Baltic J. Modern Computing, Vol. 4 (2016), No. 3, 546-560

A Dynamic Data Replication with Consistency Approach in Data Grids: Modeling and Verification

Alireza SOURI, Monire NOROUZI, Adalat SAFARKHANLOU,
Seyed Hassan ES.HAGHI SARDROUD

Department of Computer Engineering, Hadishahr Branch, Islamic Azad University,
Hadishahr, Iran

alirezasouri.research@gmail.com

Abstract. Data Grids offer distributed resources geographically for large-scale data rigorous requests that produce large data collections. Data replication is one of the most important mechanisms according to a variety of data Grid interaction between external systems. Currently, data replication is widely used to ensure the reliability in Grid environment. Also research on the consistency protocols of the data replication mechanism is a new and important challenge. In this paper, model checking a Dynamic Data Replication with Consistency (DDRC) approach has been proposed in data Grids. This paper presents a behavioral model for the proposed approach with the goals of providing correctness of the data consistency based on quorum consistency protocol and reducing propagation time in data Grids. Evaluation and simulation of the some expected specifications such as reachability and deadlock free formulas for the considered data replication approach are provided using Process Analysis Toolkit (PAT) model checker.

Keywords: Data Grids, Model Checking, Dynamic Data Replication, Consistency, Kripke Structure.

1 Introduction

Data Grids are ubiquitous to research centers, economy, management and military organizations (García-García et al., 2013). One of the important factors in data Grids architecture is the occurrence of multiple replicas in a huge data. There are four factors atomicity, consistency, isolation and durability to guarantee the data integrity of data Grids (Gray and Reuter, 1992; Ozsu, 2007). Also consistency management is an important role to maintenance highly available, reliable and performable data via replication (Brzezinski et al., 2004). By increasing the data Grids development in distributed systems, the precise examination of data consistency is attractive an inevitability. Consistency factor can confirm that each transaction will transport the data from one valid state to other valid state (Luo et al., 2014). In an active data Grid, when a sequence updates happen, the newness term of a replica is critical (García-García et al., 2013). Also some articles and researches evaluate their approaches only without considering a specific replication model using simulation and traditional experiments. On the one hand, in these experiments they cannot specify that how replication model and its consistency performance are suitable for data replication architecture exactly

(Souri and Rahmani, 2014). On the other hand, by using the simulation results the all of the state spaces of the problem is not checked and analyzed well. So, to resolving these problems formal verification as a powerful method for verifying and model checking of Data Grids is an appropriate methodology. In some popular articles, some strategies of the replication and consistency techniques in Data Grids are discussed. However, Amjad et al. (2012) presented a survey for dynamic replication strategies in data grid. But, they just considered replication protocols without consistency models. We discuss the consistency models and replication methods in each research approach. Some research analyzed the consistency models relationships (Brzezinski et al., 2004; Zhu and Wang, 2010). For example, Zhu and Wang (2010) formally defined the four client-centric consistencies, eventual consistency, appropriating the framework from the theory of database concurrency control in large-scale Data Grids . Based on their definitions, they proved relations among these consistencies. Some associations suggest how the execution of one consistency can be completed upon additional. By these definitions, they could make simple consistency verification on system implementations. Of course, they did not show any verification method in their research. There is no verification result for modeling and checking their model. In contrast, we presented a verification method for model checking our proposed model using a powerful model checker. Also they did not show a relationship between consistency models and a data replication protocol. It is very important that how consistency model is appropriate for a specific data replication. Because, in a consistency guarantee the operate-transfer and state-transfer are necessary, they considered only operate-transfer data storage system, and it is a limitation. Brzezinski et al. (2004) discussed relationships between client-centric consistency models (known as session guarantees), and data-centric consistency models. They used a consistent notation to present formal definitions of both kinds of consistency models in the context of replicated shared objects. So, they proved a relationship between causal consistency model and client-centric consistency models. Apparently, causal consistency is similar to writes-follow- reads guarantee. They were shown that in fact causal consistency requires all common session guarantees, i.e. read-your-writes, monotonic-writes, monotonic-reads and writes-follow-reads to be preserved. They did not show a relationship between consistency models and a data replication protocol. But, we specified that our data replication approach supports all of the client-centric consistency models. Dingding et al. (2013) proposed a new I/O model to achieve a good trade-off between scalability and consistency problems. Their model based on static replication and guarantee eventual consistency model. A new model based on generic broadcast was proposed by Pedone and Schiper (1999) that support causal consistency model. Also Aguilera et al. (2000) considered the problem of generic broadcast in asynchronous systems with crashes and presented a new thrifty generic broadcast based on dynamic replication approach that support causal consistency model.

