VAWKUM Transactions on Computer Sciences
http://vfast.org/journals/index.php/VTCS@ 2019, ISSN(e):2308-8168, ISSN(p): 2411-6335
Volume 16, Number 1, January-April , 2019 pp. **01--08**

SEVERAL ADOPTIVE REPLICA SYNCHRONISATION APPROACHES IN DISTRIBUTED FILE SYSTEMS

SAHER ATHER[1], HOORIA MUSLIH-UD-DIN[1], MUHAMMAD NADEEL[2],
MUHAMMAD AHSAN[2], BILAL HASSAN[2]
[1]Department of Computer Sciences, National University of Computer and Emerging Sciences (FAST-NU), Faisalabad, Pakistan
[2]Department of Software Engineering, University of Management and Technology, Lahore, Pakistan
Email: { ather.saher@gmail.com[1], l164197@lhr.nu.edu.pk[2], {muhammad.nabeel[3], muhammadahsan[4], bilal.hassan[5]}@umt.edu.pk }

ABSTRACT. *This survey paper explained the different approaches of synchronization of replicas of files placed on distributed systems. The survey tells some older and latest techniques of synchronization. Some techniques are by the interference of metadata servers and some are without any intrusion of MDS. In former technique SS storage servers are used for synchronization among replicas. To maximize the performance, scalability and reliability CEPH is a distributed file system. It makes distinction between meta data and data management by object storage file system run on object file systems. Excellent I/O and metadata management is done on CEPH. Commodity servers and disks are used for multitier distributed systems. Performance reliability, I/O rate, workload in writes operations and less overhead in synchronization are the main focus while synchronization of replicas. Hadoop and Google file system are the distributive file systems. Hadoop ensures the better input and output performance with minimal synchronization in replicas, data intensive applications and provides fault tolerance. Some strategies are used for data intensive applications. Parallel file system is type of distributed file system. Analysis enforces the best performance on small and large input output requests. Pattern direct and layout replication technique is one of the most optimized techniques for parallel file system. Data access performance, reliability, data consistency, centralized synchronization, less workload, less overhead is the main focus of all the techniques. Some other file systems like SOFA and frangipani do focus on data consistency and reduce of bandwidth.*
Keywords*:* Adaptive Replica Synchronization; File Systems; Ceph; Gmei; MDS.

1. Introduction. The distributed file system in which files are places in distributed environment improves performance of I/O and reliability of system. In older systems the metadata server was accessed by client and then the storage server. Metadata server was responsible to do all management of properties of systems. All the information of layout and properties of file system were managed by metadata server. Storage server was responsible for managing file of data, I/O operations after obtaining information from MDS. Other approaches are [2], geo distributed data centers [3], Ceph [4], Global Distributed file system [5], lustre [6], SOFA [27] are some other file systems in distributed environment. File striping units [2] are chunks of files. The chunks are replicated over storage server's results in reliability and satisfactory access of concurrent data and also increase bandwidth to transfer data.

Generally it was responsibility of metadata server to manage information of all chunks of data. The replica synchronization occur in a way that if there occur any update in a chunk of data, disk won't save new written data as it become to full the storage server because all other threads wait for synchronization to complete which was started by one storage server. To solve this parallel file system, GMEI was presented which support lazy and replica among storage servers by which MDS got free from synchronization of replica. The chunk list which was replicated had the information about all chunk replicas. The information includes locations and states. Then GMEI store the duplicates on storage servers. Write latency was shortened and write data throughput enhancement was the advantage taken from the lazy replica synchronization [7]. Different contributions were made by proposed mechanism are:

- Replication Chunk List Storage Server: It's make sure replica consistenny when ever any unpdate is occoured without intrusion of MDS and contain the data of replicas. Replica synchronization accept read requests with minumum overhead whereas conducting synchroniazation of replicas delegate by storage servers. These storage servers had information of updated chunks [7].

