International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 6, 1420–1433, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.6-112

Optimization of Replica Consistency and Conflict Resolution in Data Grid Environment

Priyanka Vashisht
Department of Computer Science and Engineering,
North Cap University, Gurugram, India
E-mail: priyanka.vashisht@gmail.com

Vijay Kumar
Department of Applied Mathematics,
Amity Institute of Applied Sciences, Amity University, Noida, India
Corresponding author: vijay_parashar@yahoo.com

Rajesh Kumar
Department of Computer Science and Engineering,
Thapar University, Patiala, India
E-mail: rakumar@thapar.edu

Anju Sharma
Department of Computer Science and Engineering,
Panjab Technical Institute of Technology, Rajpura, India
E-mail: phdanju@gmail.com

(Received February 21, 2019; Accepted September 12, 2019)

Abstract
Data replication is widely used mechanism aiming at ensuring efficient access, improving performance and providing quality data in data grids. The properties of file and their replicas can vary with time resulting to inconsistency. In this paper, an agent-based replica consistency approach, namely, Replica Consistency and Conflict Resolution (RCCR) has been proposed to improve the efficiency of the system. RCCR is hybrid approach which is used to address the problem of inconsistency in multi master environment. Additionally, the conflict resolution approach is introduced based on vector clock and version of the replica. Finally, simulated results are presented and compared with existing consistency approaches such as Pessimistic, Optimistic and One-way Replica Consistency (ORCS). The simulated results depicts that proposed RCCR approach performs better when writeable replicas are more likely to occur.

Keywords- Replica consistency, Conflict resolution, Data grid, Pessimistic approach, Optimistic approach.

1. Introduction

Distributed environment like data grid rely on sharing data and resources between various scientists and researchers to execute common tasks (Guroob and Manjaiah, 2016). Scientific applications such as high energy physics, data mining, and satellite images processing, and climate simulation, generates large amounts of data which is stored and managed by data grids (Pérez et al., 2010) The management and access to huge volume of data located at various distributed locations across the data grid, degrades due to certain network restrictions. Replication helps in maintaining and optimizing access time and the availability of data in distributed environment. Replication provides a means for allocating similar data at various distributed locations in data grid (Mansouri et al., 2013). Despite advantages of data replication technique,

the main challenging issues are consistency maintenance and conflict resolution of geographically distributed data. Replica consistency can be achieved using pessimistic or optimistic approaches in distributed environment (Vashisht et al., 2017; 2019). Pessimistic technique provides single copy serialization for achieving consistency in datasets. One copy serialization uses locking technique for propagating updates across distributed files, as a result large number of replication and frequent updates decreases the performance of pessimistic approach. This is the reason why pessimistic technique is not suitable in distributed environments such as grids and clouds. In optimistic replication technique, relaxed approach is used for propagating the updates to replicas. Delayed updates propagation helps in improving the write access even in the presence of unreliable network link. In optimistic replication the performance of the system decreases when frequency of the write request is very high and problem will worsen as the magnitude of data grid grow. Moreover, if the updates are made concurrently in multi-master environment, issue of conflict arises among the replicas. Therefore, the key issues in data grids is how the file and its replicas can be used efficiently and effectively to reduce inconsistencies and resolving conflicts.

Main contribution of this work consists of proposal of consistency management approach in large scale distributed system, which uses hybrid approach to take advantage of both optimistic and pessimistic approach. The optimistic approach enhances the performance of the system whereas the quality of service is assured by pessimistic approach. The contribution of this research has two essential aspects:

(i) An agent based hybrid approach is followed for maintaining the consistency among replicas, which rely on access frequencies of the file.
(ii) The conflicts are handled using agents at various regions in multi master environment.

The rest of the paper is summarized as follows: Section 2 contains historical background of earlier studies. Section 3 presents the fundamental concepts about Replica Consistency and Conflict Resolution (RCCR) Strategy, working of RCCR and algorithm of RCCR strategy. Section 4 describes parameters evaluation and Experimental Results. Finally, conclusion is drawn in Section5.

