ISSN:0975-9646

Vijaya -Kumar-C et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 3034-3038

Optimization of Large Data in Cloud computing using Replication Methods

Vijaya -Kumar-C , Dr. G.A. Ramachandhra

Computer Science and Technology, Sri Krishnadevaraya University
Anantapuramu, AndhraPradesh, India

Abstract-Cloud computing provides dynamically adopting and frequently used the virtual resources from the Web. We can developed different data-intensive applications have High quality of service requirements[1] - [3]. To reduce the data corruption and improve the performance of the cloud computing we proposed the Qos-aware data replication algorithms[4]. We can achieve in two ways, the first algorithm the intuitive idea of high-QoS first-replication to perform data replication using the partitions. To achieve these two minimum objectives, the first algorithm transforms the QADR problem into the well-known minimum-cost maximum-flow problem. In cloud computing system have large number of nodes to increase the performance to compress the data using Data block storage. We also propose node combination techniques to reduce the possibly large data replication time. Using the Finally, simulation experiments are performed to demonstrate the effectiveness of the proposed algorithms in the data replication and recovery

Keywords: *Cloud computing, data-intensive application, Partitions, quality of service, data replication, Node combination.*

1. INTRODUCTION

A new generation of data-intensive applications that require high speed and ultra-low latency Internet infrastructure to ensure massive amounts of uninterrupted data ingestion, real-time analysis and cost efficiency at scale are surpassing the capabilities of traditional virtualized public clouds[2]. Virtualization – along with the processing overhead of its hypervisor layer and resource constraints resulting from shared hardware. The existing Internet provides to us content in the forms of videos, emails and information served up in web pages. With Cloud Computing, the next generation of Internet will allow us to "buy" IT services from a web portal[3]. New generation of high-performance computing center does not only provide traditional high-performance computing, nor it is only a high-performance equipment solution. The management of resources, users and virtualization, the dynamic resource generation and recycling should also be taken into account. Apache Hadoop is an emerging cloud computing platform dedicated for data-intensive applications[6]. For large volume of data application like Google,Amazon,Mircosoft ,e.t.c may consist of hundreds or thousands of server machines, each storing part of the file system's data. The fact that there are a huge number of components and that each component has a non-trivial probability of failure.

To overcome the hardware failures, adopting the extensions of partitions techniques and Compression techniques. Whenever the hardware failures occurs, the Qos requirement of the application cannot be supported continuously. With a large number of nodes in the cloud computing system, it is difficult to ask all nodes with the same performance and capacity in their CPUs, memory, and disks. For example , Google is realistic heterogeneous cloud computing , which provides different infrastructure along with resource types to satisfied the users requirements[6]. Whenever using the node heterogeneity the data of the high-Qos application may be replicated in a low performance node. So that the data replication is running from low performance node has slow communication.

Architecture of HDFS

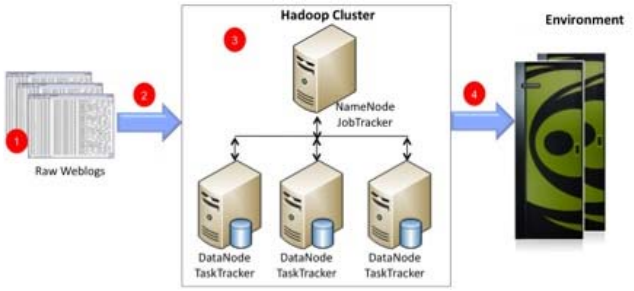

Fig 1

To solve the problem, we have proposed greedy algorithm, in this algorithm, if the application 'n' has QOS requirements it will consider next 'n-1' applications to perform replications. Using these algorithm we can't achieve the minimum cost and maximum results . To get the optimal solutions DSQS we are using the stagecoach dynamic programming. So we can solve more complex problems and easily to find out the optimal solutions in more efficiency. If we compare the HQADR and DRQS, DRQS has reduce the size of the disk and also reduce the retrieval of the time . The complexities of solving problem is dependent on the number of nodes are existing the server. We can control the web traffic by given the permissions to the users for accessing the particular disks with that access user cannot access all the servers. So we can increase the scalability and capabilities of our servers. To working on Big-data in cloud computing system, some of issues like increase data after replications are creation and updating the servers.