- Lazy Adaptive Synchronization: In this technique the replica synchronization is not done immediately at the time when any update is made becaue there is a possibility that it will be modified or updated in near future. This reduces the write latency and improves I/O performance. The daily replicas synchronization is started by MDS in off peak hours [7]. In Google file system the performance reliability and scalability are the important factors. This file system is built by client machines having thousands of storage mechanisms.

Problems like application errors, operating system errors, the failure of disks, memory etc are caused. But by constant monitoring and fault tolerance these problems become essential to system. Multi GB files of huge standards are supported. Most files are changed by appending data with older instead of overwriting it. From its design view the API of file system make it flexible which increase its benefits [8]. In Hadoop file system when any replica is updated or modified replica synchronization is triggered by MDS and storage server only manage and handle data management. Replica is just a copy of original data. Replica synchronization is done to improve the scalability and performance of system. During synchronization it is make sure that data which is forward to one replica is received by relevant replica. Number of write operations tells that how many times synchronization will be occur which sometime exceeds and cross limits. To improve it metadata servers was used. [2]. Geo distributed data centers are distributed at different geographical locations and sites. Google has 3 data centers over different countries. High cost and data loss is an issue of replication of big data [3].

For cost effective and to reduce the data loss cluster geo- replication is utilized. The technique used to split the cluster into primary and backup tier this technique is known as tiered technique. In order to independent node failure, the recovery of data is make sure because two replicas are stored in primary tier while third one is stored on backup tier placed on some other site. Although third replica is read rarely so it can be placed on another site on some other location. The tired technique also makes sure that no data loss occurs. Minimum chances of data loss and low cost are two major issues solved. Nodes leave dynamically. Tiered replication is implemented on open source storage system which causes small overhead. Generally some file systems replicated their data on different machines to check their data loss. But geo replicates its data on different locations and check correlated failure. Two replicas are usually enough for protection against any node failure and third one is just in case two are not available and also increases the durability of system [8]. Equation 1 shows the rate of data loss. MTTF stands for mean time of failure node.

$$Rate\ of\ Loss = \frac{1}{MTTF} \qquad (1)$$

CEPH is distributed file system which separates data and metadata management ensures better scalability, performance and reliability. Separation is based on replacing allocation table with pseudo random data distribution running on localized data system designed for object storage devices. In ceph clients interacts with MDS and then MDS communicate with object storage devices to perform file I/O which results in improving of scalability. The systems with petabyte scale are large systems in which node failure are more normal than any exception. The algorithm ensures that quality and work load shift over constantly. Highly distributed metadata servers improve the scalability of metadata access [4].

In distributed file system, availability of a resource is a major factor to be checked. Replication is the avoidance of fault tolerance and enhancement of fault tolerance. Also the replica consistency is the issue covered by most of the efficient techniques. It is necessary that all replicas must have same data. Asynchronous technique works in a way that whenever any update or modification occurs in a file, all replicas are updated by server immediately [9]. Lustre is the distributed and high performance system that minimizes the availability and scalability problem in distributed file systems. It is shared disk file system usually available in Linux [6] [28].

2. Replica synconization techniques.

A. ***GMEI file system:*** Two approaches are adopted by GFS. One is lazy replica synchronization and other is version based update replay. The GFS replicates chunks of files and distribute it to the relevant storage servers. Unique chunk list is stored on storage server as a file. Dirty flags and other information of replicas show that if the replica is updated or not. Access count shows the number of requests for read operations to that replica. Another field called version field is added to the information of chunk list which tells that which request is last one. The ID against version field is generated by MDS which is unique. By this ID it is now possible for client to send I/O request to storage server. After modification and completion of update operation, version field is modified which shows the last request which updated the chunk targeted. The chunk list of chunk replicas adds replica consistency for clients. Location of all replicas is stored in chunk list and dirty flags and other labels. Access count tells the total count of read requests [7], [14]. Equation (2) is to find the frequency to access the chunk of replica.

$$\text{Freq}(Ri) = \frac{\Sigma\, n=0\ (\alpha f(t) \times 2\Delta(t))}{L\ (t)} \qquad (2)$$

Ri is the latest replica. L(t) is the total count of passed time segments. αf(t) is the number of visits.

