2010, pp. 1-10.

[3]. Amazon Web Services, "Amazon S3: Object Storage Built to Store and Retrieve Any Amount of Data from Anywhere," [Online]. Available: https://aws.amazon.com/s3/.

[4]. Ghemawat, S., Gobioff, H., and Leung, S.-T., "The Google File System," in Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP '03), Bolton Landing, NY, USA, 2003, pp. 29-43.

[5]. Microsoft Azure, "Azure Storage: Cloud Storage Solutions and Services," [Online]. Available: https://azure.microsoft.com/en-us/services/storage/.

[6]. Dean, J. and Ghemawat, S., "MapReduce: Simplified Data Processing on Large Clusters," Communications of the ACM, vol. 51, no. 1, pp. 107-113, 2008.

[7]. Apache Hadoop 3.3.1 – HDFS Architecture," hadoop.apache.org. https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

[8]. AWS, "Amazon Web Services (AWS) - Cloud Computing Services," Amazon Web Services, Inc., 2018. https://aws.amazon.com/