2. Related Work

Yang et al. (2010) have suggested a model namely, One-way Replica Consistency (ORCS). In ORCS nodes are organised in hierarchical manner. Consistency of the files is maintained by automatically propagating updates from highest level to lowest level. The simulations demonstrated significant stability in system performance and maintaining consistency among replicas. Kraska et al. (2009) presented a consistency model that separates the data into three categories i.e. A, B, C. Data with highest level of consistency is reserved for category A. Category B select the level of consistency dynamically based on calculated threshold value. Data having weak consistency is enclosed in Category C and D. Dullmann et al. (2001) suggested a sophisticated service for maintaining consistency among replicas, called Grid Consistency Service (GCS). The work suggested numerous consistency levels extending from completely synchronized to loosely synchronized data based on the information regarding data and their use cases. Various levels of consistency have been presented to user with several level of guarantee for consistency and their impact on performance. Radi (2014) proposed an Improved Aggressive Update Propagation (IAUP) approach based on push protocol, where updates are transmitted to all copies by the nodes containing the file. IAUP is beneficial when replicas need to agree to take an evident degree of consistency. The read to update ratio of IAUP is comparatively higher than

pull mechanism. The average response time of IAUP is marginal with aggressive update propagation (AUP) and better than lazy update propagation (LUP) whereas, it provides consistent data all the time. Grace and Manimegalai (2014) suggested that in the course of replication of data, identical copies of the files are formed then positioned at suitable nodes and selection can be done while executing the job. At any time, write operation is accomplished on replicas of file; it must be propagated to all existing replicas to ensure consistency of data on the data grid. Chang and Chang (2006); Chang and Chang (2008) offered Adaptable Replica Consistency Service (ARCS) which employ access weight of file in order to achieve efficient load balance and consistent data in grid system. Replica Consistency Decision is done using Naive Bayesian Classifier (RCDM-NBC). The experimental results show that ARCS is better than aggressive and lazy protocols in terms of average file access delay and total job time. Whereas, RCDM-NBC are more accommodating to a dynamic nature of grid environment which takes into account the adaptability and flexibility of dynamic system. Chihoub et al. (2012) suggested a three-fold cost effective method for maintaining consistency among files in cloud environment. A performance modelling scheme is established that dynamically assess the application and find its consistency requirements. The overall system performance has enhanced by maintaining desired consistency and reducing economic cost for user. Abawajy and Deris (2013) introduced a novel quorum-based data replication protocol called Data Duplication on Grid (DDG) with the objectives of minimizing the data update cost, providing high availability and data consistency. DDG works in two phases, Firstly, it takes care of replica placement then consistency of various distributed replicas is maintained. The experimental results shows that the proposed algorithm has reduced the data replication and communication cost. Guroob and Manjaiah (2016) proposed a novel approach, namely, Efficient Replica Consistency Model (ERCM) for propagating updates in distributed grid environment. The approach has two major offerings: Firstly, it reduces response time for both read and write operations. Secondly, the management of replica consistency and acceleration of update propagation is done. The simulation results demonstrate that in ERCM execution time for both operations are reduced whereas availability of file has improved as compared to other state of art. Fetai and Schuldt (2012) used strong and weak consistency approach to introduce an adaptive and dynamic concurrency approach namely, C3. Multi-master approach is used for ensuring consistency according to various applications in cloud environment. Dynamic evaluation of consistency cost is calculated depending on weak and strong consistency strategies by defining certain set of rules for choosing the best possible level of consistency with minimum cost. The results depict that transactions that adaptively adjust thresholds in C3 approach outperform the transactions that have static mode in terms of cost and performance. Belalem et al. (2008) uses hierarchical structure with two levels for consistency management of replica. Lower level provides physical storage to the replicas. At higher level the replica consistency management is provided. The results demonstrate gain of the proposed approach over the existing pessimistic and optimistic approaches. Belalem and Slimani (2007) used multi-master approach for conflict resolution in grid environment. Experimental results proved that time of communication among the sites for achieving consistency has reduced substantially; the effectiveness of consistency management has increased and model provide resistance and adaptively to changes in large systems. Further, the model is enhanced by introducing the economic approach which uses English and Dutch auction methodology for resolving the conflicts among the replicas (Belalem et al. 2009).

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 6, 1420–1433, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.6-112

3. Replica Consistency and Conflict Resolution (RCCR) Strategy

The main goal of proposed approach is to provide efficient results by handling inconsistencies and resolving conflicts among replicas. The proposed RCCR approach follows the steps mentioned below during the execution of the request made by the client.

(i) *Request*: A client submits a request for a file f_{ij}. Request can be either read only or write anywhere.

(ii) *Execution*: Request is initially handled by the master node (node responsible for handling task globally) as shown in Figure 1. Request is analysed by the agent of master node for mapping it to appropriate region. Further, the region head is responsible for deploying a particular resource to the request. Resource utilization of each node is monitored and stored at local database.

