Scientific Journal of Impact Factor (SJIF): 5.71

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 5, Issue 04, April -2018

A REVIEW OF ADAPTIVE REPLICATION MANAGEMENT IN HDFS BASED ON SUPERVISED LEARNING

*Vaishali pandey **Varsha Sharma

*(Department of SoI,T RGPV ,BHOPAL,INDIA)
**(Department of SoIT, RGPV ,BHOPAL,INDIA)

Abstract.- *Apache Hadoop is very popular now-a-days. Hadoop Distributed file system (HDFS) is that the Heart of Apache Hadoop that is reliable and extremely accessible. At the heart of Apache Hadoop, the Hadoop Distributed file system (HDFS) provides the dependability and high availability for computation by applying a static replication by default. This paper explains the dynamic approach to replicate data Files supported supervised Learning. Hadoop, an open source implementation of the MapReduce framework, has been wide used for process massive-scale information in parallel. Since Hadoop uses a distributed file system, referred to as HDFS, the data locality drawback usually happens) and this drawback leads to the decrease in performance.*

Keywords: *Replication, HDFS, Proactive Prediction, Optimization, Bayesian Learning, Gaussian Process.*

1. Introduction:

Hadoop is an open resource implementation of the MapReduce and includes a distributed file system (HDFS), wherever application information is often kept with replication. With replication, Hadoop provides high degrees of availability and fault-tolerance. Hadoop is additionally increasingly gaining quality and has verified to be scalable and of production quality by Facebook, Amazon, Last.fm, etc. In HDFS, information are split during a fixed size (e.g., 32MB, 64MB, and 128MB) and also the split information blocks (chunks) are distributed and kept in multiple information nodes with replication. Hadoop divides every MapReduce job into a group of tasks according to the number of data blocks.[2] Basically, the Hadoop scheduler assigns a task to a node storing the information block preferentially, however it should assigns a task to a node not storing the information often consistent with the Hadoop scheduling policy. Cloud computing could be a new computing paradigm that's gaining increased quality. It permits enterprise and individual users to enjoy flexible, on demand and top quality services like high volume information storage and process while not the requirement to invest on expensive infrastructure, platform or maintenance[2].

Along with the improvement of Hadoop, the Hadoop Distributed file systems (HDFS) have been introduce to produce the consistent, high-throughput access to comparable compute. Increasingly, it become an appropriate storage space frame for equivalent and spread process, particularly for MapReduce-type solution, that was initially developed by Google to agreement with the big data.

Evolution on vast information has formed trend in appliance along with resolution improvement to remove, process, and store valuable information because it emerge to affect new challenge [3]. During this region the Apache Hadoop is that the most famous parallel framework. Not only to achieve the high availability, Apache Hadoop is intended to observe and handle failures at the applying also on maintain the data consistency.

For rising the fault tolerance and reliableness furthermore as providing the high availability and high performance, HDFS is initially equipped with a mechanism to replicate 3 copies of each data file from time to time. As time goes by, this replication strategy consumes storage resource and adds further overhead to the system by creating replicas of less frequently accessed information. Moreover, though the speed of reading operation in HDFS could be improved by the offered information, the performance of writing operation suffers the aspect effect of over-synchronizing unpopular information.[6] Thus, it's reasonable to form an inference that the static replicating mechanism results the whole system in poorer performance than the benefit it contributes.

An adaptive replication management (ARM) system is intended to produce high availability for the information in HDFS via enhancing the info locality metric. As a result, the highly local offered information improves the performance of the Hadoop system. It’s value noting that the erasure code is applied to maintain the reliability. A complexness reduction technique for the prediction technique is projected in each hyper-parameter learning and training phases. [3]This projected technique significantly will increase the performance in terms of reaction rate for the replication strategy whereas still keeping the accuracy of the prediction.

