ISSN:0975-9646

Review On Data Replication with QoS and Energy Consumption for Data Intensive Applications in Cloud Computing

Ms. More Reena S[1], Prof.Nilesh V. Alone[2]

Department of Computer Engg, University of Pune
GES's R. H. Sapat College of Engg, Nasik, India

***Abstract*— In today's IT field cloud computing provides scalability on storage resources.Data-intensive applications are also developed in this technology. Occurrence of data Corruption on these data intensive applications does not meet the QoS Requirements. Different applications have different quality of service (QoS).So in order to satisfy the requirements; the two algorithms are compared from existing paper. One is HQFR which uses the greedy algorithms. The Cost of data replication and the count of QoS violated data replicas cannot minimize by the HQFR algorithm. MCMF algorithm achieves these two objectives of QADR problem because it provides a polynomial time optimal solution. As we have to consider more number of nodes as it is in cloud environment due to which the computation time is high while compared to HQFR algorithm. Combination of nodes–Technique has been introduced in MCMF to find a solution for this time complexity. Further this implementation has been extended to concern energy consumption in cloud environment.**

KEY WORDS: QOS, HQFR, MCMF.

I. INTRODUCTION

In Cloud computing, a large pool of systems is connected in private or public networks which provide dynamically scalable infrastructure for applications, data and file storage. Cloud Computing delivers essential information for Enterprises from various storage resources without knowing their origin. Some issues are playing a major role in large scale organizations while working under cloud are areas of cloud applications management ,cloud backup and data recovery, cloud interoperability & data analytics, liability issues for data loss on clouds, data integration on clouds, cloud energy consumption, big data on clouds, etc. Out of these issues, we are focusing on data loss and energy consumption on cloud. The probability of hardware failures is more due to a large number of nodes in cloud computing system as non-trivial Based on statistical analysis of hardware failures in [6] - [8]. Some hardware failures can damage data on the disk nodes. With therefore, data-intensive applications running cannot read data from disks successfully. To endure the data corruption and to provide high data availability, the data replication technique is widely adopted in the cloud computing system. Data Replication copies a database and also synchronizes a set of replicas so that changes made to one replica are reflected in all the others. The replication enables many users to work with their own local copy of a database but have the database updated as if they working on a single & centralized database. Replication is often the most efficient method of database access for database applications where users are geographically widely distributed. Storage node has limited replication space; Due to which the data replicas of some applications may be stored in lower performance nodes. Data replicas which don't satisfy the QoS Requirements of Data Intensive Applications are called as QoS-violated data replicas [1]. The count of QoS-violated data replicas is expected to be as small as possible to provide QoS on those applications. Due to the heterogeneity of the node in Cloud Computing, the application data with a high quality of service can be reproduced in a low performance node. The low performance node is which has slowness of communications access latencies and disk. On occurrence of data corruption in the node running the application of high Quality of service, the data of the application will be recovered from the low performance node. The QoS requirement and application is defined from the features of the application information. The Data Replication technique to cloud Provides benefits such as faster recovery time, Off-site real time data copies, to store and replicate data without external software etc.The main objective is to minimize the data replication cost and the number of QoS violated data replicas .As the data replication cost minimizes, the data replication can be completed quickly. The main contributions of this paper are summarized as follows

(1) Data replication algorithms consider the QoS requirements of applications.

(2)The QADR problem is formulated as an ILP formulation. Due to considering the computational complexity in solving the ILP formulation, we transform the QADR problem to the MCMF problem to obtain the polynomial time optimal solution. (3)The proposed replication algorithms can accommodate to a large-scale cloud computing system. Node combination techniques are utilized to suppress the computational time of the QADR problem.

