



IJIRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2320-6710



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Issue 5, May 2022

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.118



9940 572 462



6381 907 438



ijirset@gmail.com



www.ijirset.com



Fog-Based Data Distributed System with Map-Reduce Technique

Prof. Chetan Kumar G S¹, Shamataj²

Assistant Professor, Department of MCA, University BDT College of Engineering, Davanagere, Karnataka, India¹

MCA Student, Department of MCA, University BDT College of Engineering, Davanagere, Karnataka, India²

ABSTRACT: The MapReduce programming model makes it easier to process large amounts of data on commodity clusters by using tasks that can be done in parallel. Even though many people work to improve the performance of MapReduce jobs, they don't pay attention to the network traffic that happens during the shuffle phase, which is very important to speeding things up. In this example, a hash function is used to divide up intermediate data between reduce tasks. This is inefficient because the network topology and data size of each key are not taken into account. In this study, a new intermediate data storage "fog centre" was made to cut down on the cost of network traffic for a MapReduce job. This study also takes into account the aggregator placement problem, which is when each aggregator can reduce traffic from multiple map tasks by putting them in the right place. Geologically based information distribution is one way to deal with the big data optimization problem. A new "Missing Block Transfer" algorithm is also being used to dynamically get only the necessary blocks from the primary cloud services when they aren't there.

KEYWORDS: MapReduce, Fog-Center, Hashing.

I. INTRODUCTION

The vast majority of monitoring solutions are based on the cloud computing architecture. High latency introduced by cloud architecture when processing large amounts of data restricts large-scale deployment of latency sensitive applications. Fog computing reduces latency by bringing computer servers closer to consumers, while also enhancing on-demand scalability and accessibility to resources. Fog-based monitoring system design is proposed by Asghar et al. to minimise latency and network utilisation. In addition, a novel Load Balancing Scheme (LBS) is proposed to balance the load across fog nodes when the monitoring system is widely implemented [9].

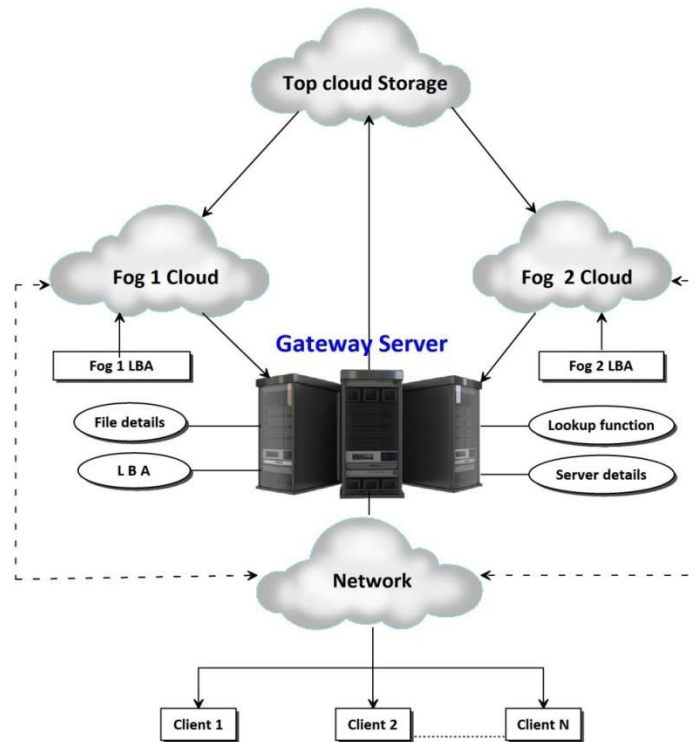
Fog computing has been suggested for next-generation mobile networks to move computation from the cloud to the network edge to meet the need of so many apps that require low latency. [10].

To improve the efficiency of IT resources, deduplication is increasingly commonly employed in cloud data centres. Dedupe was developed by Fu et al. to solve this problem by optimising distributed deduplication using inter-node two-tiered data routing and intra-node application-aware deduplication in the cloud environment. AppDedupe uses super-chunk handprints and application-aware similarity indexes to expedite intra-node deduplication [1].

Block Level Deduplication is also known as MapReduce. It is important to investigate how MapReduce programmes impact the design of distributed MapReduce systems, particularly if they gain equally from modelling the reduction process across containers rather than individual key-value pairs.

Fog-based distributed server processing may be accomplished using IP-LookUp services. Client queries are routed via a gateway server or hypervisor. The IP-Lookup method will be used to determine which server to make a request to. It is possible to do map reduction using block splitting and hashing. In order to identify whether or not a block has already been created, the hashing method is employed. The logical block addressing (LBA) approach is used to retrieve a file.