2. Literature Review:

Dinh-Mao Buiet al. [1] "Adaptive Replication Management in HDFS based on Supervised Learning", In order to enhance the availability of HDFS by enhancing the information locality, author contribution focus on following points. First, authors tend to style the replication management system that is actually adaptive to the characteristic of the information access pattern. The approach not only pro-actively performs the replication within the predictive manner, however additionally maintains the reliability by applying the erasure coding approach. Second, authors propose a complexity reduction technique to resolve the performance issue of the prediction technique. In fact, this complexity reduction technique considerably accelerates the prediction method of the access potential estimation. Finally, authors implement that technique on a real cluster and verify the effectiveness of the planned approach. With a rigorous analysis on the characteristics of the file operations in HDFS, our individuality is to form an adaptive answer to advance the Hadoop system. For any development, some components of the source code developed to check our plan would be created accessible under the terms of the GNU general public license (GPL). Author said that in the replication area, there are two main methods: the proactive approach and the reactive one. For the proactive approach, the Scarlett solution [13] implements the probability as an observation and then calculates the replication scheme for each data file. The storage budget-limitation is also measured as an issue when distributing the replica. Although this solution follows a proactive approach instead of using thresholds, the access rate of the data file as well as the suitable placement for replicas is not discussed thoroughly.

Xindong You et al. [2] "An Energy-Effective Adaptive Replication Strategy in Cloud Storage System", in this paper, authors present an Energy-effective adaptive replication strategy (E2ARS), throughout that data partition mechanism, minimal replicas crucial model, replicas placement ways and additionally the adaptive gear-shifting mechanism are intricately designed to maximize the energy conservation. Mathematical analysis is completed totally to verify that our E2ARS scheme will save energy consumption once the system's work is light-weight or desired latency is loose. and additionally the simulation experiment results demonstrate that through our E2ARS scheme energy consumption are typically saved with Qos satisfied and information convenience secured once variable the arrival rate, desired response, replicas vary and similarity degrees. However, our E2ARS scheme obtains the effective energy performance by inducing some storage capability overhead. Our any work is targeted on minimizing the overhead whereas increasing the energy consumption conservation. a good deal of work has done on energy conservation for large-scale storage systems supported information management and DVS (Dynamic Voltage Scaling), that relies on caching[14,15], information placement, information migration [16,17], and information replication. These techniques try to prolong disk idle times, therefore on make it potential to put idle disks in low-power state, therefore saving energy.

Jungha Lee et al. [3] "Adaptive Data Replication Scheme Based on Access Count Prediction in Hadoop", in this paper, authors project an adaptive information Replication scheme supported Access count Prediction (ADRAP) in a very Hadoop framework to deal with the information locality drawback. Through prediction of access counts of information files using Lagrange's interpolation, it optimizes the replication issue per data file. Our adaptive information replication scheme determines whether or not it generates a new replica or it uses the loaded information as cache dynamically throughout runtime. Moreover, author offer a replica placement algorithmic rule to enhance information locality effectively. Performance evaluations show that our theme reduces the completion time of the map part by 9.6% on the average, compared with the default information replication setting in Hadoop. In terms of information locality, the number of map tasks with node locality is increased by 6.1%, whereas the number of map tasks with rack and rack-off locality is decreased by 45.6% and 56.5%, respectively. In DARE [18], the authors planned a dynamic information replication scheme supported access patterns of information blocks during runtime to enhance data vicinity. Note that the default Hadoop distribution provides the fixed information replication within the section of data storing. DARE permits to extend the information replication issue mechanically by replicating the information to the fetched node. However, removing the replicated information is performed once only the offered information storage is short. Thus, it's a limit to produce the optimized replication issue with information access patterns.