(iii) *Response*: After execution of the request, the node responds immediately to the head node (the node responsible for handling task locally) which in turn send message to the master node. The master node is responsible for providing the response to the client.

(iv) *Coordination and Agreement*: In this step the agent on the head node will update the corresponding information and propagates the results to the other nodes containing replicas of the same file. During this phase, the operation is scheduled for execution at all relevant nodes within the region, and same is updated with the master node. The head node will ensure if there is any inconsistency with in a region, it will propagate the updates to corresponding replicas by electing leader node of the region. The leader node is a node of region with highest number of updates. On the other hand the master node will check whether the replicas are convergent, divergent or there exists some conflicts. Convergence indicates that two replicas are in consistent states, divergence signifies that two replicas are different by only one component. Two replicas are in conflicting state, if at least two or more components of a file are different.

(v) *Conflict handling*: If no conflicts are detected during the coordination and agreement step, updates are integrated permanently into the GIC (Global Information Collector). GIC is an agent which is responsible for collecting information from various regions and stores it in Global Database. If conflicts are detected, they must be handled mutually by exchanging information among various master nodes.

3.1 Working of RCCR Strategy

A hierarchical and distributed network structure has been implemented which offers the system with flexibility, scalability and tolerance of certain faults. The work is divided into two parts, first the local level consistency is maintained at region and sub-region using head node. Secondly, the conflicts and consistencies are checked among multi-masters of various regions to ensure global consistency. A two-level architectural view of RCCR strategy is presented in Figure 1 and the functional view of RCCR strategy is presented in Figure 2.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 6, 1420–1433, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.6-112

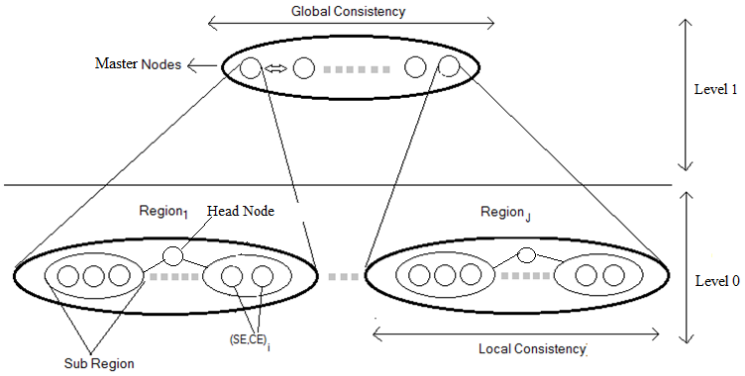

Figure 1. High level view of RCCR approach

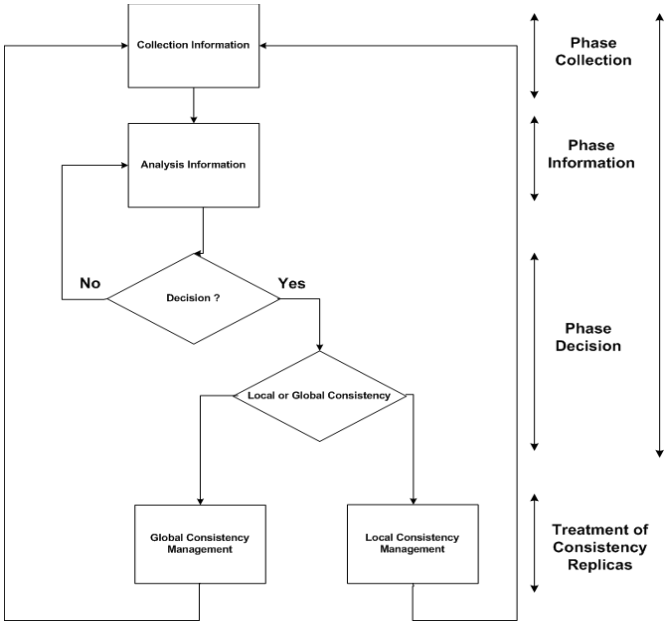

Figure 2. Functional view of RCCR strategy

At *Level 0* group of nodes form a region or sub-regions. The node contains number of replicas of several file. Each replica at level 0 contains some additional information such as (VV_{ijkp}, $timestamp_{ijkp}$, rep_{id}) where, VV_{ijkp} is a vector for storing version of replica *p* of file *k* located at node *i* of region *j*, $timestamp_{ijkp}$ is timestamp for recent update to keep track of local update and rep_{id} is unique Id of replica. At *level 0* the consistency is achieved with in the region or at local level. To achieve the local level consistencies following steps are needed:

(i) *Collection of Information*: The agents on the head node start collecting the information of all the nodes that belongs to a region. The collected information is:

- Number of replicas of a file f_i in the region.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 6, 1420–1433, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.6-112

- Number of requests for file f_i in the region.
- Latest version (Ref_{VV}) and timestamp (Ref_{time}) associated with each replica r_{ik} of a file f_i that exist in the region.

