A Survey of Dynamic Replication Strategies based on Content Popularity

Roshni Babykutty
CE, SVPCET
RTMNU, India

M.B.Gudadhe
CE, SVPCET
RTMNU, India

ABSTRACT
Cloud computing is a large-scale parallel and distributed computing paradigm. It is a collection of virtualized and interconnected computing resources which are managed to be one unified high performance computing power. Cloud environment being highly dynamic and heterogeneous, failures on data nodes are common. Therefore, to improve the availability and reliability of the system, the data is to be replicated to numerous suitable locations. By analyzing the content popularity, the popular data is replicated and the replicas are allocated to preferable data nodes. This phenomenon increases the data availability, speeding up of data access and minimizing cloud system bandwidth consumption. In this paper, various strategies of data replication based on content popularity are studied and how the replication can be carried out effectively by considering a few parameters like number of replicas, replica placement and replica management is analyzed.

General Terms
Content Popularity, Cloud Computing, Grid Computing, Dynamic Replication

Keywords
Cloud Computing, Content Popularity, Data Availability, Replication, Replica Allocation

1. INTRODUCTION
Data is the frontier of the 21st century. The ability to manage the data paves way for a revolution in the field of engineering and technology. The rises of cloud computing and cloud data stores have been a forerunner and facilitator to the emergence of big data. Cloud computing is a large-scale parallel and distributed computing paradigm. It has brought a paradigm shift by delivering IT services over Internet. It consists of a collection of interconnected and virtualized computing resources which are managed to be one unified computing resources. They provided abstract, virtual resources, such as networks, servers, storage, applications and data, can be delivered as a service. Services are delivered on demand to the end-users over high-speed Internet as three forms of computing architecture, namely Software as a Service (SAAS), Platforms as a Service (PAAS) and Infrastructure as a service (IAAS) [1,6]. The main objective is to provide users with more flexible services in a transparent manner, cheaper, scalable, highly available and powerful computing resources. Both the hardware and the software infrastructure are provided as a service through large scale data centers [9,11].Thus, the cloud applications are accessible anytime, anywhere and the data centers providing the services are low cost, powerful and energy efficient.

In the mid 1990s, the term Grid was used to describe technologies that would allow consumers to obtain computing power on demand. A question may arise in one's mind that ' Is Cloud computing the advanced form of Grid Computing?'
No. In comparison with two decades ago, there has been massive increase in the amount of data (terabytes, petabytes, zetabytes and so on) generated and processed, which demands high performance computing. The clusters are expensive to operate and provide only low cost virtualization. Therefore, there is a drastic difference in the scale of operation, and operating at more massive scales demands fundamentally diverse approaches to tackling problems of the present.

On the other hand, the vision remains the same i.e. to reduce the cost of computing, increase reliability and flexibility by transforming computers from something that we buy and operate ourselves something that is operated by a third party. [5]

1.1 Demands for Replication
Besides Cloud environment providing high availability, fault tolerance, and efficient access, the failures are common due to the large-scale data support. Therefore, the data needs to be replicated to multiple sites to ensure the data availability despite of the failures that occurs. Data replication allows reducing user waiting time, speeding up data access and increasing data availability by providing the user with different replicas of the same service, all of them with a coherent state. Replication is a frequently used technique in the cloud, such as GFS (Google File system) [12], HDFS (Hadoop Distributed File System). [7]. Now, as the cloud data centers have expanded rapidly in both size and number, and the dynamically scalable and totally virtualized resources are provided as a service over the Internet [8]. In most of the real cloud, dynamic data replication is achieved through data resource pool. The number of data replicas is statically set based on analysing the access patterns and is usually less than 3. This strategy works well at most time, but it may fail at certain times. It is not necessary to create replica for all data files, especially for those non-popular data files. In order to meet the high availability, high fault tolerance and high efficiency requirement, it is necessary to dynamically adjust the popular data files, the number of data replicas and the sites to place the new replicas according to the current cloud environments. [2]

1.2 Content Popularity
There are four important factors to be considered to achieve dynamic data replication based on content popularity. They are Selection of Replica, Number of Replicas, Placement of Replica and Managing the Replicas. [2,4]

Selection of replica is a crucial parameter as far as the effectively replicating data is concerned. The concept of content popularity here comes into the picture while selecting the data for replication. It is not required to replicate all the data files, especially for the ones which are less accessed. Therefore, to carry out replication effectively the content popularity of the data is to be determined. The content popularity of data can be deduced from the number of times it is accessed. The data which is frequently accessed will be having higher popularity and thus, it needs to be replicated to different suitable sites to ensure availability.