Julia Myint et al.[4] "Modeling a Load-adaptive Data Replication in Cloud Environments", Data replication is an important technique to reduce user waiting time, speeding up information access by providing users with totally different replicas of a similar service. To require advantage of those, authors propose an effective replication model to manage replication degree during which it takes failure rate and information access popularity into consideration. During this paper, authors quantify the effects of variations in workload (i.e information access rate) and initial system configuration (setting up the reproduction range and information access level) on cloud storage quality in terms of reliability and mean time to failure. The experimental results demonstrate that the projected model is ready to adapt the variable information access load and thus it may be additional efficient in cloud data storage. Among a large amount of researches in storage system for cloud computing, Google File storage system for cloud computing, Google File System (GFS) (Ghemawat et al., 2003) and Hadoop distributed file system (HDFS) (Borthakur, 2007) are widely used and most popular. Other cloud storage systems that use key-values mechanisms are Dynamo (Decandia et al., 2007), Pnuts (Cooper et al., 2008) and Cassandra (Lakshman and April, 2010).

Dinh-Mao Bui et al. [5] "Replication Management Framework for HDFS based on Prediction Technique", the main purpose of this analysis is to enhance the data locality metric by using the prediction technique. With rigorous analysis of the characteristics of file operation in HDFS, the uniqueness of our plan is to create an adaptive and effective resolution to increase the capability of huge information systems. For additional development, some components of the source code developed for testing our plan would be created offered below the terms of the GNU general public license (GPL). The Cost-effective Dynamic Replication Management (CDRM) [19] is a cost-effective framework for replication in cloud storage system. When the work changes, CDRM calculate the recognition of information file and confirm the situation within the cloud surroundings. However, this method follows a reactive model. As a result, by using threshold values, CDRM cannot adapt well to the speedy evolution in large-scale systems.

Pawar Abhijeet H et al. [6] "Adaptive Replication Management in HDFS based on Supervised Learning", Following points focuses on the contribution of paper to enhance the availability of HDFS to enhance the information vicinity. The planning of replication management system is formed first that is very a lot of adaptive with the information access pattern characteristics. It maintains the reliability of system using erasure code approach with pro-active performance of the replication in predictive fashion. Performance issue of the prediction technique is resolved using a complexness reduction methodology. This methodology accelerates the prediction method of access potential estimation. The execution methodology on real cluster verifies the effectiveness of the projected approach. The adaptive resolution for Hadoop system is that the key of a paper with abrasive analysis on the characteristics of the file operations in HDFS.

Matei Zaharia et al.[7] "Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling", As data-intensive cluster computing systems like MapReduce and dryad grow in quality, there's a robust ought to share clusters between users. A task tracker of a slave node is in charge of planning tasks within the node. A task tracker requests a task from a job tracker by sending a heartbeat message once it's an empty task slot. Whereas a task is current, it additionally sends a heartbeat message periodically and also the message includes info concerning the state of the node and therefore the status of tasks that the node executes. [10]Information node maintains data blocks keep within the local storage, and local disk data related to HDFS.

Kyumars Sheykh Esmaili et al.[8] "The CORE Storage Primitive: Cross-Object Redundancy for Efficient Data Repair & Access in Erasure Coded Storage", In this paper author demonstrated that some simple and standard techniques (and thus easy to implement and organically integrate) can provide significant data repair and access boost in erasure coded distributed storage systems. Authors studied our approach of introducing cross-object coding on top of normal erasure coding analytically, comparing it with both traditional MDS codes as well as very recently proposed Local Reconstruction Codes (used in Azure). The ideas were implemented (as the CORE storage primitive) and integrated organically with HDFSRAID, and benchmarked over a proprietary cluster and EC2. Analytical & numerical studies, as well as experiments with the real implementation all demonstrate the superior performance of CORE over state-of-the-art techniques for data reads and repairs. While naive solutions can be readily used, in future authors will like to explore the CORE code properties to achieve better performance also during data insertion/updates. The current evaluations are static, based on snapshots of the system state. Authors speculate that CORE's better repair properties will yield a system in a better state over time. Authors will thus carry out trace driven experiments to study the system's dynamics better. Author discuss Erasure codes have long been explored as a storage efficient alternative to replication for achieving faulttolerance [20] in the peer-to-peer (P2P) systems literature, and have led to numerous prototypes, e.g., OceanStore [21] and TotalRecall [22] to name a few. In recent years erasure codes have gained traction even in main-stream storage technologies such as RAID [23]. The ideas from RAID systems are in turn permeating to Cloud settings, and erasure codes have become an integral part of many proprietary file systems used in data-centers, as well as open-source variants.