II LITERATURE SURVEY

Failures in accessing data under cloud use techniques such as check point and data replication. Occurrences of Name node failure can be tolerated by using this checkpoint technique where the state of the file system namespace has been restored in the disk of NameNode.To protect the stored data blocks in DataNode from failure has been done by Data Replication Technique [1] in the stored data blocks

is protected by data replication technique. H. Kuang ,K. Shvachko, R. Chansler and S. Radia et.al briefly explained the HDFS where the data replication technique has been extensively adopted. HDFS [3] has master/slave architecture which consists of a single NameNode & a master server which manages the file system namespace and regulates access to files by clients and a number of DataNodes, which manages storage attached to the nodes on which they run. Internally, a file is divided into one or more blocks and these blocks are stored in a set of DataNodes. An application can specify the number of replicas of a file and the replication factor can be specified at file creation time and can be changed later. In DataNode, by default, replica factor was taken as two for a single data block. If a data block is written to the DataNode n, the original copy of this data block is stored in the disk of the DataNode n. Two replicas of this data block are stored in two different DataNodes where the rack numbers are different with that of the DataNode n. A. Gao and L. Diao, et al. discussed about the consistency maintenance problem of data replica in cloud. They proposed lazy update method [6] which improved the data access throughput and reduction in response time .

W. Li, Y. Yang, J. Chen, and D. Yuan et.al suggest a mechanism to provide data reliability for the replicated data .It is based on proactive replica checking which is cost effective since it reduces storage space consumption.

X. Tang and J. Xu et.al discussed about QADR problem and proved that it is NP-Complete. Without abusing QoS requirements, the insertion and deletion of data object replicas are done by two algorithms: l-Greedy-Insert and l-Greedy-Delete [8], which results in exceeded execution time. M. Shorfuzzaman, P. Graham, and R. Eskicioglu et.al presented a distributed replica placement algorithm based on dynamic programming for reducing the execution time and it has been done on data grid systems. It has been designed to satisfy QoS requirements by identifying locations of replication to provide data reliability and Performance measure. X. Fu, R. Wang, Y. Wang, and S. Deng et.al addressed replica problem under mobile grid environment for mobile users. They proposed solution by using dynamic programming and binary search problem resulting data availability and high data accessibility. A. M. Soosai, A. Abdullah, M. Othman, R. Latip, M. N. Sulaiman, and H. Ibrahim, et.al described a strategy called Least Value Replacement (LVR), deals about storage constraints and QoS requirements under data grid. Here the storage limitation problem on replica has been overcome by replacement i.e. listing some information such as future values of files and frequent access. QADR problem considers the replication contention among data blocks because of replication storage limitation. Due to which some data replicas that cannot satisfy the QoS requirements of Data Intensive Application [1]. Here the problem of violated replicas arises and the previous work doesn't proceed to minimize this violation. The replicas of some data objects cannot be stored successfully for the server with limited storage space if there are many data object replicas to be placed in the server. In this situation, the unsuccessful data object replicas will be put in other servers without QoS satisfaction. We can now undergo the problem of QADR [8] with various algorithms under cloud environment for data intensive applications.

III. PROPOSED SYSTEM

For data-intensive applications we Propose QoS-aware data replication (QADR) problem in cloud computing environment. The QADR problem considers the QoS requirements of applications in the data replication. This reduces the probability of data corruption significantly before completing data replication. Some storage node has limited replication space, due to which the data replicas of some applications may be stored in lower-performance nodes. This will result in some data replicas that cannot meet the QoS requirements of their applications. These data replicas are known as QoS-violated data replicas. The QoS-violated data replicas are expected to be as small as possible. To solve the QADR problem, we first propose a greedy algorithm, called the high-QoS first-replication (HQFR) algorithm. In this algorithm, if application i has a higher QoS requirement, then that application take lead over other applications to perform data replication. But the HQFR algorithm unable to achieve the above minimum objective. The optimal solution of the QADR problem can be obtained by formulating the problem as an integer linear programming (ILP) formulation. However, the ILP formulation involves complicated computation. To find the optimal solution of the QADR problem in an efficient manner, we propose a new algorithm to solve the QADR problem. This new algorithm transformed the QADR problem to the minimum-cost maximum-flow (MCMF) problem to solve the QADR problem. This existing MCMF algorithm is utilized to optimally solve the QADR problem in polynomial time. Compared to the HQFR algorithm, the optimal algorithm takes more computational time.

3.1 QADR

1. Consider a cloud computing system which runs applications and stores data in a set of storage Node S and the functionality of Nodes are based on HDFS.
2. The running application writes a data block b to the disk of r, where r ∈ S, a request has been sent from r to make a number of copies of replica of b to the disks of other nodes. Many concurrent replication requests issued from different nodes at a certain time instant. Each node cannot store too many data replicas from other nodes due to space limitation.
3. The replicated data for block b will be stored at q(q∈ S) as dr. This data replica dr is related with RC and AC i .e. Replication cost and access time respectively with desired access time T.
4. If there is any data corruption and the original data b cannot be read successfully, the node r tries to get the data from the replication which is stored in q as dr. And the dr is termed as a QoS violated data replica if AC > T.
5. The QADR Problem in cloud tries to minimize the data replication cost and the count of QoS violated data replica using optimal replica replacement strategy to achieve the objective.