In contrast to the typical cloud-based centralised approach, which distributes processing jobs to distinct subprocessors to save computing time, the suggested solution uses a fog-based data dispersed system with MapReduce as its backbone. Each subsystem may be optimised independently using the Map-Reduce approach while a traffic model and a set of communication rules between subsystems are developed to guarantee the total transportation network can be optimised worldwide.



II. LITERATURE REVIEW

Khan and others In distributed storage systems, deduplication has mostly been utilised to increase the efficiency of storage space use. A central metadata bottleneck, scalability, and storage rebalancing are all features often ignored in traditional deduplication research. Khan et al. presented GRATE, a high-performance inline cluster-wide data dedupe system that conforms to the design restrictions of shared-nothing storage systems. As an example, GRATE utilises a distributed deduplication metadata shard technique that leverages chunk content hashes to reduce the amount of I/O broadcasts and object relocation issues by eliminating redundant copies. In the case of a failure, reference errors and missing data chunks are recovered using a tagged consistency technique. At chunk sizes of less than 128KB, the test found an average 18% increase in performance bandwidth over the content addressable deduplication technique [2]. At the same time, storage space was reduced by a significant amount.

Map-reduce run-time solution presented by Arif et al. aims for scalability and integration with other parallel applications. A new programming environment defines reduction operations on data containers instead of key-value pairs. As a result of this design option, more efficiency may be achieved. Our runtime is optimised for systems with shared memory, which can now store data sets up to terabytes in memory and execute data analytics on them[3].

In the context of data deduplication, Jiang et al. address the concerns of data confidentiality and ownership management. This is an unique PoW strategy that leverages the Bloom filter to survive poison attacks and the fake faking attacks of cross-user file deduplication. Reduce update frequency and calculation cost for cross-user file-level



deduplication by using the lazy update technique, which also ensures forward and backward secrecy for successive uploaders. Deduplication at the user block level may also be accomplished with the help of the user-assisted CE. With the exception of side channel attacks, our suggested approach can fulfil the security criteria since it employs the PoW verifying process in order to accomplish mutual PoW confirmation while conserving bandwidth for data deduplication. Only 5 percent more processing is required by the proposed approach to verify and maintain the ownership of PoW than is required by REC, according to a performance study.

Luo et al. introduce Boafft, a cloud storage system with distributed deduplication. The parallelization of deduplication over several storage nodes with a minimal loss of deduplication ratio allows it to achieve scalable throughput and capacity. In order to achieve high throughput and high capacity, the Boafftutilises several servers to deduplicate data in parallel while maintaining a low loss of deduplication ratio. An excellent data routing technique is used by the Boafft, which decreases network overhead by quickly determining the storage location and thereby reduces network traffic. For the second time, local data deduplication is expedited thanks to the Boafft's in-memory similarity indexing, which reduces the amount of random disc reads and writes. Each data server has a hot fingerprint cache built in to boost the deduplication ratio [5]. [5]

For density-based clustering hierarchies, Santos and coworkers demonstrate why adapting current approaches to parallelize Single-Linkage clustering with MapReduce leads in inefficient solutions. To get you started, here's a random block parallelization strategy that's both exact and computationally intensive. We propose a different parallelization scheme that uses a much faster, recursive sampling approach to compute an approximate clustering hierarchy in order to apply hierarchical density-based clustering to large datasets efficiently using MapReduce. Using the HDBSCAN* hierarchical density-based clustering algorithm and the data bubbles data summarising methodology, this approach is based on cutting-edge research. The suggested method's efficiency and scalability have been shown on a number of datasets, both in terms of runtime and approximation quality[6].

Volunteer Computing, a system that should be able to provide computing resources for no cost, is one solution to this problem. However, variable machine behaviour is an issue with Big Data data distributions. colleagues of Dos A hybrid cloud computing and volunteer computing system was suggested by Dos Anjos and colleagues for Big Data analytics. It is important to note that the following contributions are made by this piece: Hybrid infrastructures must be evaluated in order for Big Data to be deployed efficiently; an HR Alloc Algorithm must be developed in order to put data for use in Big Data applications; and a resource allocation model for hybrid infrastructures must be developed. A hybrid cloud architecture with up to 35% unstable computers may be used in the worst-case scenario without performance degradation and for less than 20% of the cost of standard cloud computing, according to the findings. In the best-case scenario, load balancing may save communication expenses by as much as 57.14 percent [7].