Ganesh Ananthanarayanan et al.[9] "Scarlett: Copingwith Skewed Content Popularity inMapReduce Clusters", Analyzing production logs from Microso Bing's datacenters revealed a skew in popularity of les, making the current policy of uniform data replication sub-optimal. Machines containing popular data became bottlenecks, hampering the efficiency of MapReduce jobs. Authors proposed a system that replicates less according to their access patterns, ageing them with time. Using a real deployment and extensive simulations, authors demonstrated that Scarlett's replication improved data locality in two popular MapReduce frameworks (Dryad and Hadoop) and sped up jobs by 20.2%. Scarlett's guided replication used limited extra storage (less than 10%) and network resources (1%).

George Kousiouris et al.[10] "Enabling proactive data management in virtualized Hadoop clusters based on predicted data activity patterns", To require advantage of those, authors propose an effective replication model to manage replication degree during which it takes failure rate and information access popularity into consideration. During this paper, authors quantify the effects of variations in workload (i.e information access rate) and initial system configuration (setting up the reproduction range and information access level) on cloud storage quality in terms of reliability and mean time to failure. This tentative results show that the projected model is ready for the adaptation of the variable information access load and thus it may be additional efficient in cloud data storage. Among a large amount of researches in storage system for cloud computing [25].

3. Methodology:

3.1 Hadoop cluster

Hadoop is an open source software framework that supports data demanding distributed applications. [7] Hadoop was derived from Google's MapReduce and Google File System papers and was originally formed by Yahoo

The architecture of a Hadoop cluster may be divided into 2 layers as shown in Figure 1: MapReduce and HDFS (Hadoop Distributed File System). The MapReduce layer maintains MapReduce jobs and their tasks, and also the HDFS layer is liable for storing and managing information blocks and their metadata.[10]

A job tracker within the master node splits a MapReduce job into many tasks and also the split tasks are scheduled to task trackers by the job tracker.

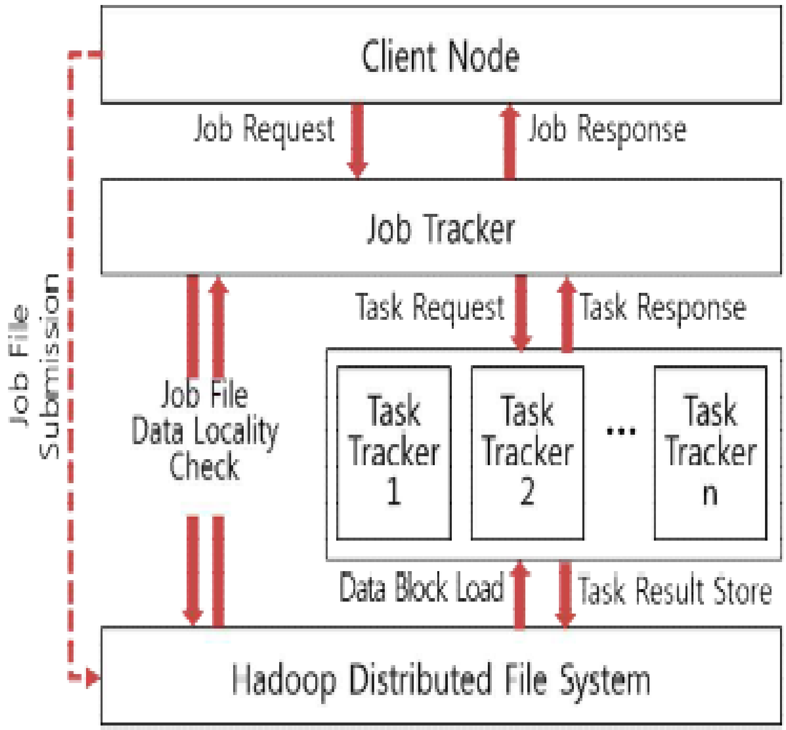

Figure.1. ***Architecture of Hadoop-based Cluster***

For the aim of observation the state of task trackers, the job tracker aggregates the heartbeat messages from the task trackers. Once storing input file into the HDFS, the information are split in fixed sized information blocks with replication (the default replication issue is 3) and the split data blocks (chunks) are keep in slave nodes. [9] The name node maintains and keeps track of information about locations related to data blocks.