3.2 HQFR algorithm

As its name indicates high QoS first replication algorithm. The main thing is that we consider the requirement of QoS aspect of the application, information and access time only. HDFS in the data is divided into data blocks of 64 Mo. The replication factor is two HDFS. There are two numbers of copies of different block than the original data. And two copies are stored on different Data Nodes or different data carriers. It keeps track of all replicas other than the original copy and they mounted on different data carriers to avoid failure of the rack. The basic idea of the algorithm: As the name suggests applications with high quality of service must be replicated first. According to our knowledge of the application of high quality service have stricter requirements in time to an access time to data than normal response applications. High quality of service implementation requirement should take precedence over the requirement of low demand quality of service to perform data replication.

3.3 Optimal replica replacement

In optimal replica replacement algorithm the set of requested nodes which is indicated as Sn and the output will be QoS-violated data replicas. By applying existing polynomial- time MCMF algorithm, it is easy to obtain the MCMF solution of the network flow graph. Here we are using node combination techniques such as rack-based combination and equivalent state combination to avoid the large computational time.

This algorithm concerns energy consumption in nodes of cloud computing techniques. As there are many storage nodes in cloud computing systems we consume energy by reducing the number of nodes. In addition to it, the Energy Efficient Storage Node Identification Technique *(EESNIT)* is proposed to reduce energy consumption in transport and switching process involved.

IV PERFORMANCE EVALUATION

We used MatLab to evaluate the performance of the proposed replication algorithms in a large-scale cloud computing system. Our simulation experiments were conducted by assuming 3,500 nodes in a cloud computing system. The HDFS cluster at Yahoo! includes about 3,500 nodes [10]. We assume that there are 100 racks in the cloud computing system, and each rack is equipped with one single switch. These 100 racks are randomly distributed over a 1000×1000 unit square plane. A rack occupies a 10×10 sub-square plane. For any two racks, there is no intersection area in their corresponding sub-square planes. Among the 100 racks, one is specified as the central rack to organize all other racks as a tree topology with the height about 10. These 3,500 nodes are randomly deployed within the 100 racks after forming the 100 racks with the tree structure connectivity. For two nodes in the same rack, their locations are within the occupied sub-square plane of the rack. In each node, the available replication space is represented as the maximum number of data block replicas allowed to be stored. It is set by randomly selecting a number from the data block interval of [0, 50].Similarly, a QoS interval is also used to set the QoS requirement of an application in the requested node. The lower bound of the QoS interval is the time to access a data block from the local disk of requested node. The upper bound is the largest access time for requested node to retrieve a data block replica from another node.