II PRELIMINARIES

2. 1 A. System Model

We refer to the architecture of the Hadoop distributed file system (HDFS), is designed for high fault tolerant conditions with low cost of hardware. Using HDFS architecture we developed the data replication algorithm and refer the Google file system as having the same properties[6]. In HDFS, a single Name node and number of data sets, these Data nodes and Name nodes are migrate and form a rack. The Name node refers to mapping of the Data blocks to Data nodes. Data nodes consists of one or more data blocks, and Data nodes maybe desirable to move from one Name node to another Name node. When the application are executed in data nodes, it should be process from the data blocks and these data blocks acts as client and send the request to the Data nodes. These data nodes are send the request to the Name nodes. We used the network topology and all switches for communication protocol : Spanning tree protocol.

2. 2 Related work

To reduce failures in cloud computing we are introducing the save status technique. Using these technique we can tolerate the server failures. However, Hadoop file systems, saves the data blocks in Name nodes or we can save the file systems as per the user needs. Whenever the failures occurs the file system and data blocks and other directories are restored using the save status technique.

Replication methods are protect and stored the data blocks against the disk failures[7]. In HDFS the data is stored data blocks, and the these data blocks are created as two replicas and stored in different Data Node. Data blocks are referred as a Data node i, original data of data block. Using the Back Method we can get the data replica updates and data access in process, which can improve the update data to the users and reduce the response time. However three replica, replication strategy is used to reduce the cost effective data and it apply the reliability management mechanism. Based on this mechanism to active the replica verify the minimum number of replications are stored so that we reduce the storage space consumptions. It dependent on the storage constraint, we used to distributed paging and it works on the dynamic allocation of the copies of data blocks in the distributed network. we have to minimize the total communication cost to set the number of sequence of write and read access to send the requests from the rack. In this work, how to define the position of the replicas of each data object replications and users can access the data objects from the server. DPQS problem is proved on the NP-Complete , have used add, modify the objects without increase the size of the data objects and improve the QOS[8]. We have proposed the algorithm takes long process to execution. To reduce the execution time we introduces cover set. The cover set of a server 'S' is set of servers , it has send requests from the 'r' , within S(r), when S(r) is requested from the QoS

Using node combination, dynamic network and find the shortest path of nodes Nij(t) changes of the functions of time. For dynamic replication D=(d0,dr1,dr2) with discrete time costs of set of nodes N,(|N| =n) node set N={1,2,3,4....n} and set of name nodes Sn (|Sn| =m) .We proposed an integrated solutions for dynamic data replication that as requests number of location of the replicas to be inserted / deleted aiming at workload balancing For creating replicas we need to Storage and network resources are consumed and handling and the node workload increases or decreases is depending upon the creation of the replicas and sizes of the replicas. To maintain the creation of replicas for efficiently in a data-oriented manner, considers the parameters to identified problems of the total disk space and space that are used for the creation of new replicas[9]. When a node receives a read request from the client, it determine the specified level and number of replicas that have stored in save check point[11]. When each replica responds with the requested data that should be compared with save check point and most recent version data.

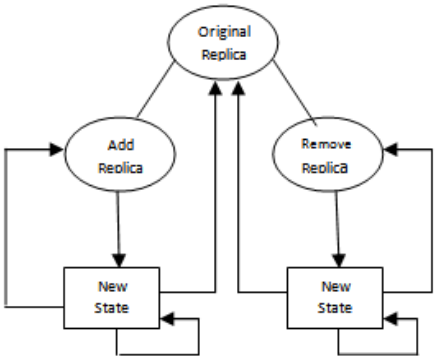

Figure 1

3. DEFINED THE "FREQUENTLY " USED DATA REPLICAS

Many clients are mainly focus on the QoS requirement for frequently access the data sets and much importance to the frequently used data sets. Importance of data set needs more replicas to be created, so clients are sending the number of retrieval requests in fixed intervals for that we get the number of the importance data sets of the system is defined Ds(i), i =1.....n where n is number of data sets in system. We get total requests for data retrievals in fixed intervals

