iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** III **Month of publication:** March 2023

DOI: https://doi.org/10.22214/ijraset.2023.49294

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue III Mar 2023- Available at www.ijraset.com

A Literature Review of Big Data and Metadata Management Systems

Prof. Shwetha Ks[1], Pruthvi M Javali[2], Ritvik RC[3], Rohan Shet[4], Nagesh Madhu[5]

[1]*Assistant Professor, Computer Science Department, Dayanand Sagar college of Engineering*
[2, 3, 4, 5]*Computer Science Department, Dayanand Sagar college of Engineering*

Abstract: ***Metadata is essential for the successful operation of data files, as it acts as the underlying data that helps maintain and manage the namespace, permissions, and address of the file data blocks. The performance of metadata services has a direct impact on the overall performance of the file system, as these operations can account for a significant portion, up to 80%, of file system activity. Large-scale distributed file systems (DFS) have emerged to meet the rising demand for data storing. These systems provide users with access to numerous storage units spread across various locations. The widespread use of large-scale DFSs as the foundation for a range of computing systems and applications, particularly in the era of Big Data, highlights the importance of efficient metadata management. This paper seeks to offer an enhanced knowledge of the current status of metadata systems in large-scale DFSs, this paper presents a comprehensive overview of the latest research in the field. DFSs are defined by three aspects: scalability, performance, and availability. Moreover, the study evaluates various challenges and limitations in the current research, providing valuable insight for future studies. Overall, the study attempts to give a complete knowledge of the present scenario of metadata services across large-scale DFSs and to recommend areas for future research and development.***

I. INTRODUCTION

New demands have been facilitated on computing systems by the diversification of big data applications, especially with regards to the scale and performance of storage systems. Big Data in sectors like healthcare, traffic, and finance can reach staggering sizes, with data sets often measured in terabytes, petabytes, or even exabytes. As a result, a significant amount of storage resources are necessary to maintain and retrieve these large quantities of data. Furthermore, many data analysis tasks require quick access to data across multiple storage servers, necessitating fast reading and writing rates from the storage system. This is particularly important in the case of distributed file systems (DFSs), which have become a popular solution for managing Big Data storage due to their ease of use and versatility. DFSs are commonly used in applications such as Big Data, high-performance computing (HPC), and web applications. As a result, effective data structure and management is mandatory, to support these huge data storage and computational requirements. This includes carefully considering hardware optimization to ensure that the storage system can meet the high demands placed on it. By addressing these challenges, storage systems can more effectively support the needs of Big Data applications and meet the growing demand for large-scale data storage and management.

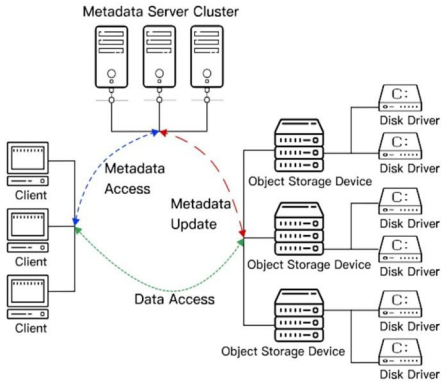


Fig. 1. A typical DFS structure. It includes three parts: MDS, OSD, and interaction with clients.

©IJRASET: All Rights are Reserved

II. CONTRIBUTION

In this study we aim to discuss the topics of big data and the metadata management systems. We discuss the advantages and shortcomings of the same. Our Contribution are:

1) We have surveyed many existing literatures works in this field in detail.

2) We have analyzed the advantages of existing systems in the market and the shortcomings of the systems and presented them in a simplified manner and will try on resolving the issues in the further works.

Index Terms – Metadata, Distributed File System, Big Data, Hashing, Scalability, Availability, Performance.

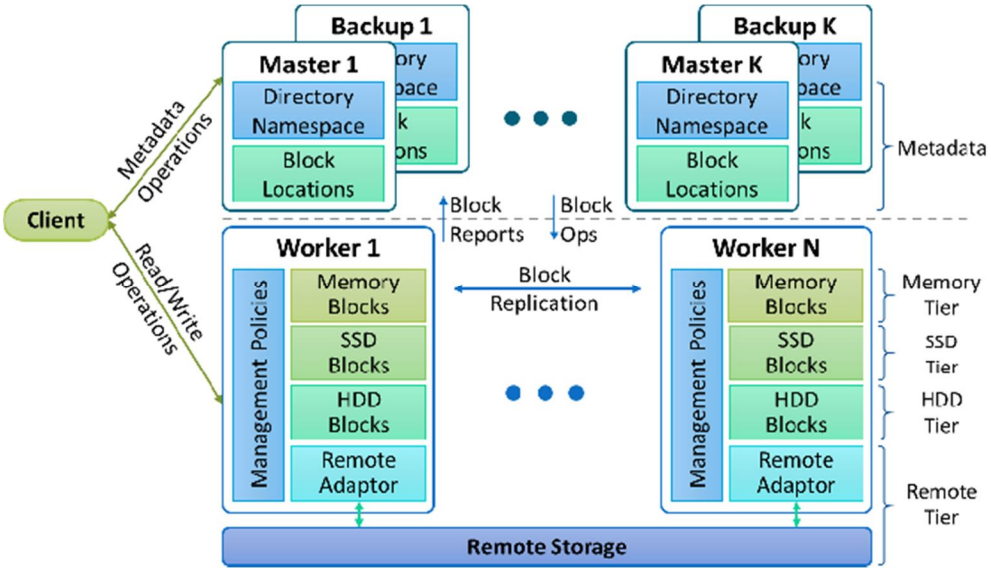
III. LITERATURE SURVEY

1) J. Sun and C. K. Reddy, "Big data analytics for healthcare,"

The authors aim to enhance healthcare systems by using the Big Data Analytics concepts. In Healthcare, a significant amount of data is generated, providing opportunities for cost-saving solutions through the use of data repositories and big data analytics. As the complexities of data continue to evolve, this field holds much potential. The authors explore concepts related to the Four Vs of big data and its impact on healthcare system. Also covered Hadoop-based applications for the health industry. However, they conclude that image data from common imaging modalities such as PET, CT, and MRI in the healthcare sector can present challenges during storage and analysis.