4.1Simulation Result

To solve the QADR problem, HQFR algorithm and the optimal algorithm is proposed by transforming the QADR problem into the MCMF problem.The optimal algorithm is also called as the MCMF Replication (MCMFR) algorithm. Node combination techniques are also applied in the algorithm for considering the computational time of the MCMFR algorithm. The new MCMFR algorithm is named as C MCMFR algorithm. In this section, we demonstrate the performance results of the HQFR, MCMFR, and C MCMFR algorithm. In addition to these three algorithms, the random replication algorithm and Hadoop replication algorithms were also evaluated in simulation experiments. The random replication algorithm randomly places the replicas of a data block at any nodes. Figure 1 shows the total cost of replication for different numbers Application nodes 500 to 2500. In Fig. 1 (a), the cloud computer system is configured with 9 (3 x 3) different types the heterogeneity of the device using the first access to three disks and transmission rates of Table I. In the Fig. 1 (b), all performance the values in Table I are used to generate the diversity of the device 36 (6 × 6) different types. As seen from Fig. 1(a) and 1(b), the total cost of replicating all increases with algorithms the required number of nodes. The RF replication factor is set 2. In fact, the replication algorithm adopts Hadoop randomly to place a replica data block manner, but it also considers the failure of unity possible. As a result, the total cost of replicating the replication algorithm Hadoop is similar to the algorithm of random replication. Both algorithms do not take the QoS requirements of applications in data replication. These two algorithms have greater replication costs that replication algorithms proposed. From Fig. 1 (a) and 1 (b), we can also see that if the device performance is more diverse, our replication algorithms can further reduce the cost of replication and random Hadoop algorithms. As shown in Fig. 1 (a), the total cost of Hadoop replication algorithm is about 2.47 times that of the MCMFR algorithm. However, in Fig. 1(b), the total replication cost ratio between the two algorithms is about replication 3.79: 1. For the replication algorithms proposed, the MCMFR algorithm can reduce the total cost of replication HQFR algorithm by approximately 29% and 44% in Fig. 1 (a) and 1 (b), respectively. Although the algorithm reduces MCMFR C computation time, it cannot reduce the cost of replication. In the algorithm C MCMFR the QoS violated replicas of data are stored randomly selecting a node from storage unskilled accumulate node. However, in the algorithm MCMFR it also reduces the total cost of data replication QoS violated replicas in addition to data replicas QoS satisfied .Therefore, the MCMFR algorithm has a smaller total replication cost than the C MCMFR algorithm. Compared to the MCMFR algorithm, it increases 21% and 36% of replication cost in Fig. 1(a) and 1(b), respectively.

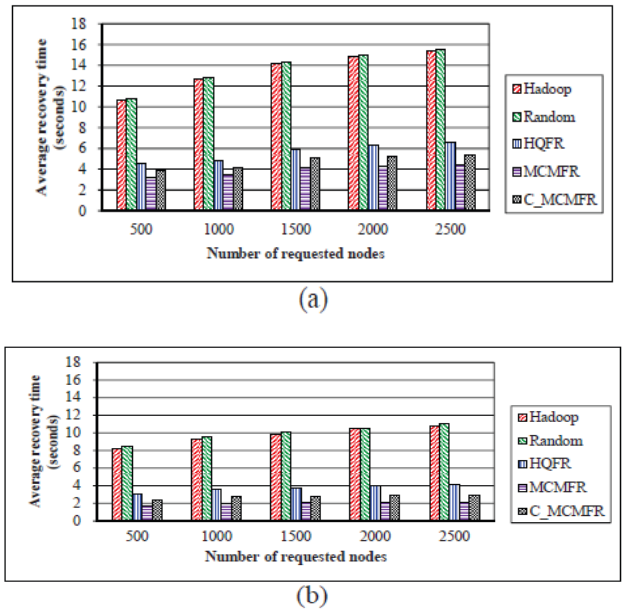


Figure 1- Total replication cost under various device performances (a) 9 types (b) 36 types.

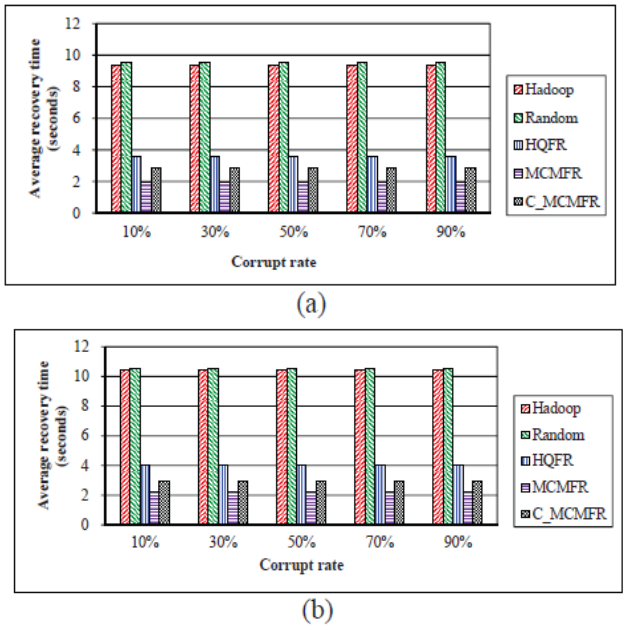


Figure 2 Average recovery time under various device performance (a) 9 types (b) 36 types.

