International Association of Scientific Innovation and Research (IASIR)
(An Association Unifying the Sciences, Engineering, and Applied Research)

ISSN (Print): 2279-0063
ISSN (Online): 2279-0071

International Journal of Software and Web Sciences (IJSWS)
www.iasir.net

Dynamic Data Storage and Replication Based on the Category and Data Access Patterns

Priya Deshpande[1], Radhika Jaju[2]
[1]Assistant Professor, [2]Student M.E. (I.T)
[1,2]Department of Information Technology Engineering,
MIT College of Engineering, Kothrud, Pune 411038, Maharashtra, INDIA

Abstract: *Now days, Big data storage is becoming tricky issue. And it's becoming current popular research topic. Dealing with the large scale massive data needed to be stored efficiently and accessed easily which is a crucial problem. This paper gives the solution to this problem. Category wise Data distribution will help to improve the data access time. And ultimately it will reduce the job execution time which will improve in the performance of data grid. Here we are using one algorithm K-Means to divide the data category wise. Then we are working on the limited storage capacity of the node. Due to limited capacity of the node we need to replace the data and to do this we need some replication strategy. So here we are using different Strategy depending on the user's data access pattern. This paper will focus on the improvement of the performance, reduction in bandwidth consumption, Efficient and easy data access.*

Keywords: *Category, Popularity, Data Access Pattern*

I. Introduction

Data grid technology has main feature, storage capability which is becoming a popular research topic now a days. Dynamic massive large size data is stored through data grid technology with the help of the internet. The data size of such a massive data is growing day by day and so there is necessity of efficient handling of such data. So for the efficient handling, data placement, replacement and the data access should be effective.

Large number of data is generating regularly, and this data is needed to be stored properly for efficient data access and also to prevent the data loss, memory loss, data redundancy, data duplications. So

- How to place the data?
- Where to place the data?
- Why to place the data? [3]

Hadoop is open source framework, works on the storage issue. Hadoop plays important role in distributed computing system. Here following diagram will show the process of storing data on hadoop.

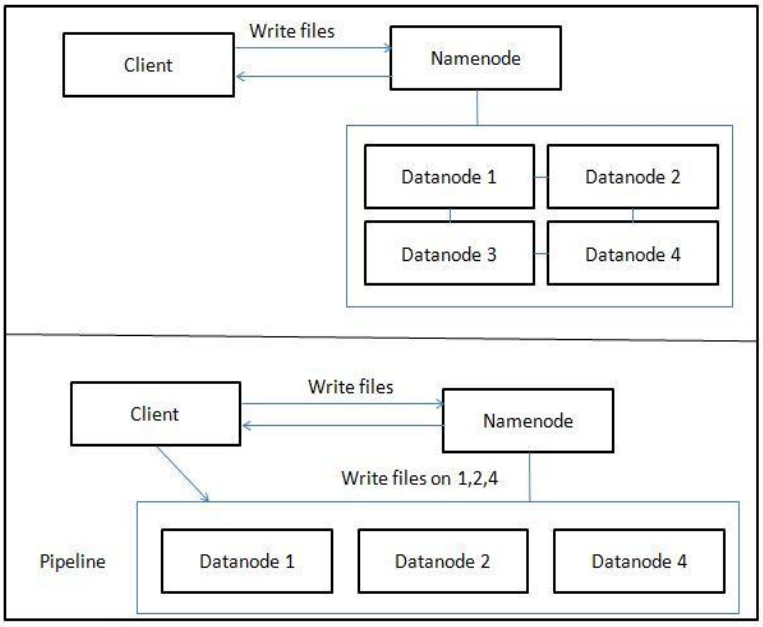

Figure 1: Hadoop Data Storage process. [7]