2) E. Kakoulli and H. Herodotou, "OctopusFS: A distributed file system with tiered storage management."

The authors present Octopus File System, a software solution that facilitates scalable and efficient data storage. This system leverages both hard disk drives (HDDs) and solid-state drives (SSDs) along with local memory. The retrieval policy of Octopus FS outperforms HDFS in terms of throughput, since it favors reads from replicates on higher layers. Placement and retrieval of data between nodes and storage racks are handled automatically. Octopus FS provides a flexible storage solution that improves I/O performance and cluster utilization, though other areas remain unaddressed.



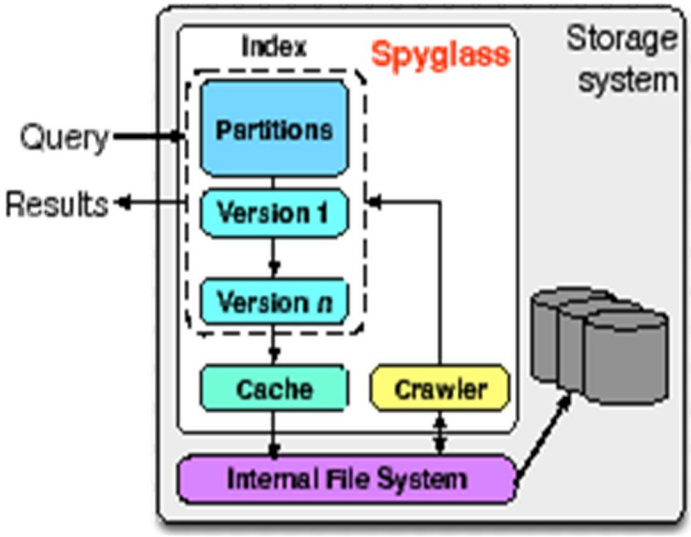
OctopusFS architecture with four storage tiers , two master tiers and two worker tiers.

3) S. A. Weil, K. T. Pollack, S. A. Brandt, and E. L. Miller, "Dynamic metadata management for petabyte-scale file systems."

The authors here describe the Meta Data storage, Writing changes to a log using a two-tiered technique which helps in stable storage. The Metadata distribution and collaborative caching framework is used to prevent flash crowds. The concepts used in achieving this were, Dynamic Subtree Partitioning and parallel coordination of Adaptive Metadata management system to manage hierarchical metadata workloads. Though the simple load balancing algorithm are sufficient for static subtree but for more workload and redistribution we will need more intelligent algorithms and heuristics.

4) *A. W. Leung, M. Shao, T. Bisson, S. Pasupathy, and E. L. Miller, "Spyglass: Fast, scalable metadata search for large-scale storage systems."*

The Spyglass system is a metadata search tool tailored for large-scale storage systems. It incorporates an efficient design, informed by real-world metadata examples, to enable fast and complex searches of file metadata. With the use of a spyglass index tree, which is a hierarchical tree with indexed partitions, searches are made easier. The Spyglass method has been shown to improve retrieval speed by a factor of 8 to 44, while also reducing storage space by 5 to 8 times. Despite its improved search capabilities, Spyglass's indexing imposes additional overhead based on the system's memory capacity and metadata arrangement. Additionally, the search process does not fully guarantee the security of specific files during metadata searches.



Spyglass overview. The Spyglass system is integrated within the storage infrastructure. It works by extracting file metadata through a crawling process, which then gets organized and stored within the index. The index is comprised of various partitions and versions, all efficiently managed by a caching system.

5) *A. Leung, M. Shao, T. Bisson, S. Pasupathy, and E. Miller, "High performance metadata indexing and search in Peta scale data storage systems,"*

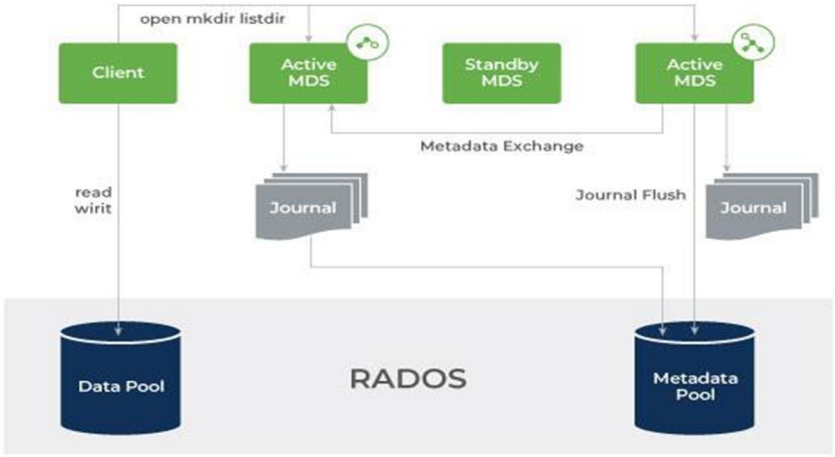
The authors discussed their thoughts on large-scale and high-performance storage, expressing concern over the need for storage systems that can accommodate petabytes of data and billions of files. To address this challenge, they introduced Spyglass. This tool was introduced to search the metadata and to implement indexing system designed to meet the demands of scalable storage systems.