Yen et al. recommend an integrated strategy to data deduplication that includes both the host I/O stack and the storage firmware in order to lessen the write load on storage devices for lifetime extension. In the first place, it was observed that the SQLite library creates a large part of copy-induced duplicate data, which may be completely disclosed to mobile storage through the right alignment of file types. The small flash read delay serves as inspiration for a firmware-based method for spotting data duplication without the need for hardware.. Enable flash trash collection to detect data duplication. Data compression and differential logging are ineffective for encrypted mobile storage, according to our simulation findings. [8]

Generally speaking, it is considered that heterogeneous MapReduce placement optimization beats homogeneous MapReduce placement optimization in terms of reducing the overall operating cost of cloud-based MapReduce calculations. Heterogeneous MapReduce placement optimization is the focus of Xu et al technique. 's Constrained combinatorial optimization is used to address this novel approach's map-reduce placement optimization issue [12], which was previously a heterogeneous optimization problem.



Cloud-based IoT services, according to Maswood et al., have higher latency and less efficient resource use. By moving cloud services closer to the edge, distributed fog computing may help alleviate these issues. In order to enable real-time applications at a cheaper operating cost, this research looked at an integrated fog cloud environment. Cooperative fog-cloud computing environments are shown first, and then a novel optimization approach is proposed. A composite goal function is used in this approach to lower bandwidth costs while also balancing the amount of traffic. Both bandwidth and CPU processing capability should be considered while balancing traffic [13].

Li and colleagues A new data-centric architecture based on three essential techniques: a cross-cloud virtual cluster, job placement focused on data, and network coding-based traffic routing – is required due to the inefficiency of the current system and the high cost of providing big data help. MapReduce tasks may be conducted in a globally dispersed cloud using our approach, which aims to minimise both compute and transmission costs. Create a parallel algorithm by splitting down the original large-scale issue into multiple distributively solvable subproblems organised by a high-level master problem.

Research attempts to increase MapReduce performance were surveyed and a simple model was developed to define the processing durations in each phase. Lee and colleagues Map and Shuffle inefficiencies have been discovered using the model and experimental data. Pre-arranging data blocks to the ultimate place rather than the intermediate site may reduce inefficiencies in subsequent stages by minimising data transmission. Deep data locality is the term for this (DDL). We investigated a way to boost DDL during the Map and Shuffle phases. It was found that RLM has a detrimental influence on performance, and that its unpredictable nature causes performance variations. Removing RLM improves performance by increasing the degree of DDL[15].

In Fog Computing, where nodes are often overburdened and response times suffer as a result, Pereira et al. report that the growing heterogeneity of the Internet of Things (IoT) is a growing source of worry. Propose an architecture model for fog computing with a main focus on load balance, with the objective of boosting fog node efficiency in response to this problem. Our study blends information and computational dynamics burden in order to minimise Fog Computing's reaction time. Direct and round-robin situations were compared, and the suggested strategy was shown to be faster than either. Even when the number of requests climbed five- and tenfold, the flow rate of the load balancer remained constant. Real-world applications of this idea may be shown by the fact that all response times for key priority jobs were less than 60 ms in both tests.

Dynamic and efficient load balancing technology might be utilised to increase performance and cut energy usage dramatically. Through the use of load balancing technology, computers may share or shift their workload. The Dynamic Energy Efficient Resource Allocation (DEER) technique proposed by Rehman et al. is a new approach to load balancing in fog computing systems. and other colleagues of Rehman Tasks are initially submitted by the user to the Tasks Manager, which then executes them. Resources from Cloud Data Centers are registered by the Resource Information Provider (RiP). It is then sent to the Resource Scheduler with the job and resources. The resource scheduler prioritises available resources based on how often they've been used. The resource engine distributes tasks to resources in the order listed after receiving information about tasks and resources from the resource scheduler. The Resource Load Manager and the Resource Power Manager are additionally informed of the current state of resources throughout task execution. Resources may be turned off or on to manage energy use. The user is notified of successful task completion by the resource engine. It was shown that the suggested technique was able to lower energy consumption and computing costs in fog conditions by 8.67 percent and 16.77 percent, respectively, via simulations [17].

Wang and colleagues Due to the restricted computing resources and processing power of fog devices, lightweight computing has been the dominant option; nonetheless, more time and load constraints have been imposed. Finally, in the Cloud-fog computing architecture, a task scheduling technique (I-FASC) for task and resource characteristics, as well as an upgraded firewall algorithm (I-FA) that attempts to include the explosion radius detecting mechanism

of fireworks, are offered as a result Experiment in a cloud-fog computing system, and two sets of results suggest that this strategy may improve execution speed and assure better load in a short time period.