Figure shows the basic architecture of Hadoop. Namenode connected to number of data nodes. Client asks to namenode to store the data. Namenode checks the availability of data nodes and send the acknowledgement to the client accordingly.
Selected datanodes are pipelined together to store the data. Then client directly send data to the datanodes and stores data on those datanodes [7]. With these good functionalities hadoop have some disadvantages also like Redundancy, overwriting, overheads, data loading, data retrieval etc. and ultimately this affects on the overall performance. So we need to find out solutions on these issues. Here numbers of strategies are defined and executed to overcome the issues. Some of them have shown effective results also.
Our data placement strategy will give the answers to these questions. Number of strategies are introduced and implemented for the category wise placement of the data. Some strategies have shown effective results also. Depends on the data, jobs will be assigned to the sites accordingly.
Replacement strategy are used for the better performance in the data intensive applications There are many replacement strategies are proposed for the better performance and some of them are showing better results. Here in this paper we are trying to apply different strategies for the different type of data for the better performance and to avoid the overheads.
Replacement of the file will be depends on the access frequency of that file and also on the requirement of that file to that particular node. Jobs will be executed on the sites where the job related data is more. So the data transfer time will be reduced and bandwidth consumption will also reduce.

II. Related Work

Many replication strategies are implemented for the replication purpose. So here some strategies which we have studied for the reference basic and the old strategies which shows good effect in replication are as follows
LRU (Least Recently used): In this strategy the replica which is not in use from long time will get deleted for the replication i.e. file which is not used recently will get deleted. And if the size of deleting file is less than the new replica then the next least recent file will get deleted for the new replica placement. And the process goes on [1].
LFU (Least Frequently Used): This strategy is based on the replication of the file. In this strategy less popular file will get replaced by newer one. Here the files on which jobs will perform will be stored in local storage. And the less accessed file will be removed from the storage for the new replica even if the file to be deleted is newly replicated [2].
DORS: in this paper different replication strategies are used for different access patterns. The replication is based on the replica's value. Replica value is calculated and accordingly. This strategy considers the parameter like file size, network status and file access history [3].
Chang has proposed LALW (Latest Access Largest weight), here largest weight will be applied to the file which is accessed most recently. Similarly SATO et. al presented small modification to the simple replication algorithms on the basis of file access pattern and network capacity [4].
DRCP is Dynamic Replica creation and placement proposes the placement policy and the replica selection to reduce the execution time and bandwidth consumption. Their replication is based on the popularity of the file and this strategy is implemented using data grid simulator, OptorSim [5, 6].
In this paper we are proposed strategy for the data placement where we are storing the data category wise and accordingly we are applying different replication policies based on the data access patterns. Rest of the paper is divided as section 3 will describe the system architecture and the storage of the data and defines the different replication strategies according to the data access pattern. Section 4 gives the conclusion and future work and Section 5 suggests the references.

III. System Architecture

Figure gives the overall idea of proposed strategy. Different jobs from different clients are requested to the job broker. Here job broker works as a namenode of hadoop. Then job broker will run the K-means algorithm to find out the category of the data to be submitted. Then according to category data will be stored on the particular categorized node.
Once the data is stored replication point comes into the picture. What if particular datanode fails? How to recover the data lost in failure. The Solution is Replication. Creating number of replicas of files and storing them on different data nodes so that we can retrieve the data from other nodes if particular data node fails. So many replication strategies are developed as a solution. Then again here we are Applying different replication strategies according to the access pattern. We will keep the pattern recognizer and log files in the replica manager to find out the access pattern of data and Applying the replication strategy accordingly. Replica log file maintains the replica records that are replica number, path etc. Here each file has a unique record number to avoid the redundancy.