According to the above the technical methods, all the researches and the related works had not described some important factors on this scope. First is that there are not any detailed papers and research that considered formal verification and behavioral modeling data replication protocols. Second defect is that in these related works they cannot specify that how consistency model is suitable for data replication architecture exactly. Also in all works and research of Data Grids are evaluated only by simulation or traditional case study, therefore, it is possible that some part of the state spaces of the problem is not analyzed and checked well. To overcome these defects, model checking approach is essential as a powerful formal verification technique to verify the systems are employed in this research (Safarkhanlou et al., 2015; Souri and Jafari Navimipour, 2014).

In this paper, we propose a practical model checking approach for a Dynamic Data Replication with Consistency mechanism (DDRC) in Data Grids. We present a behavioral model for this mechanism that separates dynamic data replication model into two behaviors: propagation behavior and consistency behavior to ensure the consistency of replicas (Hansen et al., 2003). The isolation of these behaviors permits the maintenance and verification of the dynamic data replication with consistency mechanism according to Quorum-based consistency protocol in data Grids. Also the consistency behavior is mapped on the propagation behavior to navigate the data replication approach dynamically. The mapping process will be enables between the propagation and consistency behaviors based on Binary Decision Diagram (BDD) as a formal method approach (Clarke et al., 1999). This formal method approach extracts the predictable specifications of the dynamic data replication mechanism from consistency behavior in the form of the Linear Tree Logic (LTL) formulas. We implement the proposed behavioral model by PAT model checker according to a Kripke Structure.

The rest of the paper is organized as follows. In section 2, we present a dynamic data replication with consistency mechanism in Data Grids. Then, we present a behavioral model for the proposed mechanism. We explain the separation of the dynamic data replication behavioral model into the propagation and consistency behaviors. Section 3 describes a model checking approach for the proposed behavioral models. Furthermore, the consistency properties of behavioral models are defined by using linear temporal logic and computation tree logic languages. These properties can be checked by the specification of consistency behavior which is mapped on propagation behavior. In section 4, we present a performance evaluation for proposed mechanism according to PAT model checker. This section shows the evaluation results for checking some behavioral specifications such as reachability, fairness and deadlock free automatically. Finally, conclusions and future works are provided in Section 5.

2 Dynamic Data Replication with Consistency mechanism

In this section, we present a Dynamic Data Replication with Consistency approach (DDRC) which is based on based on Uniform Total Order protocol. First, we discuss the quorum-Based consistency replication. Then, we describe the DDRC approach with added properties in proposed approach.

We briefly describe the consistency protocol according to (Powell, 1994; Schneider, 1990; Tanenbaum and Steen, 2006). A consistency protocol describes a specific implementation of a consistency model. Some common consistency exercises are: sequential consistency, weak consistency with synchronization variables, eventual consistency, and atomic transactions. There are two terminology for consistency protocols that includes: the Primary-back approach (Passive replication) and State-machine approach (Active Replication) (Wiesmann et al., 2000). In this paper we choose the quorum-based consistency protocol as an active replication for the proposed DDRC mechanism. In a quorum-based consistency protocol, a client requests and obtains replies from multiple clients before the reading or writing procedures. A client must interacts with at least one half plus one servers to performs a read or update operation in data Grids.

