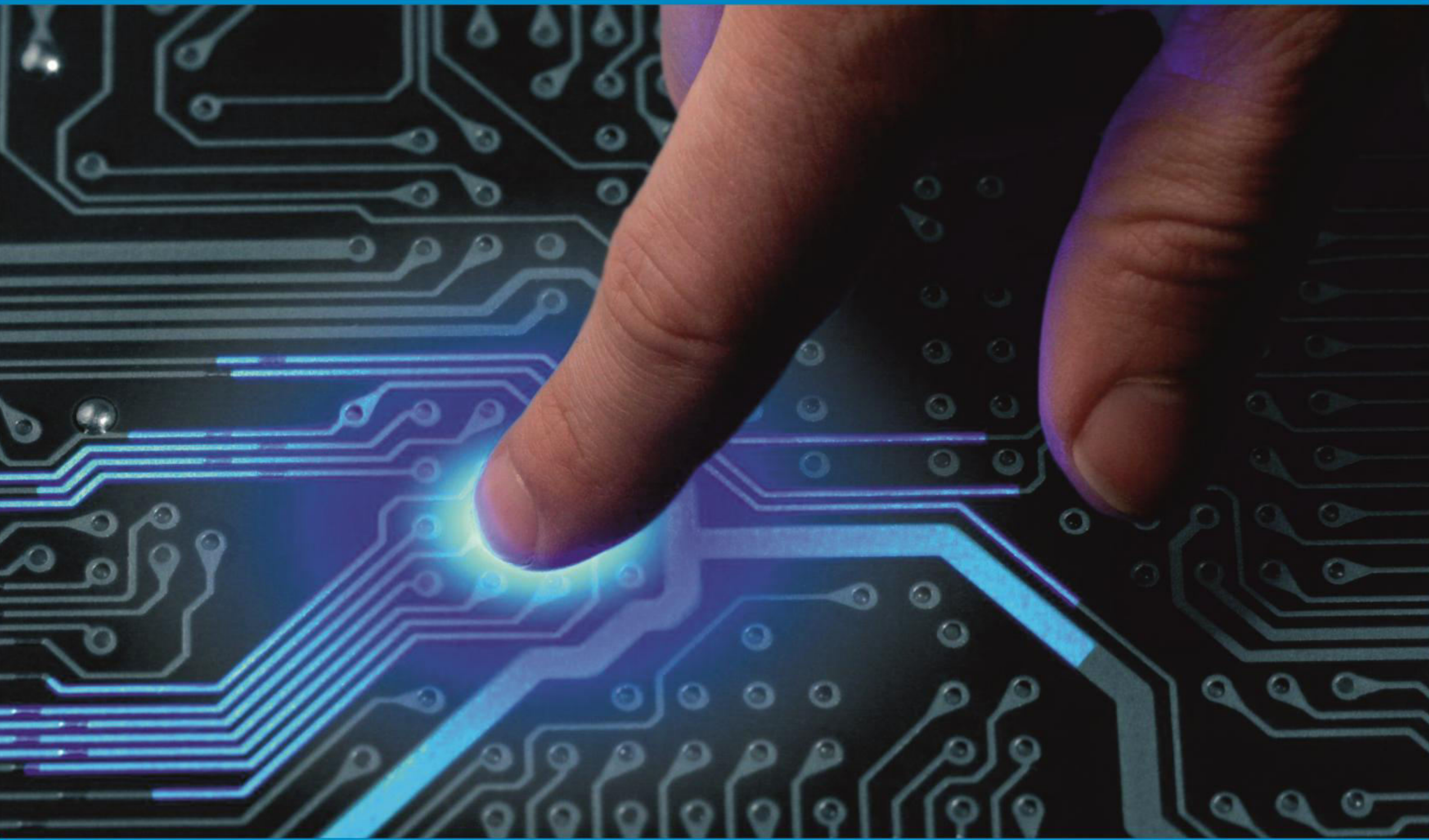
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 5, May 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.488

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905175 |

Implementation of Leveraging File Replication in Data-Intensive Clusters with Energy Adaptability

Sonali Bhalke[1], Deepti Dongare[2], Atharva Inamdar[3], Prasad Kadam[4], Pratik Kamble*[5]

Graduate Students, Department of IT, MMCOE, Pune University, India[1234]

Assistant Professor, Department of IT, MMCOE, Pune University, India[5]

ABSTRACT: File replication is a common strategy to improve data reliability and availability in large clusters. Reliability for each file under server failures based on the relationship between file reliability and replication factor when servers have a certain probability to fail. In the existing system more number of replica creates it require more energy consumption, time and storage space and sometime system fails and cannot send immediate response to the user request. In the proposed system energy efficient adaptive file replication system using bloom filtering to reduce latency time. In the propose system increasing number of file replica according to user request or user priority and vice versa. Propose strategies in reducing file read latency, replication time and power consumption in large cluster. If multiple user send request for one file then system create multiple replica according to priority and get immediate response to the user.