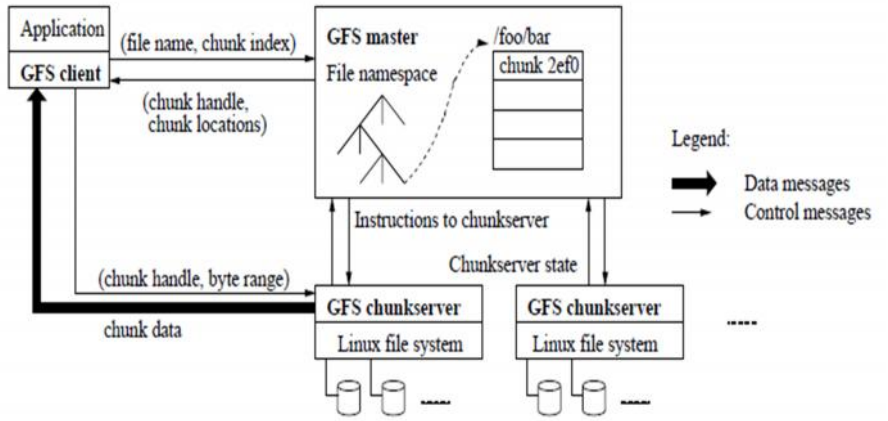

Figure.1: GFS File System [1].

B. ***Lazy replica:*** While write operations occurrence the updating in replicas are stopped that belong to same chunks until the write operations completes. Whenever ant write operation is occurred it first allows completing the operation and then broadcasting the request and notifying to update the chunk to storage server. After updating the chunk storage server notify the primary server and primary server reply back to client that write operation is completed. This technique benefits in way that no matter how many write operation occur in series, only one update operation is required to update the chunk list [7], [14].

C. ***Version based update replay:*** GFS adopts the technique to reduce write latency. Apart from its benefits, the chances of flush of primary storage server before it synchronized the replica and update them. Version Update Replay is introduced to solve above mentioned problem. To manage the number of cashed request, MDS tell storage server to perform synchronization of replica results in the removal of cashed requests. If ant of the storage server failed to respond back and client could not find required replica then to avoid the crash and loss, MDS restore and recover the lost update. And MDS locates relevant update operations [7], [14]. The I/O data rate can be calculated by Equation (3).

$$\text{I/O Rate} = \frac{(\text{Number read} + \text{Number write}) * 64\ \text{KB}}{\text{Time write} + \text{Time read}} \qquad (3)$$

D. ***In Hadoop file system:*** The design of Hadoop is capable of storing very large dataset and streaming is on very large bandwidth [12]. To improve performance of distributed file system, bandwidth and input output throughput replica synchronization is necessary. In this approach MDS is responsible for managing the information of all replicas and breaks the data into chunks. The storage server cannot take number of read operations. The replicas are performed to comply with read operations. Big data, metadata, storage servers and metadata servers are the part of hadoop file system. The user communication starts with distributed server and then it accesses the big data from server. Update query is made on big data in distributed storage systems then store that into storage system. Chunk list replication having information of chunks of files are replicated on storage servers. And update is made for replica synchronization. This technique increases the performance and reduces bandwidth of communication. [10].

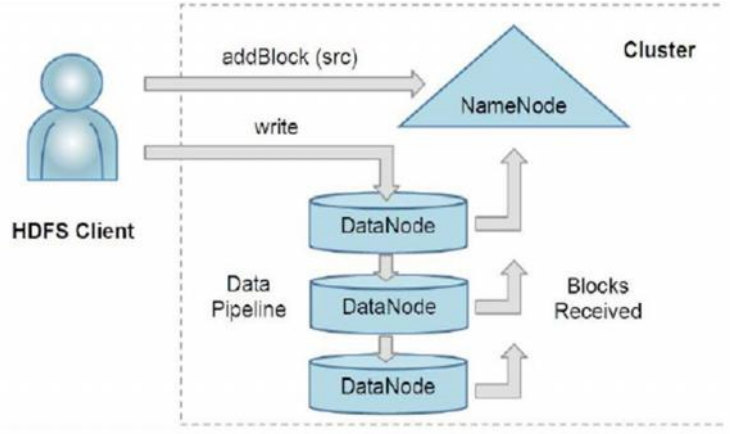

Figure.2: Hadoop File System [7].