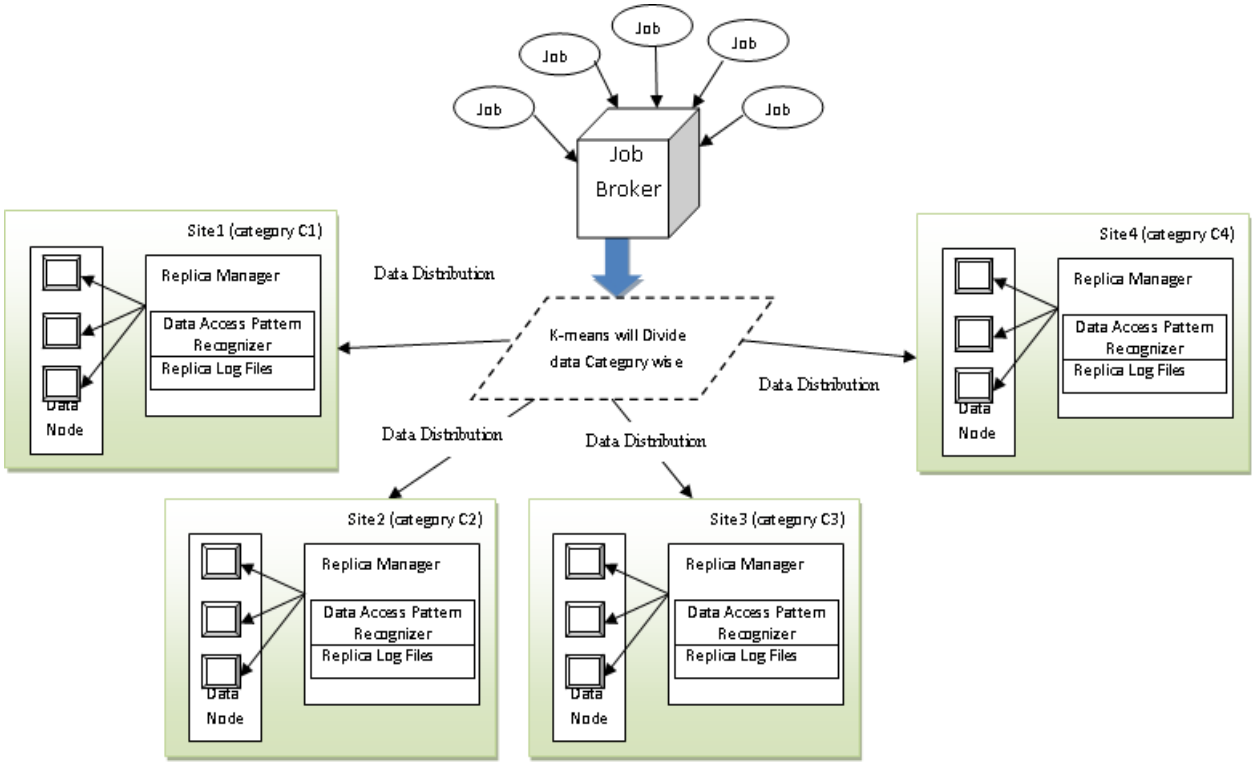

Figure 2: Data Storage and Replacement System Architecture [8].

A. Data Distribution

Data storage is the important issue as the data size is increasing rapidly day by day. After the storage of the data, information retrieval is another big issue. The data stored is of different types, so retrieving required data from the collection of the different data is quite difficult. So to retrieve required data easily, some operations on the data are needed to be performed.

Operations like dividing data into different category. And storing this category wise data on different nodes where the request ratio to the particular data is high. Data will be retrieved easily and the file transfer traffic ratio will be low ultimately it will effect on the performance and cost of the file transfer. Data will be divided category wise. For example, Plastic factory data will have different category like vendors data, supplier's data,

There are different strategies are studied to divide such a data category wise. K-means is one of the algorithms used for the data categorization purpose. It is a well known partitioning algorithm where the objects are categorized as they belong to the one of K-groups, here K is priori. Depending on the mean multidimensional version i.e. centroid of the cluster, the belonging of that object to the particular cluster is finalized. It means object is assigned to the group having closest centroid [10,11].

K-means works particularly by calculating centroid of each cluster. And it is cost effective. Basic k-means algorithm is

```
{
Select K ;
// where K = initial centroids of K points;
DO
{
Create K clusters;
// i.e. assign each point to its closest centroid
Recalculate the centroid of each cluster;
}
WHILE centroids of the cluster do not change;
}
```

Whenever data will come to the job tracker, job tracker will invoke the K-means algorithm. K-means will divide that data category wise and that categorized data will be stored on to the node assigned for that category. In case of storage full, the new arriving data will automatically propagate to the nearer node.

B Replication

When we are requesting certain files say [f1, f2, f3…fn] for executing particular job, some of the files will be available on the local storage and these file will execute directly. But the files which are not on the local storage have to be fetched from other nodes and have to store on local node, then execution will be carried out. And as

we know the limited storage capacity of the node, what if the storage is full? Where to store these files? Answer is deleting some files of local storage and store the new one. Again the question arises that which file should be deleted? How many files will be deleted for the storage of new files? These are the some key points we are considering while applying our strategy.