KEYWORDS: Data-Intensive Clusters, File Replication, Replica Placement, Energy-Efficient.

I. INTRODUCTION

In this framework some record make and store three copies for each document in haphazardly chosen servers crosswise over various racks. In any case, they disregard the document heterogeneity and server heterogeneity, which can be utilized to additionally improve information accessibility and record framework effectiveness. As records have heterogeneous popularities, an inflexible number of three imitations may not give prompt reaction to an extreme number of solicitations, So we propose a dynamic transmission rate change methodology to avoid potential incast blockage while duplicating a document to a server, a system mindful information hub determination procedure to decrease document read idleness, and a heap mindful reproduction support technique to rapidly make document copies under copy hub disappointments. Irregular choice of imitation goals requires keeping all servers dynamic to guarantee information accessibility, which anyway squanders control utilization. The irregular choice of copy goals does not think about goal transmission capacity and demand taking care of limit, arrange clogs may happen because of limit confinement of a few connections and server may end up over-burden by information demands.

II. RELATED WORK

In this paper, intelligent dynamic data replication algorithms are proposed based on bio-inspired algorithms with multi-objective (MO-PSO, and MO-ACO). The introduced strategies are used for both data replicas selection and placement in various datacenters. The introduced algorithms are tested using CloudSim. The performance of suggested techniques were evaluated against several replication strategies including, Adaptive Replica Dynamic Strategy (ARDS), Enhance Fast Spread (EFS), Genetic Algorithm (GA), Replica Selection and Placement (RSP), Popular File Replication First (PFRF), and Dynamic Cost-aware Re-replication and Re-balancing Strategy (DCR2S). The experimental results illustrate that MOPSO reaches improved data replication compared with other algorithms. Furthermore, MOACO realizes lower cost, less bandwidth consumption, and higher data availability compared with other techniques.[1]

“Cost Optimization for Dynamic Replication and Migration of Data in Cloud Data Centers,” To minimize the cost of data placement for applications with time-varying workloads, developers must optimally exploit the price difference between storage and network services across multiple CSPs. To achieve this goal, we designed algorithms with full and partial future workload information. We first introduced an optimal offline algorithm to minimize the cost

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905175 |

of storage, Put, Get, and potential migration, while satisfying eventual consistency and latency. Due to the high time complexity of this algorithm coupled with possibly unavailable full knowledge of the future workload, we proposed two online algorithms with provable performance guarantees. One is deterministic with the competitive ratio of $2\gamma -1$, where γ is the ratio of residential cost in the most expensive data center to the cheapest one either in storage or network price. The other one is randomized with the competitive ratio of $(1+\gamma) /\omega$, where ω is the size of available look-ahead windows of the future workload. Large scale simulations driven by a synthetic workload based on the Facebook workload indicate that the cost savings can be expected using the proposed algorithms under the prevailing Amazon's, Microsoft's and Google's cloud storage services prices.[2].

"Cost Optimization Algorithms for Hot and Cool Tiers Cloud Storage Services," Reducing the operational cost is one of the main driver behind migrating data to cloud data stores. This is becoming challenging particularly for tiered storage offerings with differentiated pricing models. Recently Microsoft Azure started offering hot and cool tiers storage services with opposing storage and access pricing. Storing objects in one tier all the time is not cost-effective, hence transfer of objects between tiers is required to reduce cost. Optimizing the storage service cots requires selecting the optimal placement of objects in the most appropriate storage tier. To this purpose, authors proposed an offline optimal algorithm which determines the optimal tier to use for a given sequence of read and write requests. As demonstrated in the experimental evaluation, the proposed algorithms can yield significant cost savings compared to storing data in the hot tier all the time especially for data centers with considerable pricing differences between storage tiers.[3]