A task tracker of a slave node is in charge of planning tasks within the node. A task tracker requests a task from a job tracker by sending a heartbeat message once it's an empty task slot. Whereas a task is current, it additionally sends a heartbeat message periodically and also the message includes info concerning the state of the node and therefore the status of tasks that the node executes. [10]Information node maintains data blocks keep within the local storage, and local disk data related to HDFS.

The basic flow of a Hadoop application is as follows. Once storing input data from a client, the information are divided into chunks and also the chunks are keeping to nodes. The job tracker deals with a MapReduce job request from a client. Upon reception of a job request, the job tracker divides a job into tasks, and then, the tasks are assigned to task trackers.[8] At this stage, it schedules tasks by considering information locality with information of the job. Next, every task tracker assigns a task to a node, and then, the node performs the task by loading the data block from HDFS once required. [10] A task tracker keeps track of the state of the node concerning the task. Once a task is completed, this data is sent to the job tracker and also the results of the map task are hold on at the local storage quickly. after all tasks of the job are completed, the job tracker informs the client of the completion of the job and also the client will check the results of the job, that is keep in HDFS.

3.2 ARM

ARM is that the adaptive Replication Management system. This ARM system starts by sporadically collection of signal i.e. called the heartbeat. Then, it is distributed to the heuristic detector like the training data.[4] Here this information is compared with the access patterns that are extracted from the predictor element and keep at the content. If there's a match, the access potential is then retrieved from the pattern and directly passed to the predictor element with none computation.[4] Otherwise, the training information is continuously sent. In this case, most of the computation belongs to the hyper-parameter learning and training phases of the prediction. To resolve this issue, the hyper generator is made to reduce the computational complexness of the hyper-parameter learning section. After that, the training section will begin to estimate the access potential. Finally, the access potential of the target file is passed on to the replication management element. [5]Additionally, a new pattern is additionally extracted and keeps at the knowledge base for subsequent analysis.