Definition 1: a Quorum-based consistency protocol is a tuple Q= (N, Nr, Nw, n, q, S) that is describes as follow (Tanenbaum and Steen, 2006; Wiesmann et al., 2000):

- N is a set of replicated servers where |N| = n.
- Nr as the quorum server reader
- Nw as the quorum server writer.
- q is the read/write quorum value as input.
- The N < Nr + Nw is described that prevents from read-write conflicts.
- The $\frac{N}{2}$ < Nw is described that prevents from write-write conflicts.
- S= 2^N, that means N_1, $N_2 \subseteq 2^N$ is a pair of non-empty subsets of the replicated servers, for N_1, $N_2 \in N$ where $\forall$ Nr $\in N_1$ and Nw $\in N_2$, Nr $\cap$ Nw $\neq \emptyset$.

Figure 1 represents the dynamic replication process in the DDRC approach. A client sends the request to the replicas with a multicast ordinary procedure. A replica manager coordinates all of the replicas according to Uniform Total Order (UTO) broadcast protocol using an update version factor (Berthou and Quéma, 2013; Guerraoui et al., 2006). That means, when a process p_i and p_j both deliver messages m_1 and m_0, then process p_i delivers message m_1 before message m_0, if and only if process p_j delivers message m_1 before message m_0. In other words, all processes must deliver all messages at the same order. Then, the execution process is performed to update replicas in data Grids. The replica manager navigates the agreement coordination for the other replicas. Each replica sends back the reply message to the replica manager by a new version factor. Finally, all of the reply messages send back to the client. This reply specifies that all of the replicas update itself according to the request of client.

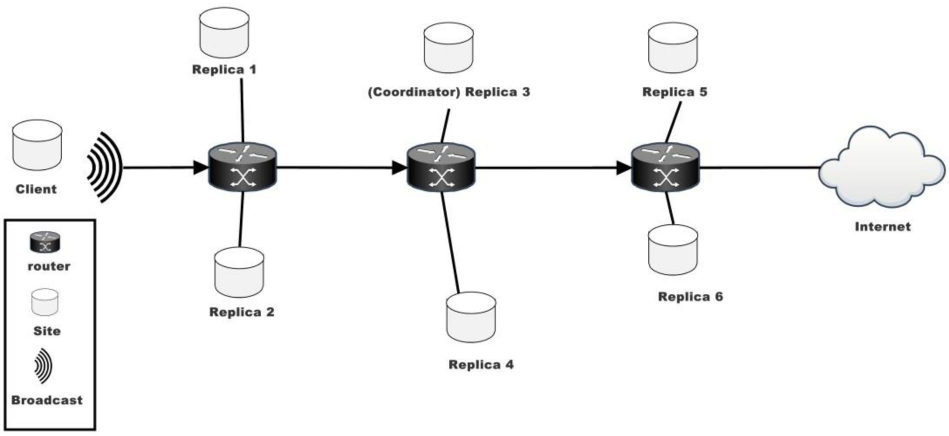

Fig. 1. Data replication framework in Data Grids.

Now we describe a behavioral model approach for the DDRC mechanism that contains: Propagation behavior and Consistency behavior according to the some researches platforms (Armendáriz-Iñigo et al., 2011; Chen et al., 2014; Zhou et al., 2004). First, we explain a formal description of offered behavioral model to separate the DDRC model into Propagation behavior and Consistency behavior.

Definition 1: The Propagation behavior of the *DDRC* is a 6-tuple $Pr_B = (A_{Pr}, a_{Pr}, U_{Pr}, V_{Pr})$ where:

- A_{Pr} is a finite set of Propagation behavior states.
- $a_{Pr} \in A_{Pr}$ is the initial state.
- U_{Pr} is a finite set of updates on a state a_{Pr} that illustrated by $U_i = (a_{Pr}, a'_{Pr}, u_i)$, $i > 0$.
- V_{Pr} is a finite set of versions for an update process u_i that is executed on state a_{Pr} and change old version v_{old} to new version v_{new} as follow: $a_{Pr}(v_{old}) \xrightarrow{ui} a_{Pr}(v_{new})$.

Definition 2: The Consistency behavior of a *DDRC* is a 4-tuple $Co_B = (A_{Co}, a_{Co}, E_{Co}, PA_{Co})$ where:

- A_{Co} is a finite set of Consistency behavior states.
- $a_{Co} \in A_{Lo}$ is the initial state.
- E_{Co} is set of *event* consistency.
- PA_{Co} is a finite set of the Consistency behavior approaches that illustrated by $a_{Co} \xrightarrow{E} a'_{Co}: pa_{Co}$ where $a, a' \in A_{Co}$, $E \in E_{Co}$ and $pa_{Co} \in PA_{Co}$. Each approach shows interaction between two states a, a′ with event *E*.