An Energy-Efficient Adaptive File Replication System (EAFR), which incorporates three components. It is adaptive to time-varying file popularities to achieve a good tradeoff between data availability and efficiency. Higher popularity of a file leads to more replicas and vice versa. Also, to achieve energy efficiency, servers are classified into hot servers and cold servers with different energy consumption, and cold files are stored in cold servers. EAFR then selects a server with sufficient capacity (including network bandwidth and capacity) to hold a replica. To further improve the performance of EAFR, we propose a dynamic transmission rate adjustment strategy to prevent potential incast congestion when replicating a file to a server, a network aware data node selection strategy to reduce file read latency, and a load-aware replica maintenance strategy to quickly create file replicas under replica node failures. Experimental results on a real-world cluster show the effectiveness of EAFR and proposed strategies in reducing file read latency, replication time, and power consumption in large clusters.[4]

In paper[5], The arranged three authentic key swaps over conventions for parallel system record framework (pNFS). Authors strategies present three alluring remunerations over the reachable Kerberos based pNFS method. Essential, the metadata server executing our strategies has much subordinate outstanding burden than that of the Kerberos based advance toward. Consequent, two techniques make accessible frontward privacy: one is not entirely frontward ensured [with profound respect to complex gatherings inside an event era], in the meantime as the extra is totally forward secured. Next, they have expected a strategy which not just make accessible ahead classification, other than is too escrowing gratis.

III. PROPOSED METHODOLOGY

A. *Design Issues:*

Files are replicated and stored on heterogenous clusters. Due to high demand of request, number of replicas created are more due to which more energy is consumed and time-storage are wasted. This allows system to fail and cannot send immediate response to user. System proposes a reliable efficient energy saving file replication system wherein files are only replicated when high user request are done and depleted when there is no demand thus saving space on clusters. Files are stored on clusters according to demand of data thus conserving energy of clusters, it selects clusters on their capacity to store replicas.

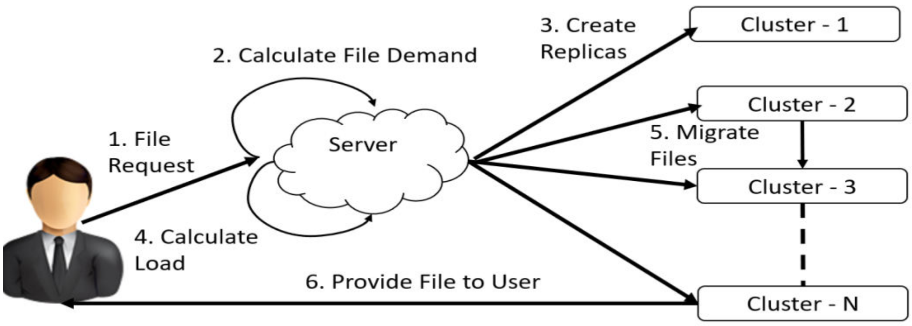
The system is designed with N number of clusters,First when user requests for file, the computing server calculates file demand of particular files and checks whether it is hot file depending on popularity. File is found on nearest cluster and send back to user.while uploading a file, replicas are created of file. In case of failure the files are migrated to different server. Computing server calculates the load to avoid congestion for file request..

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905175 |

B. *System Architecture:*



IV. PROPOSED ALGORITHM

A. *Algorithm 1. Pseudo-Code of Replication of Files*

```
Determine the popularity of file fi
        If  H(fi)=1:     //create one replica
Select hsj from the hot server pool;place replica in hsi
End if
If C(fi)=1: //reduce number of replica by one
 If number of replicas ri>r
Select sj
Delete  the replica of fi in hsj
Else
If more than € replicas fi  are stored in HS
Migrate one replica of fi from hsi  to  ssk
If equation (8) is satisfied for ssk
Ssk turns into a cold server
      End if
     End if
End if
```

B. *Algorithm 2. Pseudo-Code of Dynamic Transmission Rate Adjustment*

```
Input: set of files F,counter=1;
Output: Source server  and destination server for files in F;
For esch fi € F do:
 Order hot servers and standby servers storing fi's replicas
By their remaining service capacities Si=(s1,s2,.......,sp)
 Select s1 as the source server
If s1 is empty
 Choose a cold server with the maximum available service capacity as the asource server
End if
While true // select a destination server from DS
Choose  server dsk  with index equalled to counter
If  Csi - Σ                    //enough storage
Select dsk as the destination server
Break
Else
Increase counter by 1
End if
End while
```