(ii) *Selection of Leader Node*:

- Leader node is selected by analysing the information collected in the previous step.
- Replica having the latest version (Ref_{VV}) and timestamp (Ref_{time}) is elected as the leader node.

(iii) *Propagating the Updates*:

- The elected leader node is responsible for propagating the updates to other replicas of same file in the region.
- The updated replicas are added to the list of leader and start propagating the updates to other replicas of the sub-region.
- When all replicas in the sub-region are updated, then the leader nodes propagate updates to various sub-regions of the region. The multiple leader nodes propagate the updates in parallel, hence reducing the update time.

At *Level 1*, a node, namely, master node acts as a representative for that region. The master node is responsible for converging all the replicas to a consistent state in the entire grid system. The global consistency among the regions uses optimistic approach whereas the local consistency at head node follows pessimistic approach. Several steps are considered while achieving the global consistency

(i) *Selection of Representative*: A master node is selected among the pool of available master nodes on the basis of maximum number of updates. Here multi-master approach is used.

(ii) *Collaboration*: A representative master node will collaborate with other master nodes for propagating the updates of the replica.

(iii) *Detection and Resolution of Conflicts*: Here, divergent and conflict are two cases for representing the inconsistencies. If the amount of inconsistency among two replicas is small then it is divergent else there exist certain conflicts among two replicas of different regions.

Case I

- If Divergence is there in version and timestamp between two regions.
- Identify the missing operation.
- Propagate the updates for missing operations.

Case II

- If Conflict occur among replicas of a file at are located in different regions.
- Mutual exchange of information is done by selecting representative at global level.

3.2 Algorithms for RCCR Strategy

The user input requests to grid system. Whenever a request is made a counter, namely, *enu* is incremented by one, to keep track of access frequency of the file. Now the request requirements are stored on master node for analysing, scheduling, and allocation of file resources. The request

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 6, 1420–1433, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.6-112

are analysed based on the access frequency of the file then it is scheduled to an appropriate region. Accordingly, the global database is updated. At region level, the region head is responsible for deploying the request on an appropriate node. After allocation of appropriate node to a task, the resource utilization of node is monitored and the observed information is recorded on local database. Moreover, the reference vector version Ref_{VV} and reference timestamp$_{Reftime}$ is also updated as shown in Algorithm 1. The request is executed and results are sending back to client. For dealing with inconsistency at local level the head node will elect a leader node as depicted in Algorithm 2. The leader node is selected on the basis of latest timestamp and vector version of the file as shown in Algorithm 3. The leader node propagates its update to other replicas within a region. The updates are propagated among regions with leader nodes. Local level consistency follows optimistic strategy so that no divergence or inconsistency exists among replicas. Due to multi-master approach the client can make request to any master which may lead to have conflicts at global level. If conflicts are detected, then the conflict resolution mechanism is invoked. In this work, the conflicts are categorized as Replica Conflict and Update Conflict. Two updates *u* and *u0* to a file f_i are in conflict, if they are performed concurrently on a replica r_{ij} located on some nodes. There is an update conflict state if neither of the update has observed the effects of the other before executing. The update conflict has a consequence which results into replica conflict. If the updates are applied to different replicas r_{ij}, r_{ik} of a single file f_i, they will represent different versions of f_i. This make replicas r_{ij} and r_{ik} in conflicting state after the updates have been applied. However, when a single update *u* is applied to a replica r_{ij} of file f_i, it does not lead replicas in conflicting state. Though the state of r_{ij} differs from the state of another replica r_{ik} until update *u* has been applied to r_{ik}. As in optimistic approach the state of two replicas can vary up to a certain time but eventually achieve consistency with due course of time. To resolve the problem of update conflicts a logical time-stamping is used, while the replica conflicts are taken care by using version vectors and is depicted in Algorithm 3.