Now, we define the Kripke structures of the DDRC behavioral model.

Definition 3: A Kripke structure is a finite state machine as a 4-tuple $KS = (K, k, F, R)$, where:

- K is a finite set of states
- k is a set of initial states.
- $F \subseteq K \times K$ is a transition relation for $\forall\ k \in K, \exists\ k' \in K: (k, k') \in F$.
- R: 2^{AP} is a labeling function by *true* or *false* condition. *AP* is a nonempty set of atomic propositions. The R illustrates to each state $k \in K$ that set $R(k)$ of all atomic propositions that are valid in k.

To map the DDRC behaviors on a Kripke structure, we can use a mapping technique that illustrated in research (Souri and Jafari Navimipour, 2014).

Figure 2 clarifies the behavioral model of the DDRC approach. First, the client sends a request to propagate the version v_{new} of a process to other replicas according to the multicast mechanism. Then, a replica as a coordinator navigates the coordination of processes according to uniform total order protocol. Now, the quorum-based consistency protocol checks the consistency conditions for a data item by version v. At first, the q, Nr and Nw factors are specified. Then, this protocol uses voting mechanism for checking two read-write and write-write conflicts. After finishing the validation of two conflicts checking, the consistency completion is confirmed. If all replicas declare their agreement to the coordinator, all of the replicas update the data item version and change a new version number. That means that they change old version v_{old} to new version v_{new} and reply a response to the coordinator. The coordinator sends the response request to the client. According to definition 3, we design the behavioral model of the DDRC as a Kripke structure with supporting the quorum-based consistency protocol.

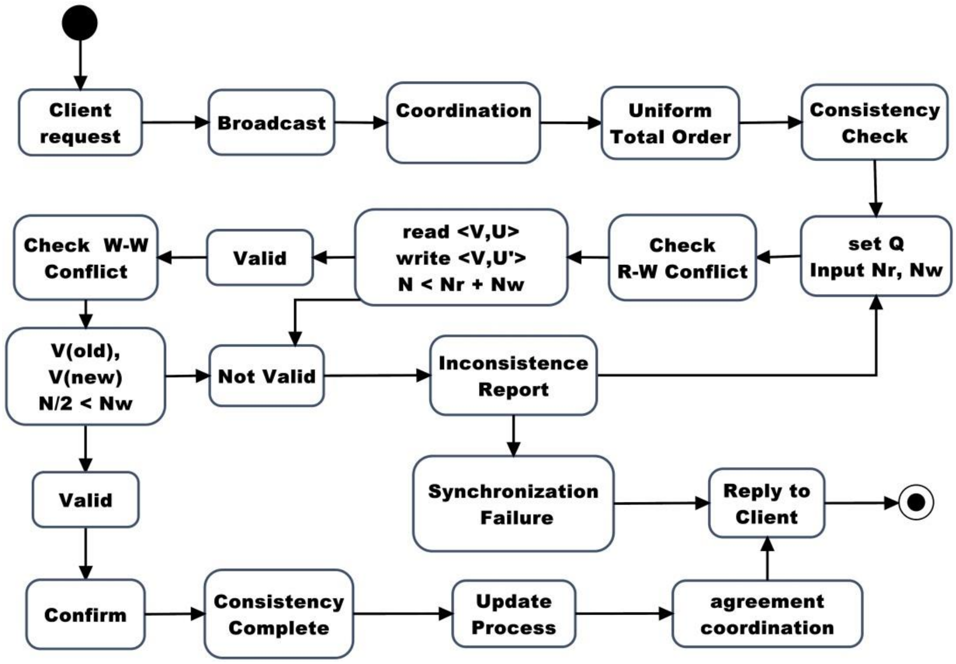

Fig. 2. Behavioral model of Dynamic Data Replication with Consistency approach (DDRC).