$$Do = \sum Ds(i)$$

Do(i) is similar to the Daccess are defined the sum of the network communication latency and disk access latency for retrieving data block replica from the node ni to node rj, includes the time to transmit a data replica from rj to node ni . Time to write the data block replica into the disk ni for limited amount disk space the requested nodes are placed their data block replicas at the same qualified nodes.

Time Complexity : We analyze the time complexity of the DQOS algorithm , the DQOS algorithm can be divided into two parts Importance replicas request and frequently replicas request. In the Qos requirement of the request node is high in the case of "high" , the data replication request is frequently replica request used and also importance replica request in the data block.

In cases of creating replicas, the save point is again used to determine the amount of nodes required to respond with server response before the duplicate replication is added. If the node is required for saving point user can able to create dynamic replication then the replica node come into online to easy access the multiple users at any time in the server response.

d[v] = Length of current shortest path from the server to v. $\lambda(s,v)$ length of the shortest path., $\pi(v)$ = predecessor of V in the shortest path from s to v.

Relax(u,v,w)
if d[v] > d[u] + w[u,v]
d[v] = d[u]+ w[u,v]
π[v] = u.

3.1 Data Replication creation approach:

Before creating the data replica we need to store the data sets in space, for that we have allocated based on the frequently used replicas and also important used replicas[11]. We approach to create and stored the data replicas in data sets and it finding how many replicas should be created for each data set.

Algorithm :
1. Sort the data files in descending order of size.
2. Define data set as i
3. For every data set i, to calculate the number of replications are created using formula n(i) = [s[i]/size(i)].
4. Using the frequently data to arranged a used data and unused data.
5. Indentified the used data as stored in spaceA.
6. Space A(i) = n(i) - n(j). Size(j)
7. Calculate the total Used space , Space A = $\sum_{i}^{n} i = 0$
8. Select the first data set in the list of descending order of size(i).
9.Calculate the unused space using spaceA^1 = space A-size (i).
10. Go to step 5 until, data set is completed.

Lemma : The relaxation operation maintains the inverts that d[u] $\geq \lambda(s,v)$ for all v€V.

Table 1- Summary of notations

Set	Description
S	A Set of all the storage nodes
Sn	Total number of nodes
Sd	A Set of storage nodes that contain the data replica
Sq	A Set of storage nodes that are qualified
Snu	A Set of nodes that are not qualified.
Str	Total number of data replications
S sm	Total number of active nodes
Scn	A node that are nearest to the data replica

Input: A set of requested nodes Sr.
Output: Optimal Placement for the QoS and get the data replicas
1: Sd ← Θ
2: for each requested node ni in Scn do
3. Sd ← for qualified nodes ni and find nearest node.
4. Sd ← Sd U Sq ri
5. End For
6. Sd and Scn are the network flow graph
7. Use the data values on the edges of the network flow graph.
8. Apply the data values and use the optimal placement for QoS data replicas and D_MCMF
9. Str ← Θ and Sqr ← Θ
10. for each requested replicas in Sd do
11. Ssm ← the amount of data nodes are in active
12. if Ssm< Sr then
13.Str ← Str U Sd
14.Sn ← S-- Suq (For unqualified nodes)
15. Snr ← Snr U Suq
16. End if
17. End for
18. Use the data values on the edges of the network flow graph
19. Apply the optimal placement for QoS data replicas.

Optimal Replica Placement algorithm.