4. Evaluation Parameters and Experimental Results

The efficiency of RCCR is considered based on five performance metrics percentage of accessing consistent data, amount of update transaction, retired request rate, conflict rate and mean job execution time. RCCR is compared with optimistic, pessimistic and ORCS approaches on *Optorsim* simulator using with varying parameters as shown in Table 1. During simulation pessimistic approach keeps entire replicas consistent all the times. In optimistic approach, consistency takes place after a small interval. In ORCS, consistency depends on the request of the user and RCCR uses hybrid approach for maintaining the consistency and resolving the conflict within the replicas.

Algorithm 1 Pseudo code for Consistency Maintenance	**Algorithm 2** Pseudo code for Electing Leader Nodes
1: **procedure** *Consistency maintenance* 2: **input**: Replica Location 3: $Ref_{VV} \leftarrow \emptyset$ 4: $Ref_{time} \leftarrow \emptyset$ 5: **if** read/write request received **then** 6: [*enu*] ← [*enu*] + 1 7: execute operation on n_{ij} 8: **if** read operation **then** 9: **return** (result) 10: **else** 11: $Ref_{VV} \leftarrow V\,V_{ijkp}$ 12: $Ref_{time} \leftarrow timestamp_{ijkp}$ 13: **end if** 14: **end if**	1: $node_{led} \leftarrow 0$ 2: **for** *all* n_{ij} *in the region* **do** 3: **if** $(VV_{ijkp} = Ref_{VV})$ **V** $(timestamp_{ij} = Ref_{time})$ **then** 4: $node_{led} \leftarrow n_{ij}$ 5: **end if** 6: $V\,V_{ijkp} \leftarrow V\,V_{ijkp} \mathbf{U}\ Ref_{VV}$ 7: $timestamp_{ijkp} \leftarrow timestamp_{ijkp} \mathbf{U}\ Ref_{timestamp}$ 8: $node_{led}$ + + 9: **end for** 10: **return**(*consistent replicas*) 11: **end procedure**

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 6, 1420–1433, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.6-112

Algorithm 3 Pseudo code for Removing Conflicts

1: Multicast (Ref_{VV}, Ref_{time})
2: **if** Convergence $((VVijkp, Ref_{VV})\ \boldsymbol{V}\ (timestamp_{ij}, Ref_{time}))$ **then**
3: convergence, no propagation
4: **else**
5: **if** Divergence((VVijkp ,Ref_{VV}) $\boldsymbol{V}$ (timestamp$_{ij}$,Ref_{time}) **then**
6: Divergence ← missing operations
7: $VV_{ijkp} \leftarrow VV_{ijkp} \boldsymbol{U}\ Ref_{VV}$
8: $timestamp_{ijkp} \leftarrow timestamp_{ijkp} \boldsymbol{U}\ Reftimestamp$
9: **else**
10: **if** Conflict($(VV_{ijkp}, Ref_{VV})\ \boldsymbol{V}\ (timestamp_{ij}; Ref_{time})$ **then**
11: Mutual exchange of information by selecting leaders at global level
12: **end if**
13: **end if**
14: **end if**
15: **return** (consistent replicas)
16: **end procedure**

4.1 Percentage of Accessing Consistent Data

As compared to the Pessimistic consistency approach, during propagation phase, the number of update message decreases considerably in proposed approach. When RCCR strategy is implemented, the performance of the system increases significantly. The update actions on replicas are not frequent due to hybrid approach followed in RCCR. The replica consistency approach is applied from time to time so that the probability of inconsistent consequences can be dealt. Consequently, system attains a balance between consistency, performance, and availability. The percentage of accessing consistent data is calculated as:

$$P_{cons} = \frac{Num_{con}}{Num_{f_i}} \times 100, \quad (1)$$

where, Num_{con} represents number of consistent replicas of file f_i at time τ and Num_{fi} is total number of replicas of file f_i at time τ. Figure 3 shows the evaluation of the percentage of reading up-to-date data at each predefined checkpoint time interval τ between Pessimistic, Optimistic, ORCS and RCCR consistency approaches.

Table 1. Parameters used in simulation

Parameter	Value
File size	10 MB - 1GB
Number of files	1000
Number of jobs	100
Network latency	10ms-20ms
Storage media latency	2ms-5ms
Transfer rate	50 Mb/s
Storage capacity	10PB

Pessimistic approach almost guarantees that every single request for read is satisfied by accessing the updated data. Optimistic and ORCS approaches guarantee weaker consistency, so the

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 6, 1420–1433, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.6-112

percentage of accessing updated data is lower than Pessimistic approach. In RCCR strategy the guarantee for consistency is guarantee for consistency is better than Optimistic strategy whereas, ORCS is marginal with Synchronous consistency strategy.