3 Symbolic model checking approach

In this section, we define Linear Temporal Logic (LTL) properties as a temporal logic language briefly. Then the proposed Kripke Structure is modeled. The verification of the proposed model is based on the state-of-art technologies such as symbolic model checking. We used PAT[1] model checker which is a powerful toolkit for modeling the DDRC behaviors.

Linear Temporal Logic is a formal specification language for verifying and describing the expected properties of the systems that is generally used in model checking tools. The formulas of the propositional linear temporal logic, defined as follow (Clarke et al., 1999; Jafari Navimipour et al., 2015; Souri and Norouzi, 2015):

- ***Expected Properties*** (***EP***): if α, β ∈ ***EP*** then α, β ∈ LTL (***EP***).
- ***True***: *T* ∈ LTL (***EP***).
- ***Next*** operator: If α ∈ LTL (***EP***), then Xα ∈ LTL(***EP***).
- ***General*** operator: If α ∈ LTL (***EP***), then Gα ∈ LTL(***EP***).
- ***Future*** operator: If α ∈ LTL (***EP***), then Fα ∈ LTL(***EP***).
- ***Until*** operator: If α, β ∈ LTL (***EP***), then α U β ∈ LTL(***EP***).
- ***Boolean connectives***: If α, β ∈ LTL (***EP***), then the formulas (! α) and (α | β) and (α & β) ∈LTL (***EP***).

Figure 3 illustrates the behavioral model of DDRC Kripke structure in PAT

[1] http://patroot.com/

environment. Also an example of the DDRC Kripke model simulation is shown in figure 4 that has generated by the simulator engine of the PAT model checker.

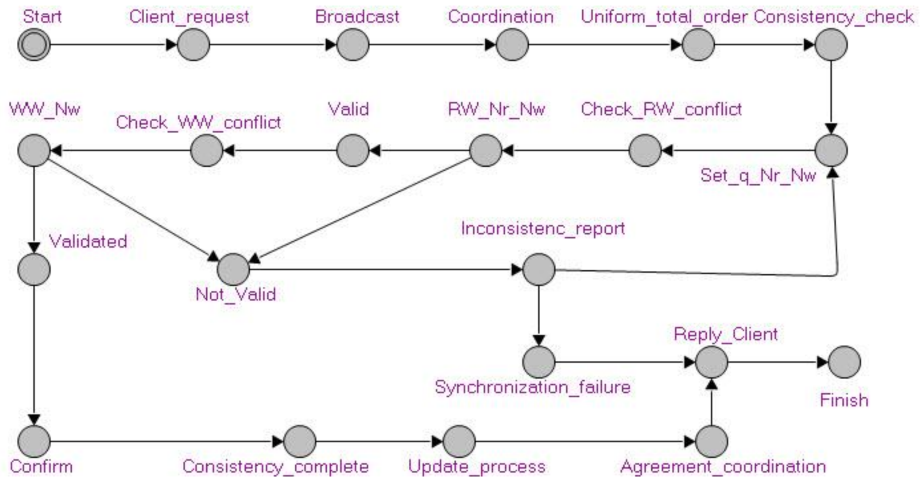

Fig. 3. DDRC Kripke structure in PAT environment.

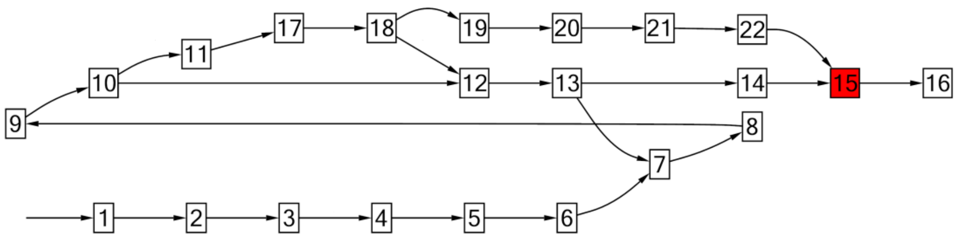

Fig. 4. The example of the the DDRC Kripke model simulation in PAT model checker.