Here we will achieve our strategy results in two steps first is to decide whether the file from other node should be stored on local device or not. And second is to apply the different replacement strategy depending on the data access pattern.

1: The storage of the file will be depends on the replication factor. If the file having copies less than the replication factor then they are copied and if they are having number more than it they will not get copied. The replication factor will be decided on the basis of the capacity of all the nodes to the total size of all the files.

$$R = C/W \quad \text{-------[9]}$$

R= Replication factor, C= capacity of all the nodes, W= Total size of all files in data grid.

Here R will decide whether to replicate the file or not. If the copies of the files are less than R then files will be replaced otherwise not.

2: The different replacement strategy will be used for the different data access patterns. We will use the strategies which are showing better results for the particular data access pattern [8]. For example for random data access pattern LRU shows the better performance. And the replication will be depends on the request rate of user. That is the file which is not in a use for long time on that particular node will get replaced by new one. And to decide how many replicas and when to replicate the file is depends on the replication factor.

Here the category is also important for the replication. Depending on the category of the data, it will goes to the particular node assigned for similar type of data. Here we are assuming that we are assigning a particular node for particular subject information. For example if particular file of data belongs to chemistry domain then it will go to the node assigned for chemistry domain.

IV. Conclusion and Future Work

In This paper, our data distribution strategy will help to improve the data access time. K-means algorithm will divide the data category wise and then it will send to respective node assigned for the particular category. So when the user request for the file or stores the file, K-Means will run and will go to particular category node and will perform on it.

As our Replication strategy is based on the user's data access pattern, replica strategy will depend on it. This will shows the better results than applying same strategy for all type of access pattern.

In future we will also consider the scheduling criteria, load balancing, recovering so that we can perform on whole system and will give better results.

References

[1] W.H. Bell, D.G. Cameron, R. Carvajal Schiaffino, AP. Millar, K.Stockinger, F. Zini, Evaluation of an economy- based file replication strategy for a data grid, in: Proc. of 3rd IEEE In!. Symposium on Cluster Computing and the Grid, CCGrid'2003, IEEE CS-Press, Japan, (2003).

[2] W.H. Bell, D.G. Cameron, R. Carvajal-Schiaffino, AP. Millar, K.Stockinger, F.Zini, Evaluating Scheduling and Replica Optimization Strategies in Data Grid, IEEE (2003).

[3] W. Zhao, et al., "A Dynamic Optimal Replication Strategy in Data Grid Environment", International Conference on Internet Technology and Applications, pp. 1-4, 2010.

[4] R.S. Chang, H.P. Chang, "A Dynamic Data Replication Strategy Using Access-Weights in Data Grids," Supercomputing, Vol. 45, No. 3, pp. 277-295, 2008.

[5] K. Sashi, A. Selvadoss Thanamani, A New Replica Creation and Placement Algorithm for Data Grid Environment, IEEE – International Conference on Data Storage and Data Engineering (2010).

[6] K. Sashi, A. Selvadoss Thanamani, Dynamic Replication in a Data Grid using a Modified BHR Region Based Algorithm, Elsevier – Future Generation Computer Systems (2011).

[7] White, Tom. *Hadoop The Definitive Guide.* Sebastopol : O'Reilly, 2010.

[8] Myunghoon Jeon, Kwang-Ho Lim, Hyun Ahn, Byoung-Dai Lee, Dynamic data replication scheme in cloud computing environment, 2012 IEEE Second Symposium on Network Cloud Computing and Applications

[9] Wolfgang Hoschek, Francisco Javier Jaén-Martínez, Asad Samar, Heinz Stockinger, and Kurt Stockinger, "Data Management in an International Data Grid Project", Proceedings of the First IEEE/ACM International Workshop on Grid Computing, Springer-Verlag, 2000, pp. 77-90

[10] Chen G., Jaradat S., Banerjee N., Tanaka T., Ko M., and Zhang M., "Evaluation and Comparison of clustering algorithms in analyzing ES cell Gene Expression Data", Statistica Sinica, vol. 12, pp. 241-262, 2002.

[11] Osama Abu Abbas, 'Comparisons between data clustering Algorithms', The international Arab journal of information technology, Vol. 5, No. 3, July 2008.