4.2 Amount of Update Transactions

In RCCR algorithm, number of update transaction decreases significantly as compared to Pessimistic consistency. The number of operations for data update has reduced, consequently enhances the performance of the system considerably. The amount of update transaction is calculated by:

$$Up_{tran} = number\ of\ write\ request\ executed\ for\ replica\ r_i \in f_i\ at\ time\ \tau \quad (2)$$

In Figure 4, comparison of total amount of updates for Pessimistic, Optimistic and proposed RCCR approach has been done. As time goes by, a number of transactions for Pessimistic consistency approach increases rapidly, whereas, in Optimistic consistency approach, the volume of transactions increases gradually. A number of transactions for proposed RCCR algorithm exceed the total transactions for Optimistic approach marginally. As compare to Pessimistic consistency, the number of transactions for RCCR algorithm reduces significantly. Also, the RCCR consistency approach decreases cost of update transaction significantly, while the requirements of application for consistency are fulfilled. Moreover, RCCR consistency approach assures higher level of consistency than Optimistic consistency while the transaction cost increases marginally. Consequently, a better balance between consistency, availability, and performance is achieved.

4.3 Average Execution Time

Average Execution Time is calculated as ratio of time taken for complete execution of a request, time spent by the request in waiting queue and the total amount of requests. It can be depicted as:

$$Meanjob\ Execution\ Time = \frac{\sum_{n=1}^{i}(T_i + W_i)}{n} \quad (3)$$

where, n is amount of jobs submitted to the system, T_i is time taken by the system to complete the i^{th} job and W_i is total time spent by i^{th} job in waiting queue. The RCCR along with Pessimistic and Optimistic approach was tested using 100, 200, 300, 400 and 500 numbers of requests. The comparative analysis of RCCR, Pessimistic, Optimistic and ORCS is presented in Figure 5. It is apparent that execution time of RCCR is lowest as the number of requests increases. The time to access a file has reduced in RCCR, which moderates the waiting time. The Pessimistic approach performs worst in all the cases whereas, RCCR performs better than Optimistic and ORCS approaches.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 6, 1420–1433, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.6-112

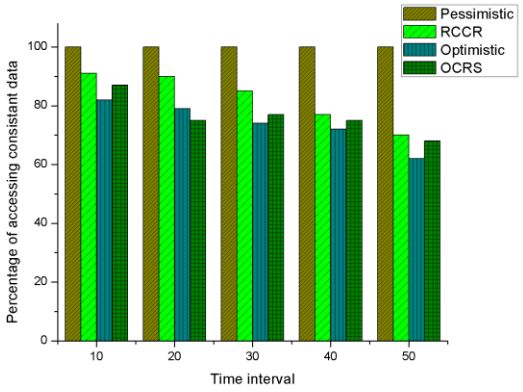

Figure 3. Percentage of accessing consistent data

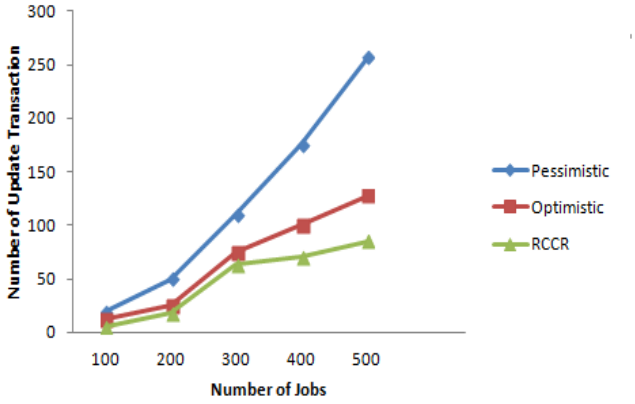

Figure 4. Number of update transactions

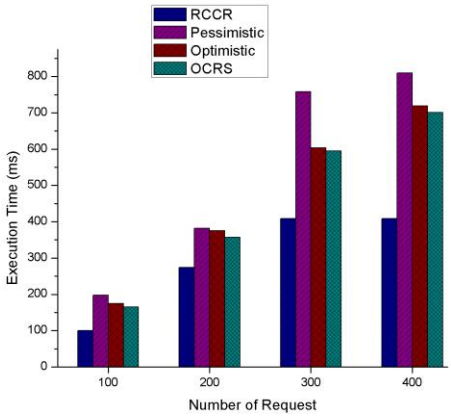

Figure 5. Average execution time

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 6, 1420–1433, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.6-112