6) *A. B. M. Moniruzzaman, "NewSQL: Towards next-generation scalable RDBMS for online transaction processing (OLTP) for Big Data management."*

The authors describe the system called NewSQL- Which provides and interface for a class of modern relational Database Management System (RDBMS) and NoSQL systems for OLTP. It Can handle read-write workloads for OLTP. Also, makes sure that ACID properties are not compromised. Authors also concluded with some limitations like Benchmarking is not done which is important in load testing, scalability testing and processing time of small and frequent requests must be improved.

7) *S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, "Ceph: A scalable, high-performance distributed file system.".*

The authors proposed a software Ceph which is a scalable and high-performance DFS that works on the dynamically distributed meta data operations, distributed object storage and data distribution with CRASH. This software efficiently coordinates with the metadata separation which increases scalability, reliability and performance. The authors have specified the future work on this software. The checksum and bit-error detection mechanisms can be improved. Also, the MDS Failure recovery is not resilient and POSIX calls are not implemented.



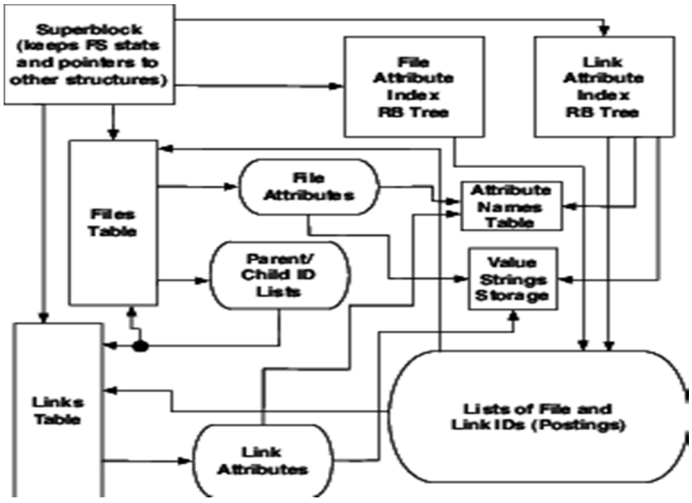
Scale-out Network attached storage (NAT) architecture in the Ceph file system.

8) Sage A. Weil Scott A. Brandt Ethan L. Miller Carlos Maltzahn. "CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data."

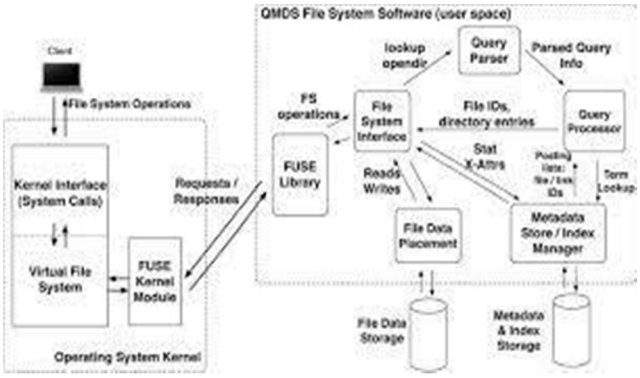
For a better Data distribution scalability, the authors use a Pseudo random mapping function which eliminates the standard need for metadata distribution system. A Cluster map hierarchy Implementation is done. The main concepts focused are Pseudo random data distribution function that is scalable for data objects, CRUSH algorithm and the types of buckets. But data safety concerns should be taken care of due to coincident failures in the system. Improved rigidity in the rule structure of CRUSH algorithm. This relies heavily on a powerful multi-input integer hash algorithm.

9) Sasha Ames, Maya B. Gokhale, Carlos Maltzahn.
"QMDS: A File System Metadata Management Service Supporting a Graph Data Model-based Query Language."

Many directories are incapable of handling complicated data linkages. Many have tried to work around this problem but instead created more inadequacies. In this paper authors have solved it by introducing QMDS file system which is a metadata management service, and it makes use of a graph data model, with properties located at nodes and edges for easier and faster searching. PostgreSQL is used in the model. QMDS completes the write operation 2.4 times faster than the traditional databases. Although there are many benefits in this existing model, it is difficult to achieve scalability for a limited number of files. When evaluated the model performance against the Berkeley DB the data ingestion did not complete even after 36 hours of processing of around 100,000 files.



The QMDS metadata store's structure is designed to maximise attribute matching, neighbour pattern matching, and navigation.



The QMDS prototype software architecture is a single-host file server exporting a FUSE interface that allows clients to the POSIX file system interface to pass Quasar expressions.