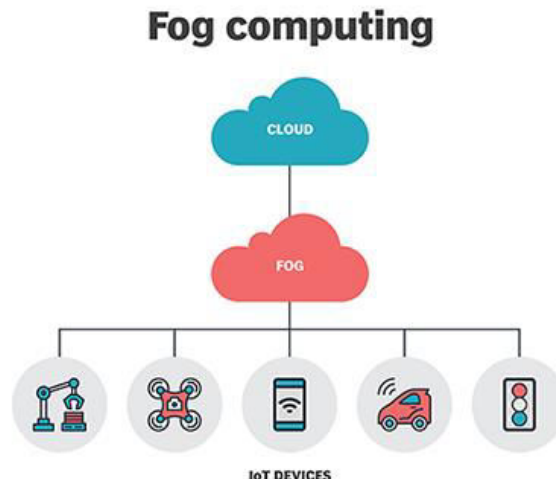
Consequently, there is a growing need for services that can be reliably delivered to users in dynamic contexts that have strict time-to-job completion requirements. Cheng et al RDS, 's a Resource and Deadline-aware Hadoop task Scheduler that considers future resource allocation, is designed to minimise job deadline misses. Using an efficient receding horizon control mechanism, create an online optimization problem from the work scheduling issue. Improve control by developing a self-learning model for estimating completion times for jobs. In order to enable a wide range of dynamic clusters, extend the RDS scheduler architecture. In place of an all-or-nothing deadline, adopt a range of flexible deadline time limits [19].

Dolev and coworkers. Businesses and universities are being forced to rethink their present big-data processing methods because of the increasing volume of data that is geographically distributed. Unlike the existing system, the new frameworks are supposed to be able to process data from all over the world without having to move the raw datasets to one central place. Task assignment, data localization, data mobility, network bandwidth, and security and privacy are all crucial to geo-distributed big-data processing. There are numerous degrees of transparency for the data in this geo-distributed system so that it may be kept in many places with varied levels of openness[20].

Research Methods & Specification

Fog Computing:

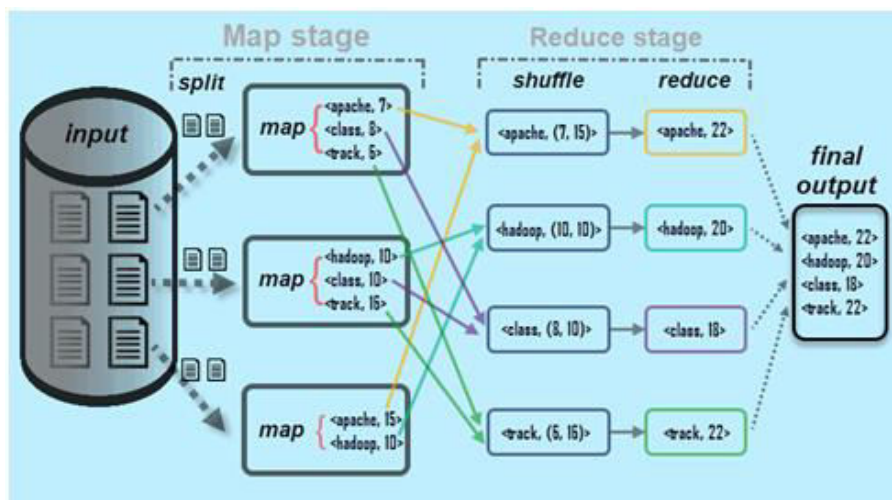
Using fog computing, data, computation, storage, and applications are spread between the cloud and the data source.. As with edge computing, cloud advantages and power are brought closer to the point at which data is produced and acted upon.



As a whole, fog computing serves to improve service quality by speeding up replies, reducing network latency, and aiding in backbone bandwidth savings (QoS). In addition, it's designed to communicate useful data to the cloud.

MapReduce:

Distributed processing of large data sets is made possible via the MapReduce programming architecture. The data must first be split and then reassembled in order to get the final result. Many programming languages include MapReduce libraries, and each has its own set of optimizations.



MapReduce distributes data pieces across a Hadoop cluster's nodes. The idea is to divide a dataset into chunks and apply an algorithm to processing all of them at once. Parallel processing on numerous workstations dramatically improves the speed with which petabytes of data can be handled.