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905175 |

Record fi's source server and destination server
End for

V. RESULTS

Fig.1.Home Page

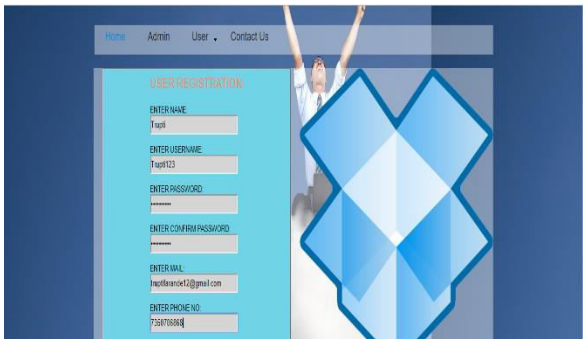


Fig. 2. User Registration

Fig. 3. User Login

Fig 4. Admin Login

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905175 |

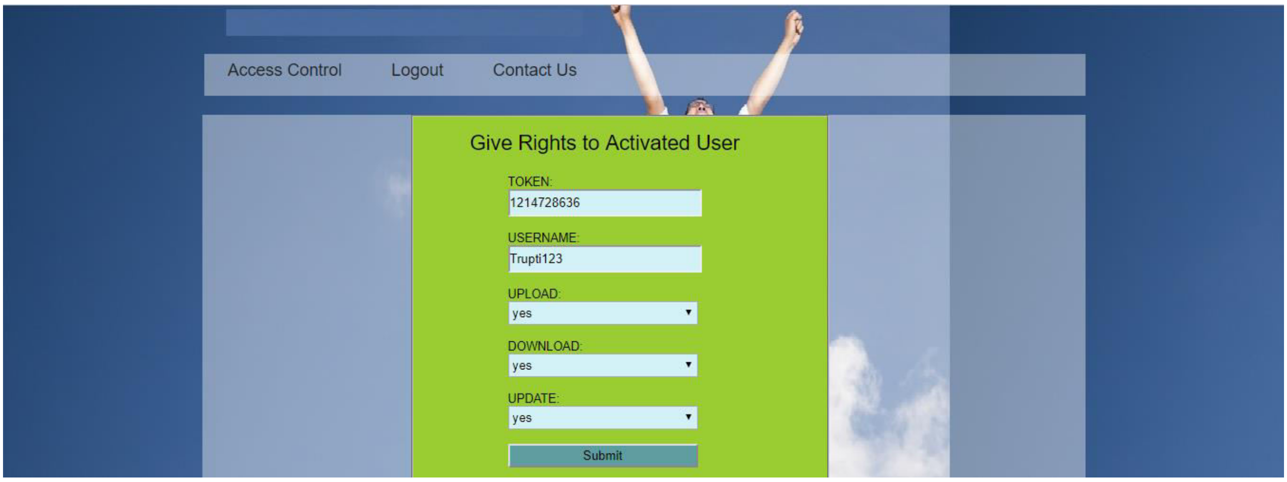


Fig 5. Rights to activated User

Fig 6. Token Activation

Fig 7. File Downloads

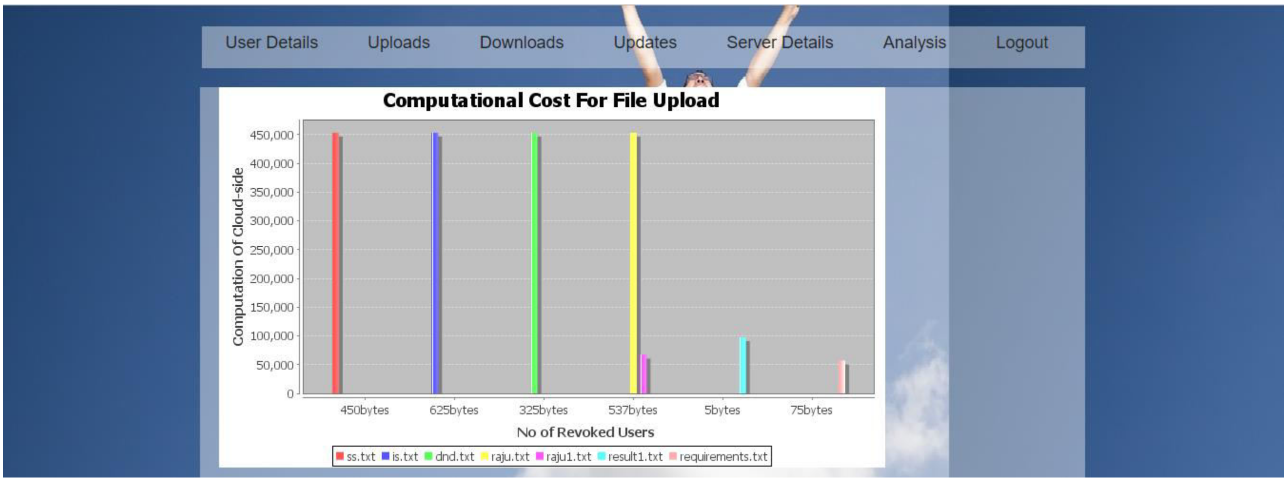


Fig 8. Computational Cost

VI. CONCLUSION AND FUTURE WORK

The popularity of data-intensive clusters places demands for file systems such as short file read latency and low power consumption. In thissystem a dynamic transmission rate adjustment strategy to prevent potential incast congestion when replicating a file to a server, a network awaredata node selection strategy to reduce file read latency, and a load-awarereplica maintenance strategy to quickly create file replicas under replica nodefailures. Experimental results from a real-world large cluster show the effectiveness of EAFR and the proposed strategies in meeting the demands of filesystems in large clusters.

In future system can be extend by adding server failure tolerance also toincrease system file security. The data partitioning algorithms is used for better performance and security. Therefore, a server's recent transmission speedcan be used to predict the transmission speed in the near future. EAFR doesnot need to look into the link utilization and monitor the network congestionstatus when allocating a new file replica. It selects the replica destinationbased on the transmission speed of recent files In the future, we will studyincreasing data locality in replica placement, and determining the optimalnumber of cold servers to maximize energy saving without compromising thefile read efficiency

REFERENCES

[1] Awad, R. Salem, H. Abdelkader and M. A. Salam, "A Novel Intelligent Approach for Dynamic Data Replication in Cloud Environment," in IEEE Access, vol. 9, pp. 40240-40254, 2021, doi: 10.1109/ACCESS.2021.3064917.