10) J. Liu, D. Feng, Y. Hua, B. Peng, and Z. Nie, "Using provenance to efficiently improve metadata searching performance in storage systems,"

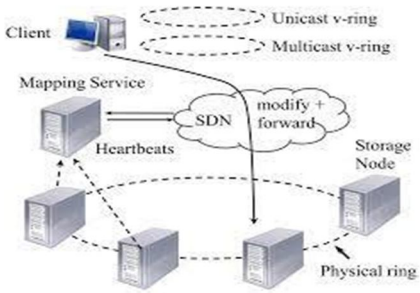
Metadata access in the distributed file system can be increased by Caching and prefetching operations. Existing systems do not take care of the file provenances and correlation between the corresponding files. Therefore, the authors of this paper have come up with a Provenance-based Metadata Prefetching (ProMP) scheme. This model performs with less overhead compared to existing systems in the market. The hit rates of this system are up by up to 49%. Previously, open provenance model (OPM) defined the term provenance, and five relations were derived among the process, data, and Agent. Having many advantages ProMP can be implemented if its memory overhead can be reduced more. Memory overload is better as sparse tree is being used instead of graph-based data and it can be reduced more then it would be a more efficient system.

11) G. A. Gibson et al., "A cost-effective, high-bandwidth storage architecture,"

The authors express their worry about the high cost associated with achieving scalable storage bandwidth in cluster systems. They highlight that this cost can be reduced by distributing data effectively across storage devices and network links with adequate bisection bandwidth. However, Workstation servers, network adapters, and peripheral adapters cost 80% higher as compared to simply purchasing the storage. To overcome this problem, the authors suggest NASD, a prototype solution i.e. Network-Attached Secure Disks. This offers scalable bandwidth at a more affordable cost without negatively impacting the performance of server-based systems.

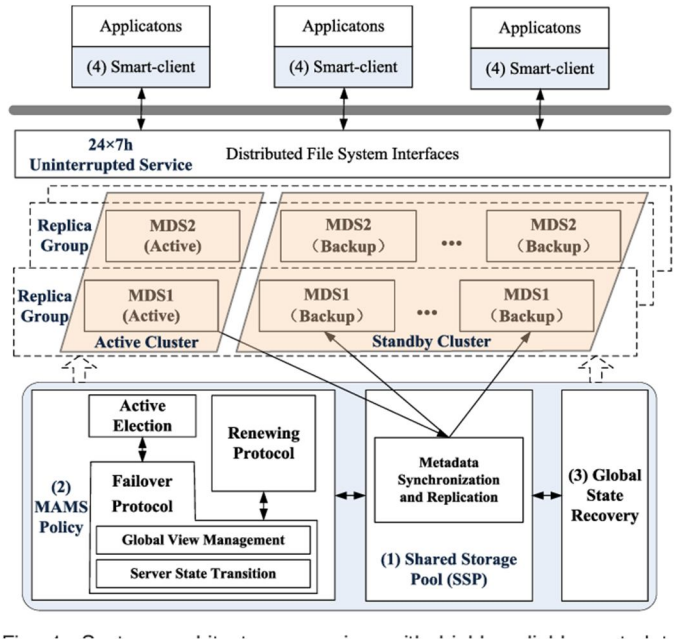
12) I. Kettaneh et al., "The network-integrated storage system,".

The authors introduce a new storage solution known as NICE (network-integrated cluster-efficient) that combines the functions of both storage and networking to deliver a more effective, scalable and dependable distribution system. NICE aims to enhance replication-based storage systems and ensures efficient replication of blocks while reducing network load. The authors conducted experiments that showed substantial improvement in performance and load balancing, resulting in a significant reduction in network load.



The client interfaces with the storage system via two virtual rings in the system architecture (v-rings). Requests are sent via the network to the appropriate storage node. Receiving heartbeat signals from the nodes allows the metadata service to keep track of them and update the mapping information in the forwarding tables.

13) J. Zhou, Y. Chen, W. Wang, S. He, and D. Meng, "A highly reliable metadata service for large-scale distributed file systems,"
Metadata failure in a system is a critical failure as it can affect directly on file and directory operations, therefore making fault tolerance an important factor. To address challenges in large-scale file systems, the authors of this research suggest a highly dependable metadata service that is Multiple Actives Multiple Standbys (MAMS). This is used to perform metadata service recovery. This paper also implements a Clover file system (CFS) and it has also been tested and evaluated. In the proposed solution the Mean-Time-To-Recovery (MTTR) for the highly important metadata services was decreased by around 81%,65%, and 29%. The availability of data is not completely taken care of in this existing system which can be improvised in the future. Also, integration of this along with fault tolerance methods and consolidation approaches, file system reliability may be improved even further.



Overview of the system architecture with extremely dependable metadata service.

14) X. Wu and Z. Shao, "Selfie: Co-locating metadata and data to enable fast virtual block devices,"
In this paper authors have introduced us with a new system called Selfie. This is a virtual disk format which alternates the process of frequently updating the metadata with write operations with writing of the associated metadata to be completed in on atomic transaction. Evaluation of this methods shows that it is 5 times better than existing virtual disks. Maintaining of metadata table, zoning on-image space, writing blocks into zones robustness of the system are some of the important topics discussed in the paper. The system has the significant advantage of not requiring any hardware support. This makes it pretty easy to deploy this system in any other existing system. This system can cause overheads on the block where the metadata updates are stored before it is implemented into the lookup table in any atomic transaction. This could in turn lower the processing speed of the system and therefore should be looked up on.