2. LITERATURE SURVEY

Mohamed-K Hussein and Mohamed-H Mousa [1] proposed a Light Weight Data Replication (LWDR) for cloud data environment wherein the popular data files for replication are selected by applying a lightweight time series technique (Holt's Linear and Exponential Smoothing-HLES) which analyzes the recent patterns of data file requests and provides predictions for the future data requests. It also identifies the best replication location based on a heuristic search for the best replication factor of each file. The proposed adaptive replication strategy enhances the response time and maintains the response time at a stable level within a short period of time. The experimental results prove that this replication strategy improves the reliability of the cloud system taken into consideration.

Sun DW, Chang GR and Gao S [2] proposed a Dynamic Data Replication Strategy (D2RS) in cloud environment based on the theory of temporal locality .i.e. the popular files in the past will be accessed more than other files in the future. The popularity of the data is determined by analyzing the access frequency of the data by the users. When the content popularity passes a dynamic threshold the replication operation will be triggered and the replica is allocated to suitable sites from where the request is been arriving, thereby reducing the bandwidth consumption. A mathematical model is formulated to describe the relationship between system availability and number of replicas. The proposed strategy effectively increases the data availability and reduces the user waiting time and speeds up the data access. The number of replicas depends on the increase of file availability; the replica placement is done in a balanced way considering the access information of directly connected data centers. The experiment setup results proved that the D2RS algorithm improved the system byte effective rate, response time and provided a good convergence rate. Also, the D2RS improved the successful execution rate, maintained the successful execution rate at a high and stable level and reduced the bandwidth consumption in cloud.

Chang et al. [3] proposed a dynamic weighted data replication strategy in data grids known as Latest Access Largest Weight (LALW). In this strategy, different weight is set for each access record according to their lifetimes in the system, and the data having higher value of weight is the most recently accessed data. The popular file is determined by the concept of temporal locality. The popular file is identified by higher value of weight as it is the most frequently accessed and it is replicated to suitable sites to achieve load balancing. The simulation results are in comparison with Least Frequently Used (LFU) and No replication on the basis of Mean time job execution, Effective Network Usage (ENU) and Storage Resources Usage. It shows that the mean time job execution of LALW is similar to LFU optimizer but excels in terms of Effective Network Usage.

N.Mansouri et al. [4] proposed an enhanced version of LALW strategy working on the concept of dynamic data replication in data grids called as Enhanced Latest Access Largest Weight (ELALW). ELALW determines in which site within the region the replica has to be placed. Storage is a matter of concern as far as the grids are considered. Therefore, the replication and the replacement of replicas is carried out efficiently. The network architecture is hierarchical consisting of two levels based on centralized data replication management, same as in the case of LALW strategy. The simulation results show that the mean time job execution of ELALW is smaller in comparison with other 8 dynamic replication strategies. The ELALW is optimized to minimize the bandwidth consumption and thus decreases the network traffic. The low value obtained for Effective Network Usage (ENU) indicated that the ELALW strategy is good at allocating the replicas to the proper sites within the cluster. The storage usage is also reduced as replication is carried out at regular intervals and replicas are stored at particular sites, thus unwanted replication is also reduced.

3. ANALYSIS OF REPLICATION STRATEGIES BASED ON CONTENT POPULARITY

The Content Popularity is a dynamic concept varied by time. The events and incidents happening around us, both informed and uninformed, has a great impact on the searches done on the web. At the advance of an event or after its occurrence, the event and the elements associated with the event remain the as the popular entities on the web, social media and even in daily communications. The popularity fades off gradually with the passing of time. Due to the dynamic nature of cloud, node failure is common. Therefore, to maintain the system availability the data is to be replicated. Replication is frequently used technique in GFS (Google File System) and HDFS (Hadoop Distributed File System) [11]. The number of data replicas is statistically set based on history experience and is usually less than 3 [2]. Increasing the data availability from the perspective of clients is the main objective of data replication. If the replicas are not used properly, then replicating all the data in the system adds to the storage utilization, bandwidth consumption and load imbalance. Thus, replication must be carried out efficiently taking into consideration the systems overall performance [10]. In order to meet the high availability, high fault tolerance and high efficiency requirement, it is necessary to dynamically adjust the popular data files, the number of data replicas and the sites to place the new replicas according to the current cloud environments. [2]

As discussed above, the four important factors to be considered to achieve dynamic data replication based on content popularity. They are Selection of Replica, Number of Replicas, Placement of Replica and Managing the Replicas.