E. ***Asynchronous replication:*** The Distributed file system use network to communicate with servers results in improvement of scalability and reliability. But challenges to DFS are partial failure and concurrency. Concurrency problem occurs when more than one node are executed at once make problem for data to be consistent. If one client is accessing a file and updating it and then 2nd client access the same file and update it. It can cause data inconsistency. Replication is one of the solutions. It allows multiple users to access its own copy and make updates. But there come another problem that is to keep replicas consistent. Changes made on one replica should be made to all. So that any client gets it for read operation may get its valid copy. In case of write operation, all operations and updates are made on primary copy. And then all other copies are updated. But there comes a limit. Replication is reduced by asynchronous technique in which when there is a write operation it is done on primary copy and new version number is added to dirty bi. But read operation depends on version or dirty bit. Value one represent data is invalid and zero tells it is valid. This technique ensures the replica synchronization efficiently and most importantly the consistency is achieved [9].

F. ***Replica synchronization in CEPH:*** CEPH include placement groups which use hash function to map objects. Placement groups then are assigned to object storage devices which use hashing which is called control duplication with scalable hashing. CRUSH requires placement group and a cluster map to locate an object. CRUSH solves the problem of data distribution and data location. Coming towards the replication and its synchronization, placement groups are the replicated data. Object placement group is stored in object storage device. The write request sent by client is first received by this OSD. It assigns new version number and forward to replicas in OSDs. When all the replica are updated they reply back to primary and then primary reply to client who initiated this update operation. That's how the serialization occurs between replicas. The advantage of this technique is less bandwidth required to client, instead the load of bandwidth transfer to the OSD internal network. It also increases the replica consistency and reliability [4] [29].

G. ***In luster:*** Lustre files system run on hardware. Disks which are based on object are used for metadata servers for storing file system and used for storage. MDS has record of failed replicated file. The coherency of files is maintained by lustre by using metadata locking. Replication is done on server side which is on metadata. Cashing is done by client side [28].

H. ***In Google file system:*** GFS is highly distributed. Hundreds of its chunk servers are distributed among machine racks. It is proprietary DFS used by Google itself. The read, write and update mechanism are defined. Network switches are used for communication between two machines [1]. Google file system includes in data intensive cloud applications runs on cluster [13]. Comet is another data intensive distributed system which is batched streamed. It streams bulk of data for effective optimization. [19].

3. Replication techniques for distributed file systems. A type of distributes file system includes parallel file system. In this system multiple files are distributed over servers. It is included in high performance computing. The tasks are running in parallel and concurrency is the factor which is covered in this system. Pattern direct and layout aware replication technique is one of the replication techniques which is for parallel input output systems. In this technique the pattern of data is accessed. And then using that pattern information the data is replicated. Five key aspects are involved space localization, size, temporal info, iteration, operations on input output.. The synchronization among these replicas occurred in a way that local data access the global data. Each replica contains an object of that data. Original data is not stored on each server. The data request is made on three factors file that is original, offset and size. Metadata which has a catalog of replication redirect the module, and translate offset and the request is fulfilled [16]. To improve high performance of input and output in parallel file systems file size capacity is increased to MB files. The technique of parallel file aggregation makes it possible. This technique was introduced to increase bandwidth [20]. GPFS general parallel file system supports chunk replication. Multiple data is stored on different storage servers. The synchronization among replicas is done in a way that the write update is made synchronously on all replicas. The track of each updated chunk is done by write operation is save by primary storage servers. And then these servers update all replicas on storage servers and update it before that write operation is completed [7].