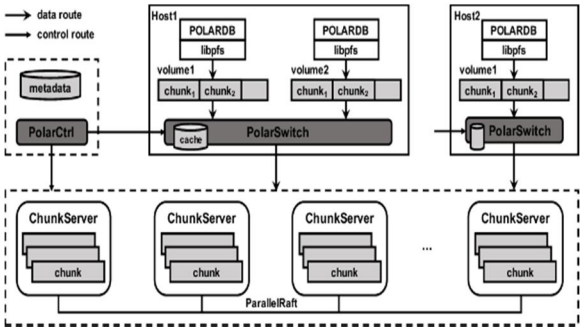
15) Alexander Thomson, Google; Daniel J. Abadi, Yale University "CalvinFS: Consistent WAN Replication and Scalable Metadata Management for Distributed File Systems."
The author presents a unique approach to scalable file systems in comparison to traditional systems, which could store metadata on a single server or using a shared-disk architecture. The proposed solution is called Replicate Scalable File System and uses high-performance distributed database system for metadata management. The system is made up of three parts: a transaction request log, a storage layer, and a scheduling layer. By dividing metadata across multiple machines, Calvin FS requires less memory per machine and can support large-scale clusters for storing billions of files, hundreds of thousands of updates per second, and millions of reads per second with minimal read latencies. Additionally, Calvin FS is highly resilient, able to withstand failures of whole datacenters with just minimal performance consequences and no loss of availability.

16) *L. Pineda-Morales, A. Costan, and G. Antoniu, "Towards multi-site metadata management for geographically distributed cloud workflows,"*

The authors of this paper have come up with ways to improve the efficiency of executing workflows across different cloud centers located in different geographic regions, while minimizing the cost of metadata management. The results of evaluating the design across servers in the EU and US showed that there was a 28% reduction in the time it took to execute parallel and geo-distributed real-world applications. It was also observed that up to 50% for a metadata-intensive synthetic benchmark was saved compared to centralized configurations. In this research, various workflow patterns were studied, and the proposed designs followed architecture and caching design principles. Decentralized strategies were evaluated based on their completion time, and efforts were made to speed up the process. Additionally, five steps were taken to enhance the performance of small file input/output operations, like Calvin FS. This system design can be implemented using cloud technology blocks located in different geographical sites.

17) *W. Cao et al., "PolarFS: An ultra-low latency and failure resilient dis tributed file system for shared storage cloud database."*

In this paper, the authors introduce Polar FS, which is a distributed file system that offers extraordinary performance, reliability, and availability with minimal latency. To accommodate the high IOPS demands of database applications, the authors propose a new consensus protocol called Parallel Raft. This protocol relaxes the strict sequential order of writes without sacrificing storage consistency, resulting in improved performance for parallel writes in Polar FS. The system was specifically designed for the POLARDB database service.



Storage Layer Abstraction representation in the Selfie architecture

18) *K. Hiraga, O. Tatebe, and H. Kawashima, "PPMDS: A distributed metadata server based on nonblocking transactions."*

This paper is centered on addressing the difficulties associated with the design of scalable distributed metadata. The authors recognize two significant challenges: first, how to efficiently maintain the hierarchical namespace in parallel and second, how to handle all file system activities across several servers. To tackle these problems, they present PPMDS, a parallel file system metadata server that improves metadata performance and scalability without jeopardizing file system semantics by employing a non-blocking transaction scheme. The PPMDS resolves the challenges by implementing server-side transaction processing, multiple readers' open-for-read status, and a shared lock mode scheme which minimizes unnecessary blocking time. The system was implemented and showed scalability up to three PPDMS servers, achieving around 62000 operations per second, a 2.58 times improvement in performance compared to one PPDMS server.

19) *Jianwei Liao, Guoqiang Xiao, and Xiaoning Peng: "Log-Less Metadata Management on Metadata Server for Parallel File Systems."*

In this paper, the author presents a new metadata management mechanism that aims to improve the performance and reliability of parallel and distributed file systems. This mechanism involves the Meta Data Server (MDS) handling metadata requests, while the client file system is responsible for caching and backing up some metadata queries in its memory. The testing findings revealed that this approach can greatly enhance metadata processing speed and I/O data throughput. In case of metadata server crashes or any failure occurs, the recovery of the metadata can be done by replaying the backup logs which is stored in the clients' memory. The authors implemented a prototype to demonstrate the feasibility of the idea. The metadata server keeps all metadata in memory, which improves the rate at which metadata is processed and I/O data is sent. The log-less metadata technique provided in this work may also be used to other parallel file systems, such as PVFS and Hadoop file systems.

20) *Dong Dai, Yong Chen, Philip Carns, John Jenkins, Wei Zhang, and Robert Ross "Managing Rich Metadata in High-Performance Computing Systems Using a Graph Model."*

The authors of the paper focus on the challenge of efficiently managing rich metadata in High Performance Computing (HPC) systems. HPC systems are capable of generating a large number of metadata every second, which includes traditional POSIX metadata, such as file size, name, and permission mode, as well as rich metadata that records a wider range of entities with more complex relationships between them. The authors propose a graph model for the uniform and efficient management of rich metadata in HPC systems, which is a critical aspect for identifying data sources, parameters behind results, and auditing data usage. The key challenge in implementing such a system is to effectively store, model, process and extract complex rich metadata in HPC platforms. To overcome this challenge, authors present a system called Graph Meta, which is optimized for HPC platforms and has been evaluated to show better results in performance and scalability when compared to existing solutions. However, the authors also acknowledge that there is still room for improvement in terms of fault tolerance and recovery capability, as well as consistency models and transaction support. In conclusion, the authors aim to provide a solution for handling extensive information in HPC systems, which is an important aspect for data management and analysis in these systems. By implementing a graph based HPC metadata management system, they hope to reduce the limitations of existing solutions and achieve a better overall efficiency and scalability of HPC systems.