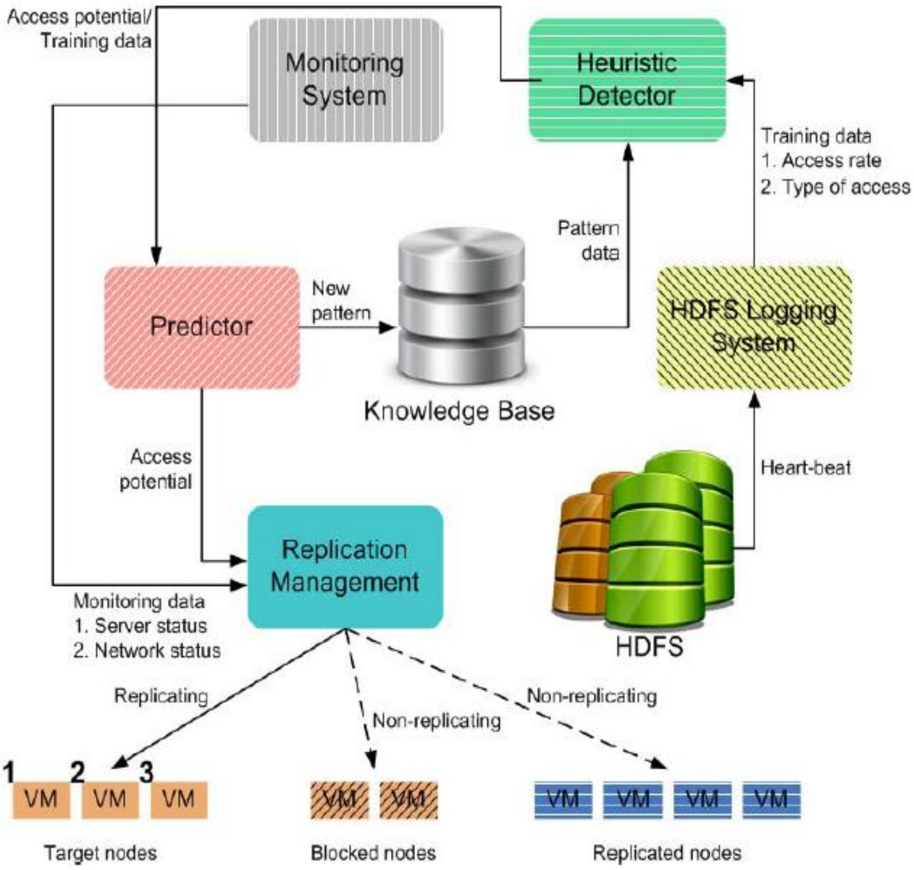

Figure.2. ***It shows an Architecture of Adaptive Replication Management system(ARM system)***

In this design, for the purpose of flexibility, elasticity, multi-tenancy and on require deployment features, virtual machinery has been chosen instead of using physical servers for node computation .[5]

4. Conclusion:

This paper study of the adaptive Replication Management in HDFS supported supervised Learning of AI attempts maybe the recent analysis work that has been done in the field. Some analysis papers were discussed, all that specialize in different aspects and technique of adaptive replication management. Though no experimental comparison was made the essence of the reviewed paper has been conferred. In this paper, these difficulty decrease methods extensively accelerate the calculation process of the access probable evaluation. In the research work we design the system that is dynamically duplicate the data files and this duplication is performed with the help of predictive analysis. By using the concepts of possibility theory, the use of each data file can be predicted to create an equivalent duplication strategy. Ultimately, the popular files can be afterward replicated according to their own access potentials and for the remaining low potential files, a removal code is applied to continue the dependability. Hence, while keeping the reliability in comparison to the default design, our advance system simultaneously improves the accessibility.

References:

1. Bui, Dinh-Mao, et al. "Adaptive Replication Management in HDFS based on Supervised Learning." IEEE Transactions on Knowledge and Data Engineering 28.6 (2016): 1369-1382.

2. You, Xindong, et al. "E^ sup 2^ ARS: An Energy-Effective Adaptive Replication Strategy in Cloud Storage System." Applied Mathematics & Information Sciences 7.6 (2013): 2409..

3. Lee, Jungha, et al. "Adaptive Data Replication Scheme Based on Access Count Prediction in Hadoop." Proceedings of the International Conference on Grid Computing and Applications (GCA). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2013).

4. Myint, Julia, and Axel Hunger. "Modeling a Load-adaptive Data Replication in Cloud Environments." CLOSER. 2013.

5. Bui, Dinh-Mao, et al. "Replication Management Framework for HDFS Based on Prediction Technique." Advanced Cloud and Big Data, 2015 Third International Conference on. IEEE, 2015.

6. Pawar Abhijeet H, 2Subhedar Dipti D., 3Phadtare Jigish K., 4Pawase Amol Ashok, 5Honrao Sayali D. “Adaptive Replication Management in HDFS based on Supervised Learning”, International Journal of Recent Engineering Research and Development (IJRERD) Volume No. 02 – Issue No. 03.

7. M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, “Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling,” in Proceedings of the 5th European conference on Computer systems. ACM, 2010, pp. 265–278.

8. K. S. Esmaili, L. Pamies-Juarez, and A. Datta, “The core storage primitive: Cross-object redundancy for efficient data repair & access in erasure coded storage,” arXiv preprint arXiv:1302.5192, 2013.

9. G. Ananthanarayanan, S. Agarwal, S. Kandula, A. Greenberg, I. Stoica, D. Harlan, and E. Harris, “Scarlett: Coping with skewed content popularity in mapreduce clusters.” in Proceedings of the Sixth Conference on Computer Systems, ser. EuroSys ’11. New York, NY, USA: ACM, 2011, pp. 287–300.