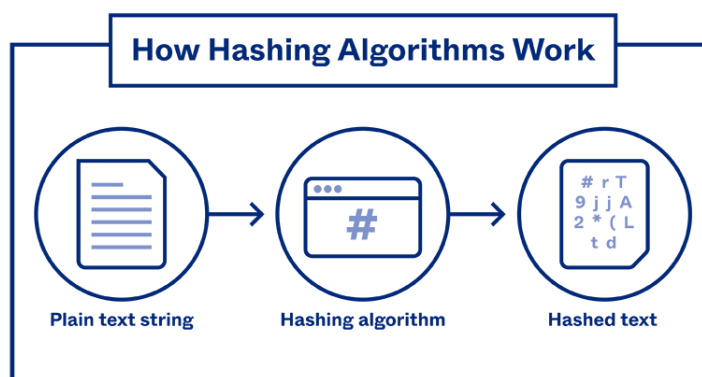
IP look up:

When trying to decipher IPV4 or IPV6 addresses, an IP lookup, also known as an IP address lookup or an IP checker, is the technique used.

In networking, IP addresses are converted to internet domain names using IP address lookup. A forward IP address lookup is used to convert a domain name to an IP address. To get the name, the IP address number must be reversed. Most computer users are unaware of this method.

Hashing:

In the hashing algorithm, a fixed-size hash is used to map data. A cryptographic hash function is the algorithm for generating a hash value. One-way hash algorithms, or hash functions, are meant to work in a similar fashion to one-way hash functions. The inversion cannot be done, i.e., the original value cannot be recovered from the hash.



Characteristic of a Hashing Algorithm

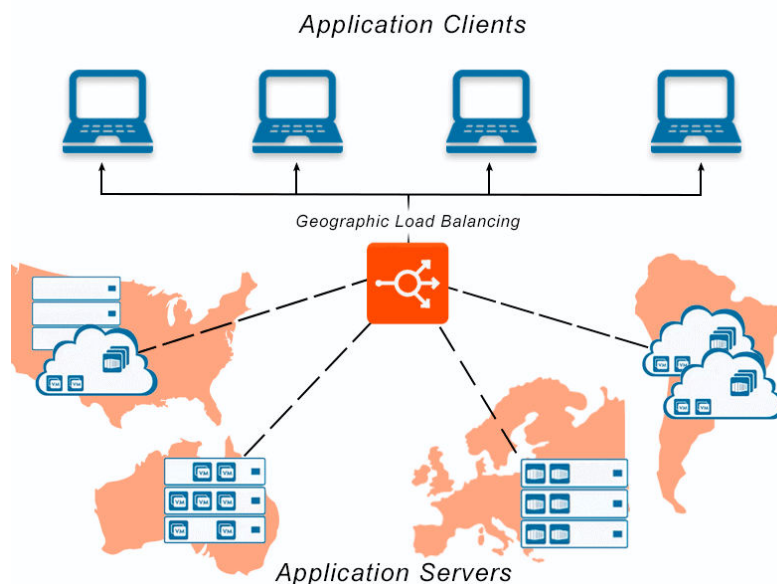
- The hashing algorithm must be able to hash any type of data quickly.
 - The algorithm should prevent the message from being regenerated from its created hash value (one way).
 - At no point in time, the two messages must have the same hash.
- To put it another way, the hashing algorithm must avoid colliding.
- If the message's hash value changes even slightly, the message's hash value must change as well.

The avalanche effect is what it's called.

Load balancing in cloud

Network traffic may be evenly distributed using load balancing to prevent network failure due to resource overload. This technique improves the speed and availability of applications, websites, databases, and other computer resources. User queries may also be handled more quickly and accurately thanks to this feature. Keep any one server from getting overworked and perhaps failing is the primary purpose of load balancing. Increased service availability and reduced downtime may be attributed to load balancing.

Geographic Load Balancing



Geographic load balancing redistributes application traffic over several data centres in order to maximise efficiency and security. Geographic load balancing makes use of various data centres, whereas local load balancing takes place inside a single data centre.

Network traffic bandwidth:

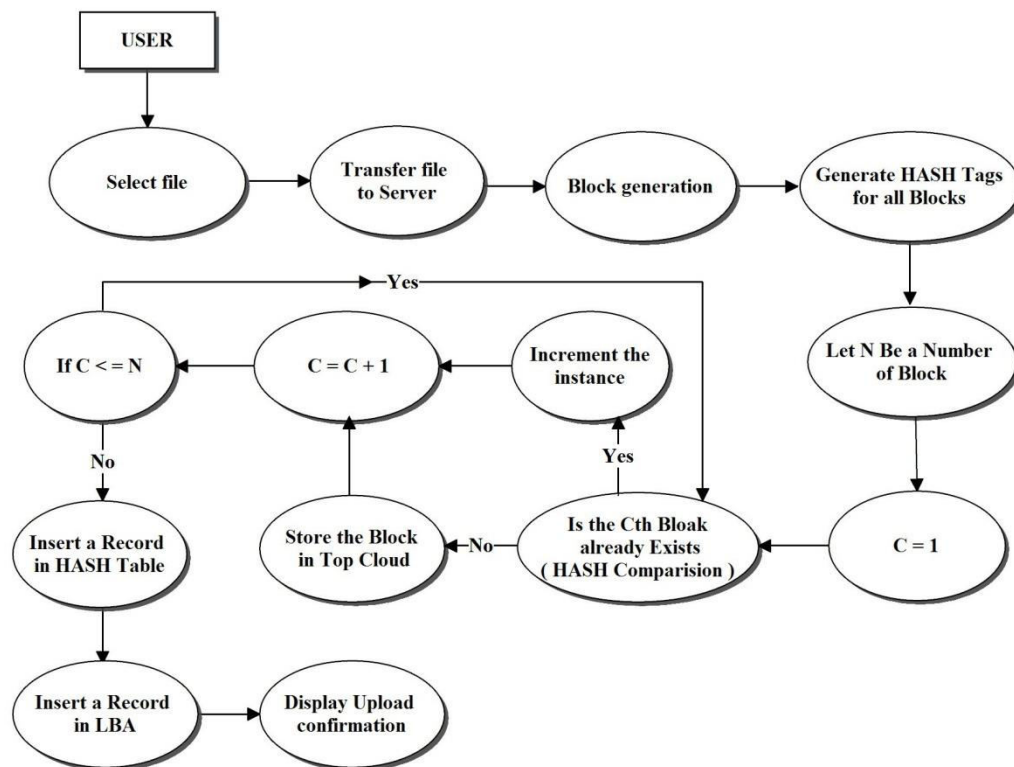
A network's bandwidth is a measure of its ability to transmit data. When you use the Internet, you're using bandwidth. Both uploads and downloads make use of available bandwidth. When the network's capacity isn't being taxed, data moves quickly and seamlessly.

How does bandwidth affect network traffic?

How much data can be transferred and received in a given period of time is measured by the term "bitrate." There are three types of bandwidth measurements: bits, megabytes, and gigabytes per second (GBPS). Because bandwidth does not really increase network speed, it is important to recognise that it just seems to do so.

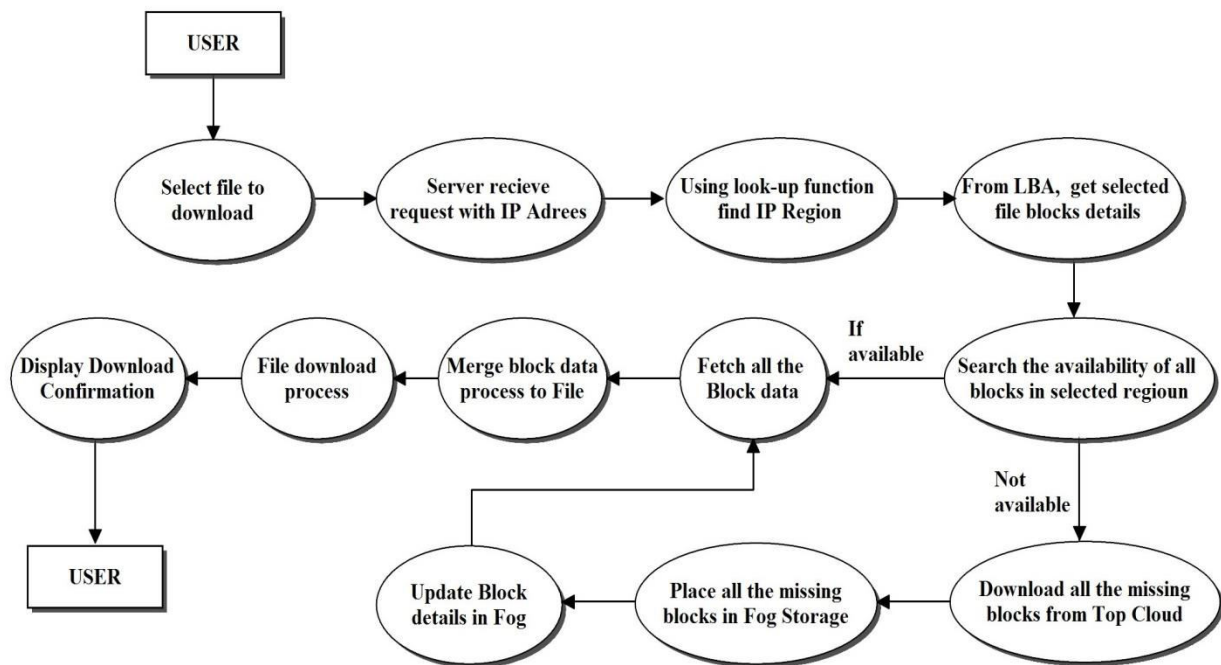
Methodology Used:

A "Missing Block Transfer" strategy to dynamically changing data division and aggregation is offered to cope with the large-scale optimization difficulty for big data applications based on decomposition.