The content popularity of a data is determined by how frequently it is accessed and different weights are set for records weights are set according to their lifetimes in the system to find the recently potential popular file [3]. The LALW, D2RS, LWDR all use the concept of temporal locality to find the access frequency of the data. The fact of temporal locality is that the recently accessed files are most likely to be requested again shortly. The ELALW strategy is based on the concept of temporal locality as well as the geographical locality. The fact of geographical locality is that the files accessed recently by a client are most likely requested by adjacent clients in the system. [4]

3.1 Hierarchical Architecture

The D2RS algorithm uses the LALW strategy to calculate the popularity degree of the data. The LALW and ELALW strategies use the same hierarchical architecture to support dynamic replication mechanism. The design of the architecture is based on a centralized data replication management. There is a Replication Manager responsible for replica management. The hierarchical architecture is given below.

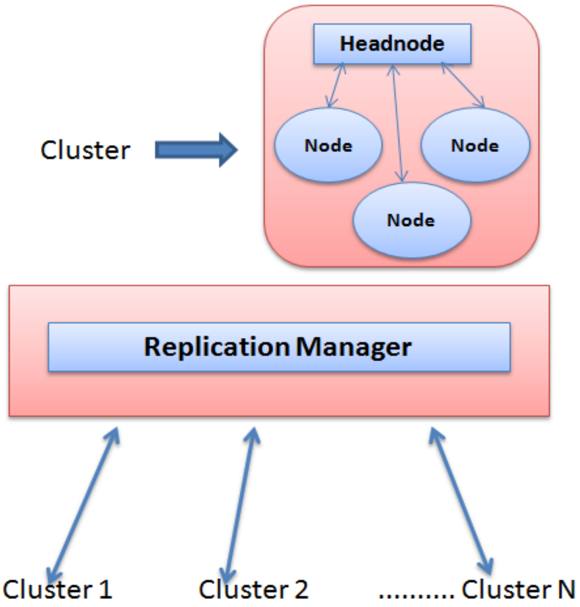

Fig 1: The Hierarchical Architecture

The grid sites are considered as a cluster. There is a cluster Headnode, which is used to manage the information about all nodes in the cluster. The Replication Manager gathers the information about accessed files from all the cluster headers. The node maintains detailed record of each file. The record in the node contains three fields which are <*timestamp*, *FileId*, *ClusterId* >. This record gives information regarding the file (*FileId)* that has been accessed by a node located in the cluster (*ClusterId)* at *timestamp*. Regularly, each site sends its records to the Headnode in the cluster. All records in the same cluster will be aggregated and summarized by the cluster Headnode. The record in cluster Headnode also contains three fields which are <*FileId*, *ClusterId*, *Number*>. This record gives information regarding the file (*FileId)* that has been accessed, how many times it was accessed (*Number)* and belonging to which cluster *(ClusterId).* After summarizing the records by a cluster Headnode, the information about file (*FileId)* and the number of times it was accessed (*Number)* is sent to the Replication Manager. The record table in the Replication Manager contains two fields- FileId and Number.

Let us take an example, consider the table entries of the Replication Manager, <A, 8> and <B, 9>. It indicates that the file A and file B have been accessed for eight times and nine times respectively. After the Replication Manager has gathered all of information in different clusters, the popular file would be selected according to the higher number of accesses for files which contributes to the weight of records. The cluster Headnode and the Replication Manager are responsible for managing the access history. The information maintained in a cluster Headnode is local and the information maintained in the Replication Manager is global.

Now, if there are two records in a cluster Header, <A, C1, 8> and <A, C2, 12>, the result after aggregating is <A, 20>. Therefore, the number of file accesses is summed up for the same FileId. The result is then sent to the Replication Manager. The Figure 2 explains the communication between the Cluster Headnode and the Replication Manager and how the records from different clusters are aggregated. [3]