Now, we define some LTL specifications for the DDRC Kripke model according to above rules. We Let → as the consistency association:

- L1 **G** (broadcast -> **F** (uniform_total_order)) **U** (consistency_check);

✓ Globally, when the broadcast protocol is happened, finally the uniform total order protocol is occurred at least until the consistency checking procedure is started at the current position. Figure 5 shows the L1 specification conversion to Buchi Automata for satisfaction procedure.

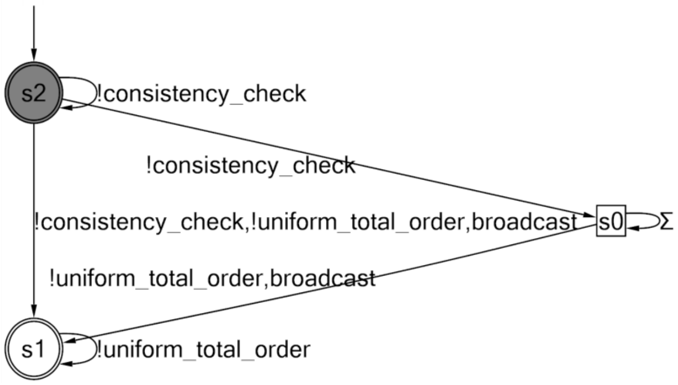

Fig. 5. Satisfaction of L1 specification in Buchi Automata.

- L2 **G (**check_WW_conflict **->** **F** (valid ||not_Valid));

✓ Globally, when the quorum-based protocol is checked the write-write conflict, finally the valid response or invalid response is occurred. Figure 6 displays the L2 specification conversion to Buchi Automata for satisfaction procedure.

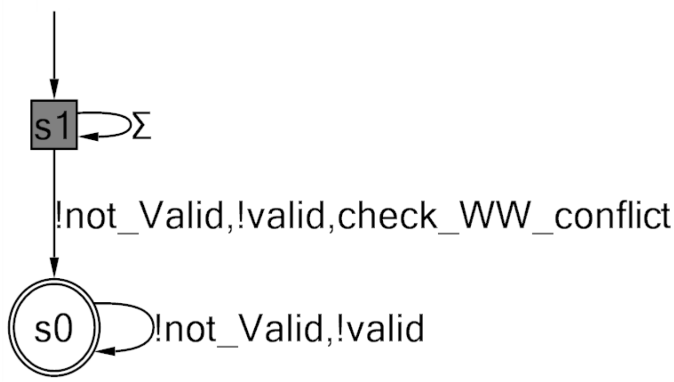

Fig. 6. Satisfaction of L2 specification in Buchi Automata.

- L3 **F**(set_q_Nr_Nw **->** **F** (check_RW_conflict **&** check_WW_conflict) **U** (confirm));

✓ Eventually, there is a state from specifying q, Nr and Nw factors to check read-write and write-write conflicts until the consistency check procedure is confirmed. Figure 7 displays the L3 specification conversion to Buchi Automata for satisfaction procedure.

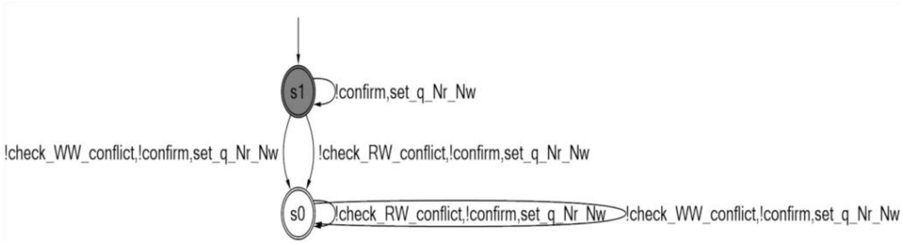

Fig. 7. Satisfaction of L3 specification in Buchi Automata.

- L4 **G**(not_Valid) **->** **X** (inconsistence_report **&** synchronization_failure);
- ✓ Globally, when invalid response is answered the next state must be an inconsistent report to the client and synchronization failure. Figure 8 displays the L4 specification conversion to Buchi Automata for satisfaction procedure.