File Upload Process Steps

- ✓ The user must choose a file from the system and then click the upload option.
- ✓ The files will be separated into little parts (N bytes each block).
- ✓ Create a hash tag for each block.
- ✓ If the created hash block matches an existing hash tag from the database, we will not upload that block to the cloud and instead increase the number of instances of that block in the database table.
- ✓ If the hash tag does not match, we will add the block's hash details to the database and upload it to the cloud.
- ✓ LBA - Logical Block Addressing - is a technique for determining which blocks are present in a file.



File Download Process Steps

- ✓ The user must choose a file to download.
- ✓ When a request is sent to the server, the server must obtain the client system's IP address and, using the Look-UP function, determine the IP address Region, which should be Selected (SR).
- ✓ There is one Storage Space and one Map Area in each Region. All downloaded block data is available in the Storage section, and downloaded block information are available in the Map area.
- ✓ The LBA server must find block numbers in the specified file using LBA.
- ✓ If all of the blocks required for the file are available in SR storage space, the server must check in the Map area of SR whether all of the blocks required for the file are available; if all of the blocks are available in SR storage space, the server must download and merge the blocks before giving it to the user. If only a few blocks are available, retrieve the missing blocks from Top Region, place them in SR storage space, and update the SR Map data.

III. CONCLUSION AND FUTURE WORK

As a distributed and hierarchical architecture with scheduled storage, virtual machine images (embedded virtual cache and temporary storage), and network resources deployment with the supplement of CC capability at the edge of IoT, we must conclude our fog computing overview. Some of the most important aspects of this study, such as how fog computing may extend CC services at the edge, were represented in this paper. There are many use cases for fog computing, such as real-time data analysis in health care, STLS, and smart grid, which led to the development of fog computing. There are three key areas of focus for our research: IoT, big data analytics, and storage. For the near-term research goals and distributed application development for flexibility in QoS of the fog computing framework, this study focused on the future possibilities of the developing fog era. Fog computing may be divided into three categories, each dependent on an existing essential technology: compute (e.g., computation offoading and latency management), communication (e.g., SDN, NFV, and 5G), and storage (precache system and



storage expansion). Existing essential technologies might improve fog networking's intelligence. It is possible to boost network scalability and save costs by using SDN and NFV in tandem, as well as live VM migration (computation offloading). Fog computing offers numerous advantages for company continuity, efficiency, and cost reduction, but it also comes with a number of difficulties when it comes to industrial data management. Researchers are looking at IIoT applications, especially in emerging areas such as manufacturing and oil & gas, which use fog computing the most. Research on smart city development is still in its early stages, and many studies are focusing on combining smart buildings, STLs, and sensor networks that are currently in place. In contrast to cloud computing, which offers wide-area connection, global coordination, high-performance computing, and enormous storage, fog computing focuses on the needs of individual users, pools edge resources, accelerates innovation, and allows for real-time processing [6]. During this changing moment, it's a good time to begin exploring what fog may look like and how it might affect the virtual computing world in the next 10 years.