21) *M. Satyanarayanan, J. J. Kistler, P. Kumar, M. E. Okasaki,E.H. Siegel, and D. C. Steere, "Coda: A highly available file system for a distributed workstation environment,"*

The authors in this introduce Coda file system which is derived from the Andrew File System (AFS) design storage but is more resilient to failures. The storage design uses two major mechanisms that are server replication and disconnected operation. This file system is usually used in large-scale distributed computing environment made up of Unix workstations. Coda helps in the improvement of availability and performance of the file system but is not specific to only these. The performance evaluation for this file system is not done, and hence there are possibilities of some performance loss and also occasional conflicts to provide higher availability. Server replication and replica management are studies in this file system.

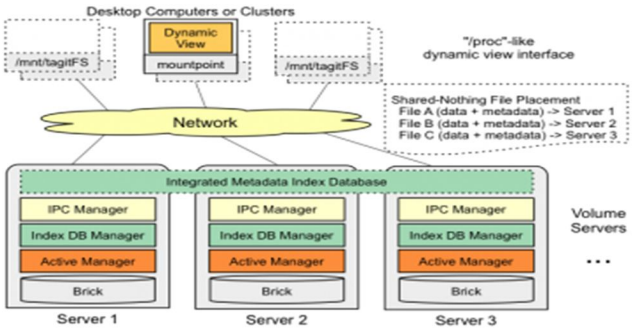
22) *Siyang Li, Tsinghua University, Youyou Lu, Tsinghua University, Jiwu Shu, Tsinghua University Tao Li, University of Florida, Yang Hu, University of Texas, Dallas. "LocoFS: A Loosely coupled Metadata service for distributed file system."*

The paper presents a novel solution to the challenge of managing metadata in distributed file systems. The current systems either use key-value storage or file system metadata to store and manage metadata, but both approaches have their limitations. Key-value storage has great performance for small metadata, but it lacks the necessary structure and relationships for large-scale metadata, making it difficult to maintain metadata in a file system. On the other hand, traditional file system metadata has the structure and relationships necessary for file systems, but its performance decreases with increasing metadata size. ToSS overcome these limitations, the authors proposes Loco FS, which is a loosely connected metadata service for distributed file systems. Loco FS uses two strategies to distribute the relationships between different types of metadata. Firstly, it separates the directory content and structure, this is achieved by organizing file and directory index nodes in a flat area and reversing the indexing of directory entries. Second, it separates the metadata of the file into access and content sections to enhance the performance of key-value access. With these strategies, Loco FS can achieve significantly better performance than either traditional file system metadata or key-value storage. In evaluations, Loco FS was able to achieve 1/4 latency compared to using one metadata server. These results demonstrate the feasibility of using Loco FS for metadata management in distributed file systems and provide a promising solution to the challenge of metadata management in large-scale systems.

23) *Hyogi Sim, Awais Khan, Sudharshan S. Vazhkudai, Seung-Hwan Lim, Ali R. Butt, Yougjae Kim. "An integrated indexing and search service for distributed file system."*

The authors in this paper present a scalable data management service framework called TagIt. This service provides scalable and distributed metadata framework.

This tool enhances the data discovery by flexible tagging capabilities. On evaluation of TagIt when implemented on two distributed file system that is glusterFS and ceph FS It was observed that TagIt can perform data search operation up to 10x over decoupled approach. This service is user-friendly and provides a dedicated command line utility and dynamic view which organizes data collection in a coherent directory hierarchy. It can be particularly useful when user wishes to extract a large multidimensional variable in a huge database.



GlusterFS TagIt architectural overview.

24) Qing Zheng, Charles D. Cranor, Gregory R. Ganger, Garth A. Gibson, Gary A. Grider, Bradley W. Settlemyer, George Amvrosiadis "DeltaFS: A Scalable No-Ground-Truth Filesystem for Massively-Parallel Computing."

The authors of this paper introduce Delta FS, a novel distributed filesystem metadata paradigm that replaces High-Performance Computing (HPC), which is recognized for its widespread use and difficult parallel filesystem control plane. A Scalable No-Ground-Truth Filesystem for Extremely Parallel Computing is called Delta FS. By enabling effective inter-job communication and drastically lowering client metadata operation latency up to 49 and resource utilization up to 52, Delta FS shortens the entire workflow runtime. On top of a shared object store, Delta FS is a set of library functions and daemon processes that offer scalable parallel filesystem metadata access. A C++ prototype called Delta FS was created. By spreading work over all accessible nodes, Delta FS exhibits good results in scalable performance.

25) J. Chen, Q. Wei, C. Chen, and L. Wu, "FSMAC: A file system metadata accelerator with non-volatile memory,"

The authors of this paper proposed a solution to the performance gap between key-value storage and file system metadata. They suggested using Nonvolatile Memory (NVM) as a potential alternative to the traditional storage medium (e.g., Hard Disk Drive, Solid State Drive) for file system metadata. To achieve this, the authors introduced a filesystem metadata accelerator (FSMAC) which separates the data and metadata paths. The metadata is stored in NVM which has faster access speed and is byte-addressable, allowing for efficient metadata access. The data is stored on disk and accessed in block form from the I/O bus. The authors tested their solution using the Linux kernel's FSMAC implementation on a standard server equipped with NVDIMM and found that it had improved performance and consistency compared to the traditional Ext4 file system. The consistency of FSMAC was validated through regular and abnormal shutdown tests.

26) Swapnil Patil and Garth Gibson. "Scale and concurrency of GIGA+: File system directories with millions of files."