10. G. Kousiouris, G. Vafiadis, and T. Varvarigou, “Enabling proactive data management in virtualized hadoop clusters based on predicted data activity patterns.” in P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2013 Eighth International Conference on, Oct 2013, pp. 1–8.

11. Q. Wei, B. Veeravalli, B. Gong, L. Zeng, and D. Feng, “Cdrm: Acost-effective dynamic replication management scheme for cloud storage cluster.” in Cluster Computing (CLUSTER), 2010 IEEE International Conference on, Sept 2010, pp. 188–196.

12. Tudoran, Radu, et al. "Adaptive file management for scientific workflows on the azure cloud." Big Data, 2013 IEEE International Conference on. IEEE, 2013.

13. G. Ananthanarayanan, S. Agarwal, S. Kandula, A. Greenberg, I. Stoica, D. Harlan, and E. Harris, “Scarlett: Coping with skewed content popularity in mapreduce clusters.” in Proceedings of the Sixth Conference on Computer Systems, ser. EuroSys ’11. New York, NY, USA: ACM, 2011, pp. 287–300.

14. Zhu, Q. and Zhou, Y. 2005. Power-aware storage cache management. IEEE Trans. Computer, **54**, 587-602 (2005).

15. X. Yao and J. Wang. Rimac: a novel redundancy-based hierarchical cache architecture for energy efficient, high performance storage systems. In EuroSys ’06: Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems. ACM, New York, NY, USA, **2006**, 249- 262 (2006).

16. E. Pinheiro and R. Bianchini. Energy conservation techniques for disk array-based servers. In ICS ’04: Proceedings of the 18th annual international conference on Supercomputing, ACM, New York, NY, USA, 68-78 (2004).

17. Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, and J.Wilkes. Hibernator: helping disk arrays sleep through the winter. In SOSP ’05: Proceedings of the twentieth ACM symposium on Operating systems principles, ACM, New York, NY, USA, 177-190 (2005).

18. C.L. Abad, Yi Lu, R.H. Campbell, “DARE: Adaptive Data Replication for Efficient Cluster Scheduling”, *IEEE International Conference on Cluster Computing* (CLUSTER 2011), pp.159-168, 2011.

19. Q. Wei, B. Veeravalli, B. Gong, L. Zeng, and D. Feng, “Cdrm: A costeffective mdynamic replication management scheme for cloud storage cluster.” in *Cluster Computing (CLUSTER), 2010 IEEE International Conference on*, Sept 2010, pp. 188–196.

20. H. WEATHERSPOON ET AL. Erasure Coding vs. Replication: A uantitative Comparison. In Proc. IPTPS (2002).

21. J. KUBIATOWICZ ET AL. OceanStore: An Architecture for Global-Scale Persistent Storage. In Proc. ASPLOS (2000).

22. R. BHAGWAN ET AL. Total Recall: System Support for Automated Availability Management. In NSDI (2004).

23. A. PATTERSON ET AL. A Case for Redundant Arrays of Inexpensive Disks (RAID). SIGMOD Records 17, 3 (1988).

24. Sriram Rao, Benjamin Reed, and Adam Silberstein, HotROD: Managing Grid Storage With On-Demand Replication, Workshop on Data Management in the Cloud (DMC'13), April 2013.

25. Zhenhua Guo, Geoffrey Fox, and Mo Zhou. 2012. Investigation of Data Locality in MapReduce. In Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012) (CCGRID '12). IEEE Computer Society, Washington, DC, USA, 419-426.

26. Spyridon V. Gogouvitis, Gregory Katsaros, Dimosthenis Kyriazis, Athanasios Voulodimos, Roman Talyansky and Theodora Varvarigou, "Retrieving, Storing, Correlating and Distributing Information for Cloud Management," 9th International Conference on Economics of Grids, Clouds, Systems, and Services, November 2012.