[2] Y. Mansouri, A. N. Toosi and R. Buyya, "Cost Optimization for Dynamic Replication and Migration of Data in Cloud Data Centers," in IEEE Transactions on Cloud Computing, vol. 7, no. 3,pp. 705-718, 1 July-Sept. 2019, doi: 10.1109/TCC.2017.2659728.

[3] Y. Mansouri and A. Erradi, "Cost Optimization Algorithms for Hot and Cool Tiers Cloud Storage Services," 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), 2018, pp. 622-629, doi: 10.1109/CLOUD.2018.00086.

[4] Yuhua Lin and Haiying Shen, "EAFR: An EnergyEfficient Adaptive File Replication System in Data-Intensive Clusters", IEEE Transactions On Parallel And Distributed Systems, Vol. 28, No. 4, April 2017.

[5] M. Rengasamy , "Energetic Key Exchange Protocol Authentication for Similar Network File Systems" October 2015

[6] Sankalp Mitra , Suchit Bande, " Efficient FP Growth using Hadoop - (Improved Parallel FP-Growth) " July 2014

[7] Karakoyunlu, C., Kimpe, D., Carns, P., Harms, K., Ross, R., & Ward, L. (2013, September). Toward a unified object storage foundation for scalable storage systems. In 2013 IEEE International Conference on Cluster Computing (CLUSTER) (pp. 1-8). IEEE.

[8] Carns, P., Lang, S., Ross, R., Vilayannur, M., Kunkel, J., & Ludwig, T. (2009, May). Small-file access in parallel file systems. In 2009 IEEE International Symposium on Parallel & Distributed Processing (pp. 1-11). IEEE.

[9] Konstantin Shvachko, HairongKuang , " The Hadoop Distributed File System " 2010 IEEE

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905175 |

[10] Sagar S. Lad P,NaveenKumarP, " Comparison study on Hadoop's HDFS with Lustre File System " November 2015

[11] Z. Cheng, et al., “ERMS: An elastic replication management system for HDFS,” in Proc. CLUSTER Workshops, 2012, pp. 32–40.

[12] Q. Chen, J. Yao, and Z. Xiao, “Libra: Lightweight data skew mitigation in MapReduce,” IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 9, pp. 2520–2533, Sep. 2014.

[13] Verma, G. Dasgupta, T. Nayak, P. De, and R. Kothari, “Server workload analysis for power minimization using consolidation,” in Proc. Conf. USENIX Annu. Tech. Conf., 2009, pp. 28–28.

[14] K. Shvachko, K. Hairong, S. Radia, and R. Chansler, “The hadoop distributed file system,” in Proc. IEEE 26th Symp. Mass Storage Syst. Technol., 2010, pp. 1–10.