SOFA is search oriented distributed file system. File or any data is stored on different physical nodes on different locations. It makes sure that transparency occurs between replicas and any case of failure node. Cashing and replication mechanism is used to make system a fault tolerated and consistent. Replication is similar to GFS with little change of pipelining. Due to passive replication, communication overhead among client and SOFA is reduced. Communication is by remote procedural call. The system consists of a master server, client server, chunk server and file server. Read and write operations are never started by master server. Chunk is replicated over many servers [27].

GPFS is extremely scalable shared disk. It has cluster of nodes and the files are on disk the files are stored on disk connected by switches. Logging is maintained by this file system which helps in consistency. If any update operation was failed the logs maintained and recovery can be made on the its base and reapply those changes. A concept of locking is also applied. Read and write locks are applied while operations. Synchronization updates are made on servers by appropriate locks [24]. When it comes to the consistent and redundant data, Redundant Array make sure of independent Disks RAID that data and replicated data stored on disks are consistence and redundant. It made tradeoff balanced between performance and reliability and availability [7]. It is always be able to stand with the case of failure. Redundant Array of independent Disk AFRAID's write operation include two things.one is to update the replica of original copy and make the parity bit updated. Also set a bit per stripe and make updates to all other parities. It waits till the array is idle. Disk reliability and data redundancy are the main qualities of AFRAID. It reduces failure node rate too [23].

An example of scalable distributed system is Frangipani. For synchronization it uses locking mechanism. It uses strong read and write locks. When more than one server is doing update operations, synchronization is done for data consistency. Read write locks are for synchronization. Read lock is for accessing any data from server. Write lock allow both read and write. A buffer pool is used for data cashed for file server [29].Grid computing is a kind of distributed computing. That also has the advantages of distribute files over different geographical locations. Dynamic resource availability, replication, data consistency are the key factors that improve reliability of the source. Replication reduces bandwidth. In this technique before calculating replica the risk rate is being calculated that optimized utility. It also reduces bandwidth. Replicas are managed statically or dynamically. Offline process is done by static replicas. In which replicas are placed at design time using systems snapshot. Before a system comes online sites are chosen, and it will allow storing replica even if an changes are made on system. BHR is a technique to store

copy of data, managing data file. Replica location service and metadata server are two components in management system. BHR is an algorithm which is region based for dynamic replication [18].

High performance computing includes parallel file system which accesses so many sized files. Gmei is on previous parallel file system which was prototyped [21]. Internet of thing is an emerging trend. Nodes of WSN collect data. The replicas which are distributed on other nodes with same data are broadcasted that is logical [22]. Some architecture of distributed file systems is designed in a way that they are cost efficient, reliable. Their I/O input output is very high. Storage are durable by which data to be restored on nodes that are volatile. But there always remain tradeoffs. By improving one factor, other factor can be disturbed [25]. Research shows that techniques for synchronization of replicas are making sure that data remain consistent. And synchronization helps to decrease bandwidth and there is least case of data loss.

Table.1: Replica synchronization of different techniques of distributed file systems.

Factors	**Distributed file Systems**					
	GFS	**CEPH**	**Hadoop**	**Lustre**	**SOFA**	**GMEI**
Replication technique	Asynchronous replication	Placement groups, Hashing function, OSDs	Asynchronous replication, chunk replication	Metadata locking, RAID	Master server and replica daemon	Lazy replica, Version based update replay, chunk replication
Consistency	Yes	Yes	Yes	Yes	Yes	Yes
Synchronization	Write-one Read-many	OSD , CRUSH	MDS	Object based disks, MDS	Read write locks	Storage server
Communication	RCP/TCP	TCP	RCP/TCP/UDP	TCP	UDP	TCP

4. Comparison between distributes file systems. Table [1] compares the characteristics of various distributed files systems. There are several replication techniques for different distributed files systems. Consistency exists in each file system. For synchronization, there are different algorithms which are used in all file systems. GFS use write-one-read-many algo, CEPH use OSD. Hadoop use MDS approach. Lustre use object based disks algo and read write locks. Different protocols are use for communication RCP/TCP use by GFS, TCP use CEPH. Hadoop use RCP, TCP, UDP. SOFA used by UDP and TCP used by GMEI. Detail comparison between different distributed file systems are provided in table [1].