Fig. 2 shows the comparison of the average recovery time for a corrupt data block. If the requested node cannot read a data block from its disk due to data corruption, how much time is taken by requested node to retrieve one replica of the data bock from another node? In Fig. 2(a) and 2(b), the MCMFR algorithm has the smallest average recovery time, which can improve the average recovery time of the Hadoop algorithm by about 71% and 79%, respectively. From Fig. 2(a) and 2(b), we also see that the average recovery time increases with the number of requested nodes in the proposed replication algorithms. The replication contention probability has an upward growth trend as increasing the number of requested nodes. Due to the limited replication space in a qualified node, this node may not serve the replication requests from all its correspondingly requested nodes. As a result, some requested nodes cannot select their best qualified nodes to store their data block replicas. Later, if such requested node reads a corrupt data block, it may take more time to retrieve the data block replica.

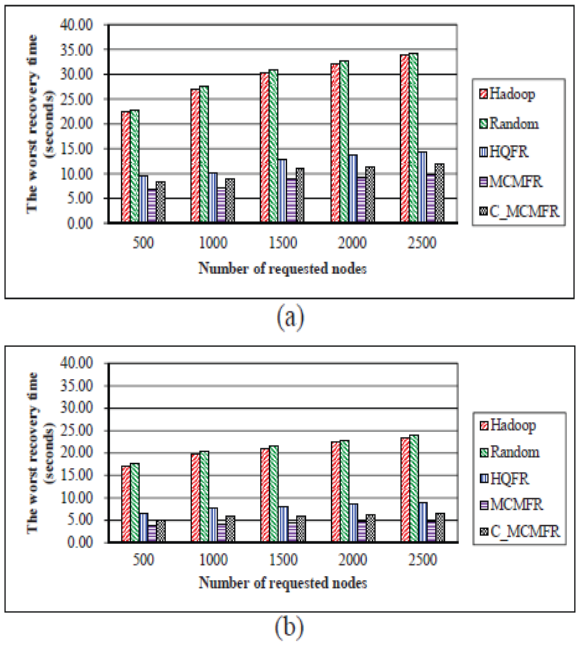
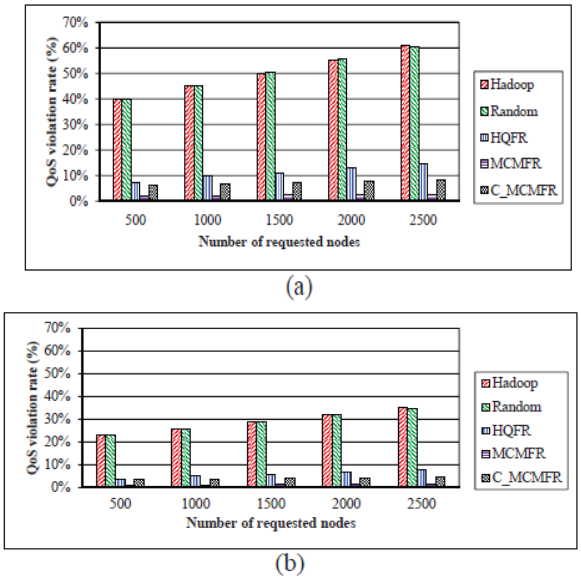


Figure 3 The numbers of QoS-violated data blocks under various device performance (a) 9 types (b) 36 types.

Fig. 3 shows the comparison of the QoS violation ratios in the above concerned algorithms QoS violation ratio [1] is the ratio of total number of QoS-violated data block replicas to total number of data block replicas. In the Hadoop and random algorithms, the QoS requirement of an application is not considered in the data replication. In Fig.3(a) and 3(b), the QoS violation ratios of these two algorithms are approximately 50% and 28%, respectively. The QoS requirement is considered in the proposed replication algorithms. As mentioned above, the QoS-violated data replicas are generated due to the limited replication space of a node. In addition to minimizing the replication cost, the MCMFR algorithm can also minimize the number of QoS violated data replicas. Compared to the HQFR and C MCMFR algorithms, the MCMFR algorithm can reduce at least 78% and 67% of QoS-violated data replicas. Note that the main advantage of the C MCMFR algorithm is in reducing the computation time of the QADR problem.

For concerning the scalable replication issue, rack-based and equivalent state combination techniques are utilize to reduce the execution time in solving the QADR problem. For this characteristic enhancement , we perform the execution time comparison among different replication algorithms, as shown in Fig. 4.