The authors in this paper discuss about the problem of scalability of file system directories they introduced a tool called GIGA+ which divides the entries in the directories over a cluster of server nodes. The scalability is achieved by giving each server the control of the decision about the migration for load balancing GIGA+ works on two internal implementation principle they are firstly partitioning index among all servers without synchronization or serialization and secondly coherently tolerate stale index state at the clients. The evaluation of the experimental results shows that GIGA+ delivers a sustainable throughput for more than 98000 files create per second on a 32-server cluster and a resilient load balancing.

27) M. Burrows, "The chubby lock service for loosely coupled distributed systems."

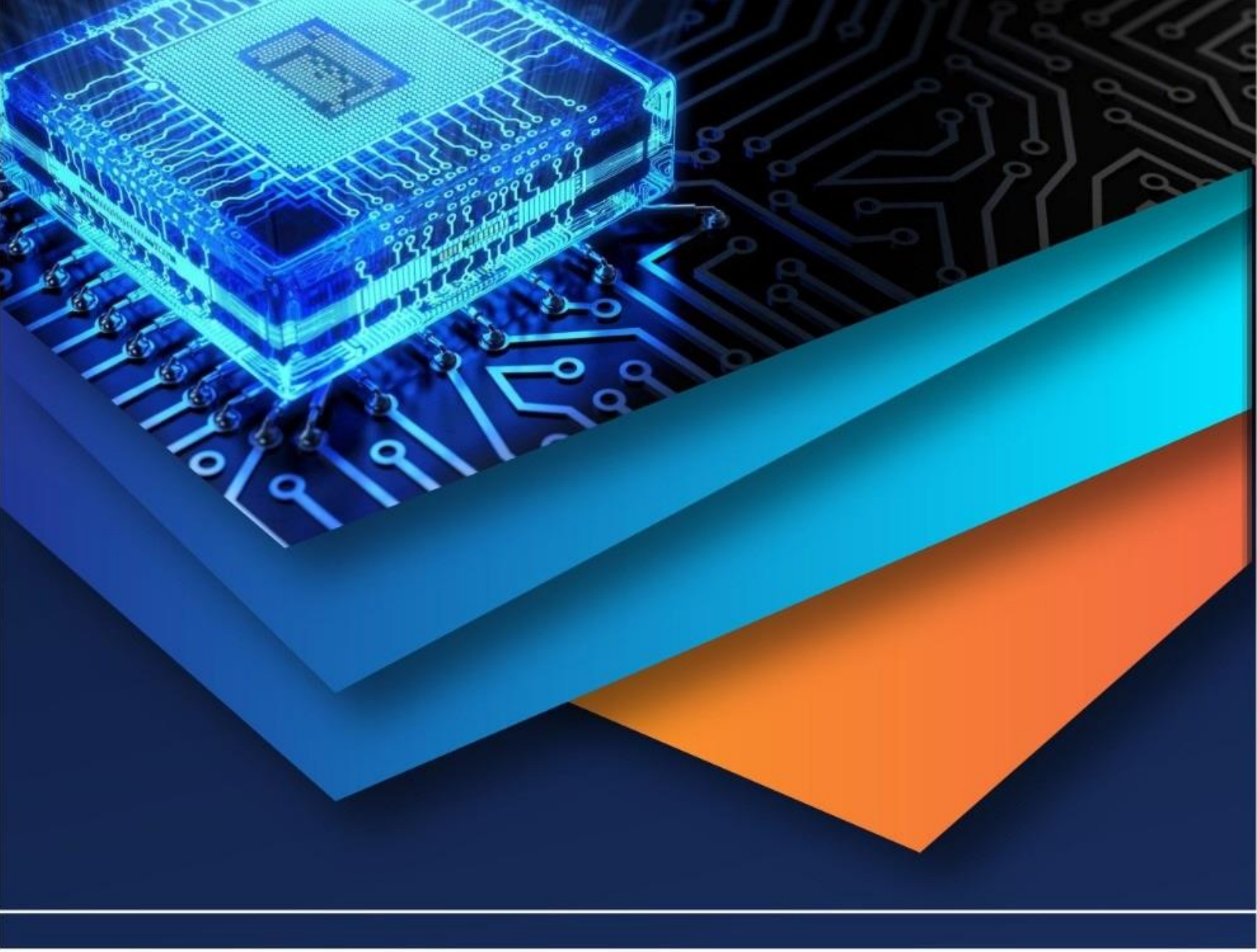
The Chubby lock service, designed to offer coarse-grained locking and dependable storage for a loosely coupled distributed system, is discussed in this work by the author. Although it has an advisory lock and provides an interface for a distributed file system, the architecture prioritizes availability and stability above high performance. The author shows how the design had to be altered to account for the changes after describing the initial design and intended usage, contrasting it with the actual use, and explaining the discrepancies. The client can coordinate their actions and decide on the most important pieces of information for the lock service. With name components separated by slashes, it is composed by a rigid tree of files and directories in the typical fashion. It is a standard repository for files that require high availability, such as access control lists, and is built on well-known concepts like distributed consensus across a few clones for fault tolerance and other things.

IV. CONCLUSION

In this paper we have surveyed about various methods of metadata management systems involving distributed file systems, cloud storage systems, on-site systems and many others. There were three areas where we mainly focused on, high performance, high scalability and high availability of the file storage systems. There are already many tools and systems present in the market which improvise the above stated areas but many of them have certain disadvantages attached to them. A more efficient metadata management tools with faster processing will be a fine add to the field. Therefore, with the help of the knowledge gained through literature survey we will try on implementing a plausible solution for the metadata management with high performance, availability and scalability and ultimately try to remove as much as limitations from the system as possible.