5. Conclusion. This survey paper has briefly explained the synchronization techniques of replicas in distributed file system. Distributed file system is one of the vast storage systems which have been widely used. The distributed file systems include a GMEI file systems which includes lazy and adaptive synchronization mechanism. Data structure adopted by GMEI is chunk list. Chunk list is distributed over storage servers. Data consistency is made sure by storage server which mainly helps in synchronization of replicas. Crash is removed by updating replicas on all storage servers before primary server. I/O operations occurred are guided by MDS. GMEI and its hybrid mechanism technique are quite effective. Some other files system like Google file system, performance, reliability and scalability are important features. By constant monitoring issues like memory and disk failure are removed.

CEPH is another distributed file system. Data and metadata management is separated to increase the performance. Chunk list is involved in updating read write operations in replicas. The main feature that makes it different is that only one operation is required to update chunk list. CEPH works with CRUSH algorithm and placement groups. A distributed system hadoop is capable of storing a large number of data. Unlike Gmei,

Hadoop use metadata server to manage all information of replicas. All the communication done by user is with metadata server. But chunk list is still managed by storage server. This technique helps in reducing bandwidth. Asynchronous replication techniques make sure data consistency by read write operation technique in a way that updates are made on only primary copy and then all other replicas are updated. Object storage device. Metadata locking is involved while making read write operations. SOFA is search oriented system. Replication is based on GFS. Concluding all DFS main factors that are data consistency, fault tolerance are almost achieved in all techniques.

REFERENCES

[1] Ghemawat, S., Gobioff, H., & Leung, S. T. (2003). The Google file system (Vol. 37, No. 5, pp. 29-43). ACM.

[2] Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010, May). The hadoop distributed file system. In Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on (pp. 1-10). Ieee.

[3] Gu, L., Zeng, D., Li, P., & Guo, S. (2014). Cost minimization for big data processing in geo-distributed data centers. IEEE transactions on Emerging topics in Computing, 2(3), 314-323.

[4] Weil, S. A., Brandt, S. A., Miller, E. L., Long, D. D., & Maltzahn, C. (2006, November). Ceph: A scalable, high-performance distributed file system. In Proceedings of the 7th symposium on Operating systems design and implementation(pp. 307-320). USENIX Association.

[5] Ford, D., Labelle, F., Popovici, F. I., Stokely, M., Truong, V. A., Barroso, L., ... & Quinlan, S. (2010, October). Availability in Globally Distributed Storage Systems. In Osdi (Vol. 10, pp. 1-7).

[6] Salunkhe, R., Kadam, A. D., Jayakumar, N., & Joshi, S. (2016, March). Luster a scalable architecture file system: A research implementation on active storage array framework with Luster file system. In Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on (pp. 1073-1081). IEEE.

[7] Liao, J., Li, L., Chen, H., & Liu, X. (2015). Adaptive replica synchronization for distributed file systems. IEEE Systems Journal, 9(3), 865-877.

[8] Cidon, A. (2014). Data Durability in Cloud Storage Systems(Doctoral dissertation, Stanford University).

[9] Jain, N., Dahlin, M., & Tewari, R. (2005, December). Taper: Tiered approach for eliminating redundancy in replica synchronization. In Proceedings of the 4th conference on USENIX Conference on File and Storage Technologies-Volume 4 (pp. 21-21). USENIX Association.

[10] Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010, May). The hadoop distributed file system. In Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on (pp. 1-10). Ieee.

[11] Kumari, J., Biswas, T., & Vuppala, S. (2018, July). Enhancing Replica Synchronization in Hadoop Distributed File System. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.