(a)

(b)

Figure 4 Execution time of the four replication algorithms (a) 9 types (b) 36 types.

With considering the QoS requirement in the data replication, the proposed replication algorithms take more execution time. Compared to the Hadoop algorithm, the HQFR algorithm increases at least 1.24 times of the execution time. For the MCMFR algorithm, its execution time is about 6.4 times of that of the HQFR algorithm. However, if the rack-based combination and the equivalent state combination techniques are applied in the MCMFR algorithm, the execution time of solving the QADR problem can be reduced significantly. This can be obviously seen from the execution time of the C MCMFR algorithm in Fig. 4. It is about 1.26 times of that of the HQFR algorithm. From Fig. 4, we can also observe that the execution time of the C MCMFR algorithm has not a linear increase with varying the number of requested nodes. The reason is explained as follows. Using the rack-based and equivalent-state combination techniques, the requested and qualified nodes can be respectively combined as a smaller number of group nodes. In simulation experiments, the number of formed group nodes is at most 100 since there are 100 racks in the referred cloud computing system. Therefore, the execution time of the C MCMFR algorithm cannot be large even if there are a large number of requested (qualified) nodes.

V CONCLUSION:

To solve the QADR problem, the device heterogeneity is also considered in addition to the QoS requirements of applications. Two replication algorithms have been proposed.We have solve the minimum objectives of the existing system by preceding the QADR problem. The data replication cost and the number of QoS-aware data replicas cannot be minimized. We optimally solve the QADR problem in polynomial time by transforming the QADR problem to the MCMF problem. We also present node combination techniques to handle the scalable replication issue of the QADR problem. This technique is more useful for the users those who are working under cloud as it contains many storage nodes. Here we solved the problem of QoS requirement of an data intensive application .

REFERENCES

[1] QoS-Aware Data Replication for Data Intensive Applications in Cloud Computing Systems, Jenn-Wei Lin, Chien-Hung Chen, and J. Morris Chang.

[2] Apache Hadoop Project. [Online]. Available: http://hadoop. apache.org.

[3] E. Pinheiro, W.-D. Weber, and L. A. Barroso, “Failure Trends in a Large Disk Drive Population,” in Proc. 5th USENIX Conf. File and Storage Technologies, Feb. 2007, pp. 17–28.

[4] F. Wang, J. Qiu, J. Yang, B. Dong, X. Li, and Y. Li, “Hadoop High Availability through Metadata Replication,” in Proc. 1st Int. Workshop Cloud Data Manage., 2009, pp. 37–44.

[5] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The Hadoop Distributed File System,” in Proc. IEEE 26th Symp. Mass Storage Systems and Technologies (MSST), Jun. 2010, pp. 1–10.

[6] A. Gao and L. Diao, “Lazy Update Propagation for Data Replication in Cloud Computing,” in Proc. 2010 5th Int. Conf. Pervasive Computing and Applications (ICPCA), Dec. 2010, pp. 250–254

[7] W. Li, Y. Yang, J. Chen, and D. Yuan, “A Cost-Effective Mechanism for Cloud Data Reliability Management Based on Proactive Replica Checking,” in Proc. 2012 12th IEEE/ACM Int. Symp. Cluster, Cloud and Grid Computing (CCGrid), May 2012, pp. 564–571

[8] X. Tang and J. Xu, “QoS-Aware Replica Placement for Content Distribution,” IEEE Trans. Parallel and Distrib. Syst., vol. 16, no. 10, pp. 921–932, Oct. 2005.

[9] S. Ghemawat, H. Gobioff, and S.-T. Leung, “The Google File System,” in Proc. 19th ACM Symp. Operating Systems Principles, vol. 37, no. 5, Dec. 2003, pp. 29–43.

[10] IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges, IEEE 802.1D Std., 2004.

[11] M. Shorfuzzaman, P. Graham, and R. Eskicioglu, “QoS-Aware Distributed Replica Placement in Hierarchical Data Grids,” in Proc. 2011 IEEE Int. Conf. Advanced Inform. Networking and Applicat., Mar. 2011, pp. 291–299.

[12] H. Wang, P. Liu, and J.-J. Wu, “A QoS-Aware Heuristic Algorithm for Replica Placement,” in Proc. 7th IEEE/ACM Int. Conf. Grid Computing, Sep. 2006, pp. 96–103.