REFERENCES

[1] J. Sun and C. K. Reddy, "Big data analytics for healthcare," in Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2013, Art. no. 1525.
[2] E. Kakoulli and H. Herodotou, "OctopusFS: A distributed file system with tiered storage management," in Proc. ACM Int. Conf. Manage. Data, 2017, pp. 65–78.
[3] S. A. Weil, K. T. Pollack, S. A. Brandt, and E. L. Miller, "Dynamic metadata management for petabyte-scale file systems," in Proc. ACM/IEEE Conf. Supercomputer., 2004, Art. no. 4.
[4] A. W. Leung, M. Shao, T. Bisson, S. Pasupathy, and E. L. Miller, "Spyglass: Fast, scalable metadata search for large-scale storage systems,"
[5] A. Leung, M. Shao, T. Bisson, S. Pasupathy, and E. Miller, "High performance metadata indexing and search in Peta scale data storage systems," in Proc. J. Phys. Conf. Ser., vol. 125, no. 1, 2008, Art. no. 012069
[6] A. B. M. Moniruzzaman, "NewSQL: Towards next-generation scalable RDBMS for online transaction processing (OLTP) for Big Data management," Int. J. Database Theory Appl., vol. 7, no. 6, pp. 121–130, 2014.
[7] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, "Ceph: A scalable, high-performance distributed file system," in Proc. 7th Symp. Oper. Syst. Des. Implementation, 2006, pp. 307–320.
[8] Sage A. Weil Scott A. Brandt Ethan L. Miller Carlos Maltzahn. CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data.
[9] Sasha Ames, Maya B. Gokhale, Carlos Maltzahn. QMDS: A File System Metadata Management Service Supporting a Graph Data Model-based Query Language.
[10] J. Liu, D. Feng, Y. Hua, B. Peng, and Z. Nie, "Using provenance to efficiently improve metadata searching performance in storage systems," Future Gener. Comput. Syst., vol. 50, pp. 99–110, 2015.
[11] G. A. Gibson et al., "A cost-effective, high-bandwidth storage architecture," in Proc. 8th Int. Conf. Architectural Support Program. Lang. Oper. Syst., 1998, pp. 92–103.
[12] I. Kettaneh et al., "The network-integrated storage system," IEEE Trans. Parallel Distrib. Syst., vol. 31, no. 2, pp. 486–500, Feb. 2020
[13] J. Zhou, Y. Chen, W. Wang, S. He, and D. Meng, "A highly reliable metadata service for large-scale distributed file systems," IEEE Trans. Parallel Distrib. Syst., vol. 31, no. 2, pp. 374–392, Feb. 2020.
[14] X. Wu and Z. Shao, "Selfie: Co-locating metadata and data to enable fast virtual block devices," in Proc. 8th ACM Int. Syst. Storage Conf., 2015, pp. 1–11.
[15] Alexander Thomson, Google; Daniel J. Abadi, Yale University CalvinFS: Consistent WAN Replication and Scalable Metadata Management for Distributed File Systems.
[16] L. Pineda-Morales, A. Costan, and G. Antoniu, "Towards multi-site metadata management for geographically distributed cloud workflows," in Proc. IEEE Int. Conf. Cluster Comput., 2015, pp. 294–303.
[17] W. Cao et al., "PolarFS: An ultra-low latency and failure resilient dis tributed file system for shared storage cloud database," Proc. Very Large Data Base Endowment, vol. 11, no. 12, pp. 1849–1862, Aug. 2018.
[18] K. Hiraga, O. Tatebe, and H. Kawashima, "PPMDS: A distributed metadata server based on nonblocking transactions," in Proc. 5th Int. Conf. Social Netw. Anal., 2018, pp. 202–208.
[19] Jianwei Liao, Guoqiang Xiao, and Xiaoning Peng: Log-Less Metadata Management on Metadata Server for Parallel File Systems.
[20] Dong Dai, Yong Chen, Philip Carns, John Jenkins, Wei Zhang, and Robert Ross Managing Rich Metadata in High-Performance Computing Systems Using a Graph Model.
[21] M. Satyanarayanan, J. J. Kistler, P. Kumar, M. E. Okasaki, E. H. Siegel, and D. C. Steere, "Coda: A highly available file system for a distributed workstation environment," IEEE Trans. Comput., vol. 39, no. 4, p. 447–459, Apr. 1990
[22] Siyang Li, Tsinghua University, Youyou Lu, Tsinghua University, Jiwu Shu, Tsinghua University Tao Li, University of Florida, Yang Hu, University of Texas, Dallas. LocoFS: A Loosely coupled Metadata service for distributed file system.
[23] Hyogi Sim, Awais Khan, Sudharshan S. Vazhkudai, Seung-Hwan Lim, Ali R. Butt, Yougjae Kim. An integrated indexing and search service for distributed file system
[24] Qing Zheng, Charles D. Cranor, Gregory R. Ganger, Garth A. Gibson, Gary A. Grider, Bradley W. Settlemyer, George Amvrosiadis DeltaFS: A Scalable No-Ground-Truth Filesystem for Massively-Parallel Computing.
[25] J. Chen, Q. Wei, C. Chen, and L. Wu, "FSMAC: A file system metadata accelerator with non-volatile memory," in Proc. IEEE 29th Symp. Mass Storage Syst. Technol., 2013, pp. 1–11.
[26] Swapnil Patil and Garth Gibson. Scale and concurrency of GIGA+: File system directories with millions of files
[27] M. Burrows, "The chubby lock service for loosely-coupled distributed systems," in Proc. 7th Symp. Oper. Syst. Des. Implementation, 2006, pp. 335–350.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)