3.2 Deactivated replication method.

Data replication is very consuming the data stored in the disk space , whenever the data replication is not used , that have dropped into the unused disk space and it can be deleted, to free the storage space. We proposed algorithm to number of replications are unused to be deleted for reduce the data sets and increase the disk space. we defined the replications s(d) is activated and 1-s(d) defined deactivated[13]. For using 'n' number of storage data replications we can define

$$\sum_{i=1}^{n} \{1 - s(d)\} = n - \sum_{i}^{n} s(d) = n - 1$$

we can get

$S^1(d) =$ 1-s(d) / n-1

Algorithm.
1. Sort the data files in list of descending order of size.
2. Define the data sets as i and data sets size = i_{λ} for λ = 1,2,...n-1.
3. λ =1
4. if λ= n then the algorithm is exit.
5. calculate the number of deactivated replicas by using $p(i\lambda) = \min[s(i\lambda)]/size(i\lambda)], q(i\lambda)-1)$.
6. Quality Q= $s(i\lambda)$- $p(i\lambda)$ * $size(i\lambda)$, data sets are storage and that have to shared in to data blocks.
7. λ = λ +1
8. go to step 4

Defined the data replications are deactivated : Let S be the storage nodes that represents the data set |S| = i, j number of data replications that are deactivated $P^{s}_{(i-j)}$, (i-j) indicates combinations of S and R data sets that are data request to server[15]. However the data replication that are stored nodes of the data set S- $P^{s}_{(i-j)}$ can be deleted from the data blocks and it will update automatically, finally it increase the disk space and improve the performance of the data blocks.

4. IMPLEMENTED SIMULATION

In simulation implementation, we have used a tree structure to the network topology. Using STP protocols can form a network topology to provide the inter-rack node and intra rack node for communication. In the HDFS, we implemented the rack formation and also node distributions, we consider 100 racks and each rack is contains the one switch, that are distributed in a unit square plane for using that we can implemented the simulations. To organize the 100 racks , we can make them one rack is mentioned as the central rack, 4000 nodes are randomly deployed and specified nodes are in active mode and remaining nodes are in deactivate mode. For the deactivate mode we are consider the unqualified nodes for the network topology for the "frequently" used data . Whenever the nodes are in the same rack along with active nodes and deactivate nodes we are occupied the locations only for the active nodes.

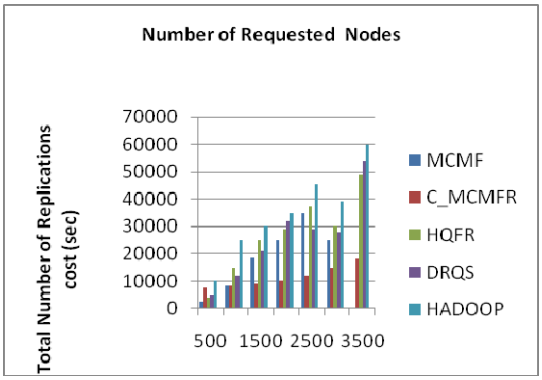

Figure 2

We have used parameter setting to generated network topology to get the simulation experiment results. In this network topology we are stored the maximum number of data block replicas and each node is available in replication space. For experiment, selecting the random number of data blocks intervals of [0,50] [51,100], and also we used same data blocks intervals for QoS requirements of an application in the node Sd . To analysis the data replicas we used lower bound to access the time for less local disk and upper bound to large access time for local disk for specified QoS intervals.

In Simulation results, we randomly used 2000 nodes to run the big data applications instead of 4000 nodes. However we used the frequently used data replications as activate nodes and remaining nodes we make them as deactivated. For this experiment we consider requested replicas as 500, 1000,1500,2000 nodes. We consider in disk access latency and applied to the workloads in sort order, whenever the rack space get request from the one or more nodes are distributed with their respectively to data replications[14] . For using the write and read operations, we can read the data block i node from the particular disk and transmits data blocks from the node j and it received according to the disk queue. We used lower bound and upper bound parameters to minimize the requested data replications from the rack space. In simulation experiments

we used[0 50] [51 100] simulations run and get the results mean values of 50.