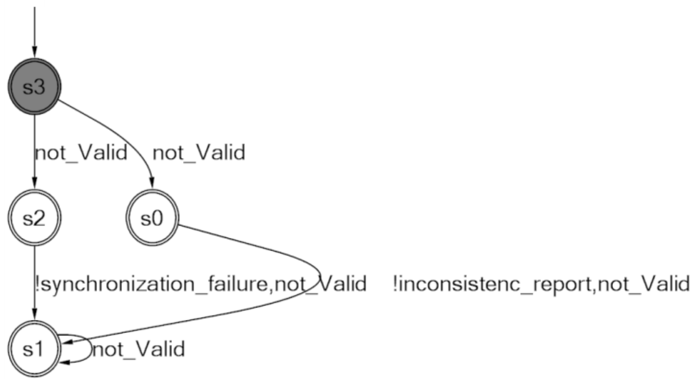

Fig. 8. Satisfaction of L4 specification in Buchi Automata.

- L5 **G**(reply_Client) **->** **X** (finish));
- ✓ Globally, when the coordinator replies the agreement coordination to the client, the next state is ending the data replication protocol. Figure 9 displays the L5 specification conversion to Buchi Automata for satisfaction procedure.

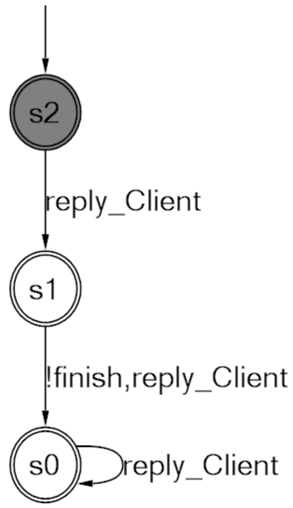

Fig. 9. Satisfaction of L5 specification in Buchi Automata.

- L6 deadlock-free ;
- ✓ In all of the state spaces of the model, there is no any deadlock condition.
- L7 reaches finish;
- ✓ The finish state for ending the protocol is is always potentially reachable in all of the state spaces.
- L8 reaches consistency_check;
- ✓ The consistency conflict checking is always potentially reachable in all of the state spaces.
- L9 reaches agreement_coordination;
- ✓ The agreement coordination is is always potentially reachable in all of the state spaces.

4 Experimental results

The verification and simulation results of the DDRC specifications are implemented by an Intel Core i5, 2.8 GHz, 4GB RAM, Windows 7 platform by PAT model checker 3.4.1. The proposed Dynamic Data Replication with Consistency approach has three advantageous in comparison of similar works. First, unlike many papers in this scope, the proposed approach used multicast protocol and uniform total order protocol for navigating the updates propagation in data Grids. Second advantageous is using quorum-based consistency protocol for detecting the consistency factor in data replication approach. Third advantageous is decreasing propagation time in this framework. The proposed approach is evaluated according to the faithfulness of the formal models and their usefulness for model checking. Due to the size and complexity of current systems reachability and deadlock-free conditions are not always checked, therefore two requirements is necessary to verifying the proposed model:

- The presented model contains all the necessary hints for checking a required property.
- The model covers only the effective behaviors of the actual system.

Figure 10 demonstrates the verification results of the DDRC Kripke model in PAT model checker. Some LTL specifications of the DDRC model are shown in this figure. According to this figure, all of the expected specifications are satisfied by state spaces model using automated verification method. That means, the proposed model supports some critical specifications according to quorum-based consistency protocol and data propagation mechanism. Eventually, the system shows that the proposed DDRC model is reachable and deadlock-free.