REFERENCES

- [1] Fu, Y., Xiao, N., Jiang, H., Hu, G. and Chen, W., 2017. Application-aware big data deduplication in cloud environment. *IEEE transactions on cloud computing*, 7(4), pp.921-934, JCR factor= 5.938.
- [2] Khan, A., Hamandawana, P. and Kim, Y., 2020. A Content Fingerprint-Based Cluster-Wide Inline Deduplication for Shared-Nothing Storage Systems. *IEEE Access*, 8, pp.209163-209180, JCR factor= 3.367.
- [3] Arif, M., Vandierendonck, H., Nikolopoulos, D.S. and de Supinski, B.R., 2016, December. A scalable and composable map-reduce system. In *2016 IEEE International Conference on Big Data (Big Data)* (pp. 2233-2242). IEEE, CORE RANK=B.
- [4] Jiang, S., Jiang, T. and Wang, L., 2017. Secure and efficient cloud data deduplication with ownership management. *IEEE Transactions on Services Computing*, 13(6), pp.1152-1165. JCR factor= 8.216.
- [5] Luo, S., Zhang, G., Wu, C., Khan, S.U. and Li, K., 2015. Boafft: Distributed deduplication for big data storage in the cloud. *IEEE transactions on cloud computing*, 8(4), pp.1199-1211, JCR factor= 5.938.
- [6] dos Santos, J.A., Syed, T.I., Naldi, M.C., Campello, R.J. and Sander, J., 2019. Hierarchical density-based clustering using MapReduce. *IEEE Transactions on Big Data*, 7(1), pp.102-114. JCR factor= 3.344.
- [7] Dos Anjos, J.C., Matteussi, K.J., De Souza, P.R., Grabher, G.J., Borges, G.A., Barbosa, J.L., Gonzalez, G.V., Leithardt, V.R. and Geyer, C.F., 2020. Data processing model to perform big data analytics in hybrid infrastructures. *IEEE Access*, 8, pp.170281-170294. JCR factor= 3.367.
- [8] Yen, M.C., Chang, S.Y. and Chang, L.P., 2018. Lightweight, integrated data deduplication for write stress reduction of mobile flash storage. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(11), pp.2590-2600, JCR factor= 2.807.
- [9] Asghar, A., Abbas, A., Khattak, H.A. and Khan, S.U., 2021. Fog based architecture and load balancing methodology for health monitoring systems. *IEEE Access*, 9, pp.96189-96200. JCR factor= 3.367.
- [10] Yuan-Yao Shih, Chih-Yu Wang, Ai-Chun Pang, "Fog Computing Service Provision Using Bargaining Solutions", *IEEE Transactions on Services Computing* (Volume: 14, Issue: 6, Nov.-Dec. 1 2021), Pp: 1765 – 1780. JCR factor=8.216.
- [11] H. Ke, P. Li, S. Guo and M. Guo, "On Traffic-Aware Partition and Aggregation in MapReduce for Big Data Applications," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 3, pp. 818-828, 1 March 2016, doi: 10.1109/TPDS.2015.2419671, JCR factor= 2.687.
- [12] Xu, X. and Tang, M., 2015. A new approach to the cloud-based heterogeneous MapReduce placement problem. *IEEE Transactions on Services Computing*, 9(6), pp.862-871. JCR factor= 8.216.
- [13] Maswood, M.M.S., Rahman, M.R., Alharbi, A.G. and Medhi, D., 2020. A novel strategy to achieve bandwidth cost reduction and load balancing in a cooperative three-layer fog-cloud computing environment. *IEEE Access*, 8, pp.113737-113750. JCR factor= 3.367.
- [14] Li, P., Guo, S., Yu, S. and Zhuang, W., 2015. Cross-cloud mapreduce for big data. *IEEE Transactions on Cloud Computing*, 8(2), pp.375-386. JCR factor= 5.938.
- [15] Lee, S., Jo, J.Y. and Kim, Y., 2016, October. Performance improvement of mapreduce process by promoting deep data locality. In *2016 IEEE international conference on data science and advanced analytics (DSAA)* (pp. 292-301). IEEE. CORE RANK=A.



- [16] Pereira, E.P., Padoin, E.L., Medina, R.D. and Méhaut, J.F., 2020, July. Increasing the efficiency of Fog Nodes through of Priority-based Load Balancing. In 2020 IEEE Symposium on Computers and Communications (ISCC) (pp. 1-6). IEEE. CORE RANK=B.
- [17] Rehman, A.U., Ahmad, Z., Jehangiri, A.I., Ala'Anzy, M.A., Othman, M., Umar, A.I. and Ahmad, J., 2020. Dynamic energy efficient resource allocation strategy for load balancing in fog environment. IEEE Access, 8, pp.199829-199839, JCR factor= 3.367.
- [18] Wang, S., Zhao, T. and Pang, S., 2020. Task scheduling algorithm based on improved firework algorithm in fog computing. IEEE Access, 8, pp.32385-32394, JCR factor= 3.367.
- [19] Cheng, D., Zhou, X., Xu, Y., Liu, L. and Jiang, C., 2018. Deadline-aware MapReduce job scheduling with dynamic resource availability. IEEE transactions on parallel and distributed systems, 30(4), pp.814-826. JCR factor= 2.687.
- [20] Dolev, S., Florissi, P., Gudes, E., Sharma, S. and Singer, I., 2017. A survey on geographically distributed big-data processing using MapReduce. IEEE Transactions on Big Data, 5(1), pp.60-80, JCR factor= 3.344.



Impact Factor: 8.118



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



www.ijirset.com

Scan to save the contact details