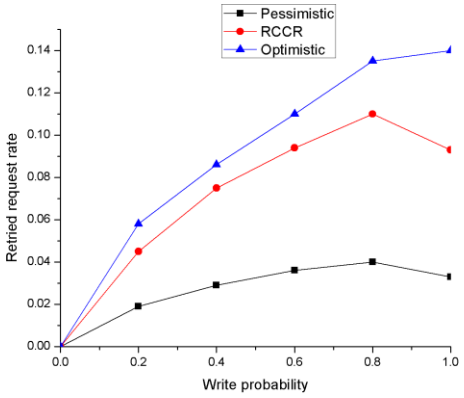

Figure 6. Comparison of retried request rate

4.4 Retired Request Rate

Retired Request rate is defined as number of requests than have been reissued due to conflict upon the total number of requests made for the file. As compared to Optimistic approach the proposed RCCR approach has shown an improvement, whereas, the results are marginal to Pessimistic approach. The Retired Request rate Ret_{RR} is calculated as:

$$Ret_{RR} = \frac{R_{ret}}{Total_{req}} \tag{4}$$

where, R_{ret} is defined as amount of reissued requests after failure and $Total_{req}$ is the total number of requests made by the client for a file. However, $Total_{req}$ is calculated as:

$$Total_{req} = r_k + w_k \tag{5}$$

where r_k and w_k are read and write requests. Figure 6 shows the evaluation of the retired request rate with respect to various write probability of the file. For the simplicity, an assumption has been made that all files have same write probability. The simulation is done in batches, each batch having different write probability. Pessimistic consistency approach guarantees that the Ret_{RR} of each request is minimum. The number of retired requests rate increases as the number of write request for a file increases. The proposed RCCR approach shows a significant improvement by lowering the number of reissued requests as compared to Optimistic approach. Retirement rate of the request is not increasing much with respect to amount of write requests. It provides better handling of retired requests as compared to Optimistic approach.

4.5 Conflicts Rate

The parameter is defined as number of time stale data of a file come across when the update requests is made over a period of time. The conflict rate is calculated as:

$$Conf_{rate} = \frac{Conf_{num}}{W_{rep}} \tag{6}$$

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 6, 1420–1433, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.6-112

where, $Conf_{num}$ is number of conflicts and W_{rep} is number of write request of a replica over a period of time. W_{rep} is calculated as:

$$W_{rep} = \frac{Total_{rep} - r_k}{\tau} \quad (7)$$

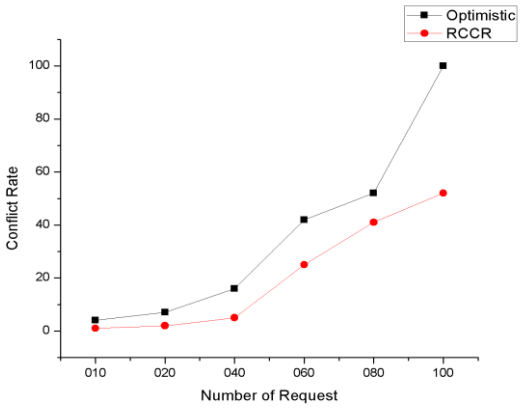

Figure 7. Comparison of conflict rate

Figure 7 shows the comparison of conflict rate of replicas. The results depict the conflict rate RCCR performed better than Optimistic approach, as the number of write/update request increases over a period of time. Hybrid approach in RCCR ensures that the performance of the system is maintained by taking advantage of both Pessimistic and Optimistic approach. Pessimistic approach is not considered as there does not exist any conflicts.

5. Conclusion

In this strategy a hybrid approach namely, RCCR is proposed and implemented for maintaining consistency and resolving conflicts in multi-master grid environment, which take advantage of both optimistic and pessimistic approaches. The pessimistic approach assures the quality of service and the optimistic approach is responsible for enhancing performance of the system. The contribution of this work, aims to resolve conflicts through forward conflict resolution, which means that conflicting updates are merged rather than rolled back and re-executed. The forward resolution generates a new consistent state which is more suitable for dynamic environments like data grids. The RCCR is compared with well-known approaches such as Optimistic, Pessimistic and ORCS. Various parameters has been considered such as percentage of accessing consistent data, amount of update transactions, average execution time, access cost, retired request rate and conflict rate. The results show that proposed RCCR approach assures consistency of replica while maintaining efficiency of the system.