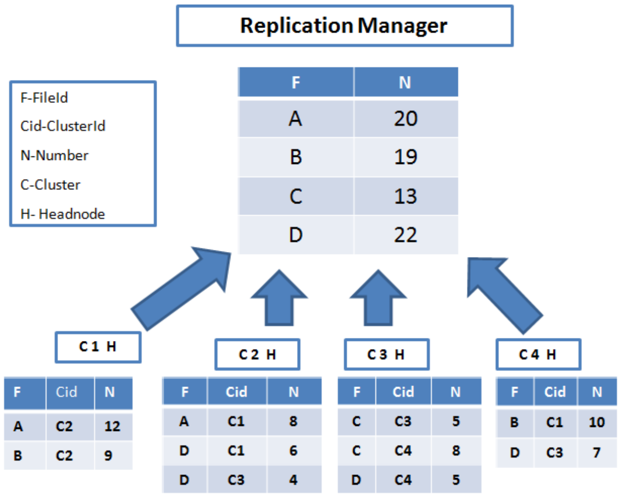

Fig: 2 An example for record aggregation

3.2 Setting Weights and Calculating Content Popularity

The Replication Manager gathers information from different clusters at different time intervals. Each file has different weights to distinguish the importance of access history. The rule of setting weight uses the concept of Half-life. Half-life indicates the time required for the quantity to decay to half of the initial value, where the quantity may be radioactive element or chemical element. In this algorithm, the weight represents the quantity, and a time interval represents the time for half-life. That is, the weight of the records in an interval decays to half of its previous weight. Setting different weight is used to evaluate the importance for history records. Older history records have smaller weights. It means that the recent history tables are preferred for referencing than previous. [3].The higher the value of the weight, more frequently it is accessed and therefore it is more popular.

The popularity degree of a block is defined as the access frequency based on time factor. It is the summation of the product of access frequency during a particular time interval and the weights associated with that block during the same time interval. [2]

The Access Frequency of each file is analysed by considering the number of times the file was accessed during that particular time interval. The file with greater value of Access Frequency is selected as the popular file. [3,4]

In replica selection step, LALW considers only the data transfer time. Considering the transfer time alone is not a sufficient parameter. The key parameter is the response time that influences the replica selection and thus the job turnaround time. ELALW strategy selects the best replica location for the users' running jobs by considering few parameters such as the storage access latency, waiting time in the storage queue and distance between nodes along with the data transfer time. Normally, in order to improve system performance the operating system dispatches the I/O requests. Scheduling can be implemented by keeping a queue of requests for the storage device. Therefore, the storage media speed and the number of requests in queue affects the average response time experienced by applications. So, the storage access latency is the delayed time for the storage media to perform the requests. This time delay depends on the file size and storage type. [4]

In LWDR the popularity degree of a block can be calculated by

using Holt's Linear and Exponential Smoothing (HLES). The

Holt’s Linear and Exponential Smoothing is a computationally cheap time series prediction technique. It is capable of smoothing and providing short-term predictions for the measured requests arrival rates and service demand rates.Hence, HLES enables the proposed framework to examine thearrival rates and service rates and to provide a short-termprediction for the future arrival rates and service rateswith low computation time. [1]

3.3 Number of Replicas

In LALW strategy, the Number of Replicas (N.R) can be calculated by comparing the average access frequency of the popular file with all other files that has been requested by the users. For example, during the first time interval if A is the popular file and N.R is four, then four replicas of file A will be created. During the next time interval if B is the popular file and N.R is three, then three replicas of file B will be created [3].

The ELALW strategy also uses the similar method to calculate the number of replicas in data grids [4].

The D2RS algorithm and LWDR calculates the Replica Factor to find the number of replicas to be placed in the cloud data centres. The Replica Factor is defined as the ratio of the popularity degree and the total number of bytes of data file requested by all tasks under given constraints. It is used to determine whether the data file should be replicated or not. Once a replication factor based on the popularity of the files is less than a specific threshold, the replication signal will be triggered. [1,2]

3.4 Replica Placement

The placement of replica to suitable data sites plays an important role in enhancing the system availability and reliability of the system. In case of node failure, the replicas are to be placed accordingly to make the data available.