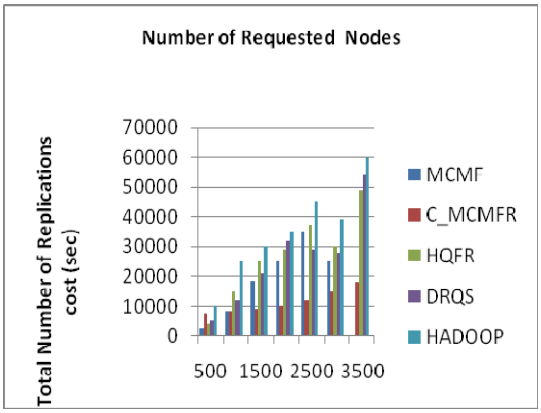

Figure 3

In simulation , the total replication cost for different number of requested nodes from 500 to 2000 , it configuration to the rack space and used different types of disk access time. Fig --- the total number of requested nodes are minimize and also minimize the total replication cost. We consider the HQRS, Hadoop replication algorithm to replace the data blocks when the rack space is failure scenario. We have implemented DRQS replication algorithm has consider failure scenario and also improve the performance of the rack space . We proposed the algorithm reduce the total replication cost and reduce the total time.

CONCLUSIONS AND FUTURE WORK

We have investigated the QoS data replication for Big data applications in cloud computing .

To find the sloution for HQRS problem and we have implemented the DRQS algorithm to improve the performance of the QoS data replication in cloud computing.

In future we have to implement the DRQS algorithm in a real cloud computing platform and also to decrease the storage in data centers and energy consumption to save the power and importance for the green cloud computing.

ACKNOWLEDGMENT

This research was supported by the Sri Krishnadevaraya University , Anantapuramu,Andhra Pradesh.

REFERENCES

[1].http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf.

[2]. M. Creeger, “Cloud Computing: An Overview,” Queue, vol. 7, no. 5, pp. 2:3–2:4.

[3] S. Venugopal, J. Broberg, and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," Future Gener. Comput. System.
[4] B. Kemme, R. Jim´enez-Peris, and M. Pati˜no-Mart´ınez, Database Replication, Synthesis Lectures on Data Management. Morgan & Claypool Publishers.
[5] S. Savinov and K. Daudjee, "Dynamic database replica provisioning through virtualization," in CloudDB.
[6] Apache Hadoop Project. Available from internet. http://hadoop.apache.org.
[7] E. Pinheiro, W.-D. Weber, and L. A. Barroso, "Failure Trends in a Large Disk Drive Population," .
[8] A. Gao and L. Diao, "Lazy Update Propagation for Data Replication in Cloud Computing,"
[9] Amazon EC2 Available from internet http://aws.amazon.com/ec2/
[10] X. Tang and J. Xu, "QoS-Aware Replica Placement for Content Distribution,".
[11] R.-S. Chang, P.-H. Chen, Complete and fragmented replica selection and retrieval in data Grids, Future Generation Computer Systems.
[12] K. Ranganathan, I. Foster, Identifying dynamic replication strategies for a high performance data Grid and Grid Computing.
[13] M. Karlsson, C. Karamanolis, M. Mahalingam, A framework for evaluating replica placement algorithms, Technical Report, Hewlett Packard Labs.
[14] Sodan, A. (2009). Adaptive scheduling for qos virtual machines under different resource availability first experiences. 14th Workshop on Job Scheduling Strategies for Parallel Processing, Vol. LNCS 5798, Rome, Italy, 259–279.
[15] Synodinos, D. G. . LHC Grid: Data storage and analysis for the largest scientific instrument on the planet. http://www.infoq.com/articles/lhc-grid.

C. Vijaya Kumar is currently a Ph.D. student in the Department of Computer science and Technology from Sri Krishnadevaraya University. He received the B.SC degree in computer science from Sri Krishnadevaraya University and the M.C.A degree in computer science and information engineering from .the Sri Venkateswara University, Trirupathi, in 2008, . His research interests include cloud computing,
and Hadoop.

Dr . G.A Ramachandra is an associate professor at Sri Krishnadevaraya University. His research interests are Data Mining , Computer networks.