Conflict of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

Acknowledgements
The authors would like to express their sincere thanks to the editor and anonymous reviews for their time and valuable suggestions.

References

Abawajy, J.H., & Deris, M.M. (2013). Data replication approach with consistency guarantee for data grid. *IEEE Transactions on Computers*, *63*(12), 2975-2987.

Belalem, G., & Slimani, Y. (2007). Consistency management for data grid in optorsim simulator. In *2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07)* (pp. 554-560). IEEE. Seoul, South Korea.

Belalem, G., Benotmane, Z., & Benhallou, K. (2009). Self adjustable negotiation mechanism for convergence and conflict resolution of replicas in data grids. *International Journal of Cognitive Informatics and Natural Intelligence*, *3*(1), 95-110.

Belalem, G., Haddad, C., & Slimani, Y. (2008). An effective approach for consistency management of replicas in Data Grid. In *2008 IEEE International Symposium on Parallel and Distributed Processing with Applications* (pp. 11-18). IEEE. Sydney, Australia.

Chang, J.S., & Chang, R.S. (2008). An innovative replica consistency decision model with Naive Bayesian Classifier in data grids. *Journal of the Chinese Institute of Engineers*, *31*(7), 1101-1111.

Chang, R.S., & Chang, J.S. (2006). Adaptable replica consistency service for data grids. In *Third International Conference on Information Technology: New Generations (ITNG'06)* (pp. 646-651). IEEE. Las Vegas, NV, USA.

Chihoub, H.E., Ibrahim, S., Antoniu, G., & Perez, M.S. (2012). Harmony: towards automated self-adaptive consistency in cloud storage. In *2012 IEEE International Conference on Cluster Computing* (pp. 293-301). IEEE. Beijing, China.

Dullmann, D., Hoschek, W., Jaen-Martinez, J., Segal, B., Samar, A., Stockinger, H., & Stockinger, K. (2001). Models for replica synchronisation and consistency in a data grid. In *Proceedings 10th IEEE International Symposium on High Performance Distributed Computing* (pp. 67-75). IEEE. San Francisco, CA, USA.

Fetai, I., & Schuldt, H. (2012). Cost-based data consistency in a data-as-a-service cloud environment. In *2012 IEEE Fifth International Conference on Cloud Computing* (pp. 526-533). IEEE. Honolulu, HI, USA.

Grace, R.K., & Manimegalai, R. (2014). Dynamic replica placement and selection strategies in data grids-a comprehensive survey. *Journal of Parallel and Distributed Computing*, *74*(2), 2099-2108.

Guroob, A.H., & Manjaiah, D.H. (2016). Efficient replica consistency model (ERCM) for update propagation in data grid environment. In *2016 International Conference on Information Communication and Embedded Systems (ICICES)* (pp. 1-7). IEEE. Chennai, India.

Kraska, T., Hentschel, M., Alonso, G., & Kossmann, D. (2009). Consistency rationing in the cloud: pay only when it matters. *Proceedings of the Very Large Data Bases Endowment*, *2*(1), 253-264.

Mansouri, N., Dastghaibyfard, G.H., & Mansouri, E. (2013). Combination of data replication and scheduling algorithm for improving data availability in data grids. *Journal of Network and Computer Applications*, *36*(2), 711-722.

Pérez, J.M., García-Carballeira, F., Carretero, J., Calderón, A., & Fernández, J. (2010). Branch replication scheme: a new model for data replication in large scale data grids. *Future Generation Computer Systems*, *26*(1), 12-20.

Radi, M. (2014). Improved aggressive update propagation technique in cloud data storage. *International Journal of Emerging Trends & Technology in Computer Science*, *3*(2), 102-106.

Vashisht, P., Kumar, V., Kumar, R., & Sharma, A. (2019). Optimizing Replica Creation using Agents in Data Grids. In *2019 Amity International Conference on Artificial Intelligence (AICAI)* (pp. 542-547). IEEE. Dubai, United Arab Emirates.

Vashisht, P., Sharma, A., & Kumar, R. (2017). Strategies for replica consistency in data grid–a comprehensive survey. *Concurrency and Computation: Practice and Experience*, *29*(4), e3907.

Yang, C.T., Fu, C.P., & Hsu, C.H. (2010). File replication, maintenance, and consistency management services in data grids. *The Journal of Supercomputing*, *53*(3), 411-439.