[12] Wang, F., Qiu, J., Yang, J., Dong, B., Li, X., & Li, Y. (2009, November). Hadoop high availability through metadata replication. In Proceedings of the first international workshop on Cloud data management (pp. 37-44). ACM.

[13] Tantisiriroj, W., Son, S. W., Patil, S., Lang, S. J., Gibson, G., & Ross, R. B. (2011, November). On the duality of data-intensive file system design: reconciling HDFS and PVFS. In Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis(p. 67). ACM.

[14] Liao, J., Li, L., Chen, H., & Liu, X. (2015). Adaptive replica synchronization for distributed file systems. IEEE Systems Journal, 9(3), 865-877.

[15] Nieuwejaar, N., & Kotz, D. (1997). The Galley parallel file system. Parallel Computing, 23(4-5), 447-476.

[16] Yin, Y., Li, J., He, J., Sun, X. H., & Thakur, R. (2013, May). Pattern-direct and layout-aware replication scheme for parallel i/o systems. In 2013 IEEE 27th International Symposium on Parallel and Distributed Processing (pp. 345-356). IEEE.

[17] Gonizzi, P., Ferrari, G., Gay, V., & Leguay, J. (2015). Data dissemination scheme for distributed storage for IoT observation systems at large scale. Information Fusion, 22, 16-25.

[18] Sashi, K., & Thanamani, A. S. (2011). Dynamic replication in a data grid using a modified BHR region based algorithm. Future Generation Computer Systems, 27(2), 202-210.

[19] Tantisiriroj, W., Son, S. W., Patil, S., Lang, S. J., Gibson, G., & Ross, R. B. (2011, November). On the

duality of data-intensive file system design: reconciling HDFS and PVFS. In Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis(p. 67). ACM.

[20] Kato, J., & Ishikawa, Y. (2011, May). Design and implementation of parallel file aggregation mechanism. In Proceedings of the 1st International Workshop on Runtime and Operating Systems for Supercomputers (pp. 33-40). ACM.

[21] Silberstein, M., Ganesh, L., Wang, Y., Alvisi, L., & Dahlin, M. (2014, June). Lazy means smart: Reducing repair bandwidth costs in erasure-coded distributed storage. In Proceedings of International Conference on Systems and Storage (pp. 1-7). ACM.

[22] Savage, S., & Wilkes, J. (1996). AFRAID—a frequently redundant array of independent disks. parity, 2, 5.

[23] Schmuck, F. B., & Haskin, R. L. (2002, January). GPFS: A Shared-Disk File System for Large Computing Clusters. In FAST (Vol. 2, No. 19).

[24] Gharaibeh, A., & Ripeanu, M. (2009, June). Exploring data reliability tradeoffs in replicated storage systems. In Proceedings of the 18th ACM international symposium on High performance distributed computing (pp. 217-226). ACM.

[25] Salunkhe, R., Kadam, A. D., Jayakumar, N., & Joshi, S. (2016, March). Luster a scalable architecture file system: A research implementation on active storage array framework with Luster file system. In Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on (pp. 1073-1081). IEEE.

[26] Choi, E., Thanh, T. D., Upadhyaya, B., Azimov, F., Long, L. H., Phuong, T., ... & Kim, P. (2008). SoFA: A Distributed File System for Search-Oriented Systems. 한국시뮬레이션학회논문지, 17(4), 229-239.

[27] Depardon, B., Le Mahec, G., & Séguin, C. (2013). Analysis of six distributed file systems.

[28] Thekkath, C. A., Mann, T., & Lee, E. K. (1997). Frangipani: A scalable distributed file system (Vol. 31, No. 5, pp. 224-237). ACM.

[29] Thanh, T. D., Mohan, S., Choi, E., Kim, S., & Kim, P. (2008, September). A taxonomy and survey on distributed file systems. In Networked Computing and Advanced Information Management, 2008. NCM'08. Fourth International Conference on (Vol. 1, pp. 144-149). IEEE.