The LALW determines in which region the replica has to be placed and the number of replicas has to be placed. But LALW doesn’t determine in which site within the region the file has to be placed. The ELALW overcomes this drawback of LALW, by determining the site within the region where the replica must be placed. It places replicas in two stages. In the first stage ELALW like LALW determines how many replicas have to be placed in each region. The second stage is to place the replica in the Best Storage Element within the region. To select the BSE, ELALW finds Storage Element with minimum Value-Storage Element in the region. During the calculation of minimum value Storage Element, the frequency of requests of the replica and the last time the replica was requested are taken into consideration. These parameters are essential because they give a sign of the probability of again requesting the replica. [4]

In D2RS algorithm, to achieve the system task successful execution rate and bandwidth consumption requirement, different tiers of data centers which have the selected replica data file will decide the replica placement and the placement of new replicas to be created according to the access information of directly connected data centers. The number of new replicas created at the directly connected data center is calculated based on the total number of new replicas and replica factor. [2]

3.5 Replica Management

The replicas after certain time intervals need to be managed to maintain the performance of the system. The LALW strategy uses the LFU technique which may delete some valuable files that may not be available in local region and may be required in the future. Deletions of such kind will result in a high cost of transfer. The ELALW considers three prominent parameters while considering the replacement decision: the number of requests in future based on Economic Model, the size of replica, the number of copies of the file. [4]

In the D2RS and LDWR the replicas with the smallest Replica Factor are deleted and blocks are made available for storing of new replicas. [1,2]

4. CONCLUSION

The LALW and ELALW strategy is implemented on data grids. The Data grid simulator and Optorsim were used to provide simulation environment. The performance evaluation metrics used were Mean Job Time, Effective Network Usage, and Storage usage. The results prove that ELALW achieves significant improvement of performance over LALW, as ELALW reduces unnecessary replications as it replicates at regular intervals and stores the data in best sites data is being frequently accessed. ELALW places the replica based on the concept of temporal locality and spatial locality. It selects the best replica among other replicas and improves the response time of the system. While replacing the replicas the ELALW considers the number of requests in future based on Economic Model, the size of replica and the number of copies of the file, whereas the LALW ends up deleting certain valuable files which may be requested later. This costs high cost of transfer which affects the performance of the system. The D2RS and LWDR are implemented on Cloud simulation environment using the CloudSim simulator. Both the strategies improve the system availability and reliability. They can be further improved by considering the concept of geographical locality and spatial locality. Thus, by determining the Content Popularity and carrying out the replication accordingly increases the availability of the data in the system. In the future, the existing work can be improved by enhancing the response time, speeding up of data access, data availability and maintaining the consistency of the data. Also, these strategies can be implemented and tested over real cloud environment. Caching the popular contents enables in achieving higher cache hit. Collaborative Caching and Predictive Pre-fetching can be used to achieve high data reuse rates in the network.

5. REFERENCES

[1] Mohamed-K HUSSEIN, Mohamed-H MOUSA, "A Light-weight Data Replication for Cloud Data Centers Environment", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Issue 1, January 2014

[2] Sun, D.-W., et al., "Modeling a Dynamic Data Replication Strategy to Increase System Availability in Cloud Computing Environments", Journal of Computer Science and Technology, 2012. **27**(2): p. 256-272.

[3] Chang, R.-S. and H.-P. Chang, "A dynamic data replication strategy using access-weights in data grids",J.Supercomput., 2008. **45**(3): p. 277-295.

[4] N. Mansouri and A.Asadi, "Weighted Data Replication Strategy for Data Grid Considering Economic Approach", International Journal of Computer, Electrical, Automation, Control and Information Engineering, 2014

[5] Foster I, Zhao Y, Raicu I, Lu S Y, " Cloud computing and grid computing 360-degree compared" In Proc. Grid

Computing Environments Workshop, Austin, TX, USA, Nov. 12-16, 2008,pp.1-10.

[6] Buyya R, Yeo C S, Venugopal S, Broberg J, Brandic I. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, 2009, 25(6): 599-616.

[7] Shvachko K, Hairong K, Radia S, Chansler R, "The Hadoop distributed File system", In Proc. the 26th Symposium on Mass Storage Systems and Technologies, Incline Village, NV, USA, May 3-7, 2010, pp.1-10.

[8] Wang S S, Yan K Q, Wang S C. "Achieving efficient agreement within a dual-failure cloud-computing environment", Expert System with Applications, 2010, 38(1): 906-915.

[9] Armbrust M, Fox A, Grffith R, Joseph A D, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M, "A view of cloud computing", Communications of the ACM, 2010, 53(4): 50-58.

[10] Wei, Q., et al., "CDRM: A cost-effective dynamic replication management scheme for cloud storage cluster", IEEE International on Cluster Computing. 2010. p. 188 - 196

[11] Mell P, Grance T, "The NIST definition of cloud computing", Communications of the ACM, 2010, 53(6): 50.

[12] Ghemawat S, Gobioff H, Leung S T, "The Google File system", ACM SIGOPS Operating Systems Review, 2003, 37(5): 29-43.