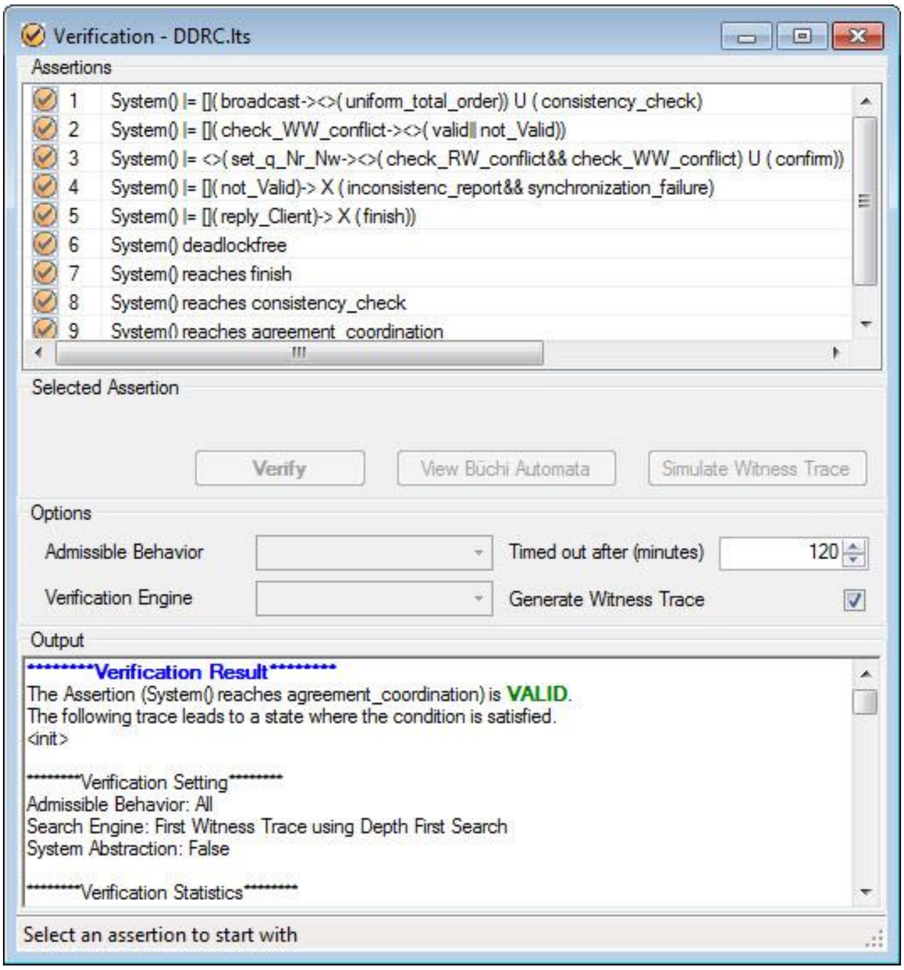

Fig. 10. The verification results of the DDRC Kripke structure in PAT model checker.

Figure 11 illustrate the model checking time for each LTL specification using PAT model checker. The maximum time belongs to the L9 specification ID that checks the deadlock-free condition for the proposed Kripke model.

Also Figure 12 demonstrates the memory consumption to verify the specifications of the Dynamic Data Replication with Consistency approach.

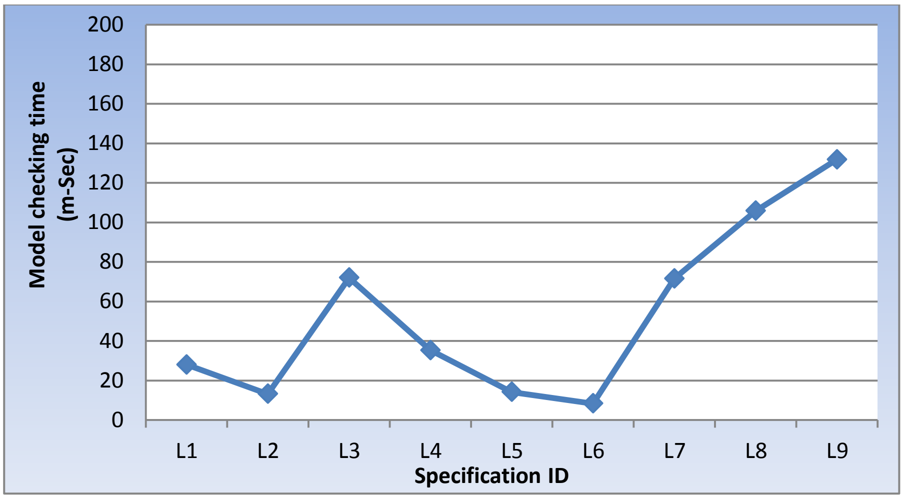

Fig. 11. The model checking time for each specification using PAT model checker.

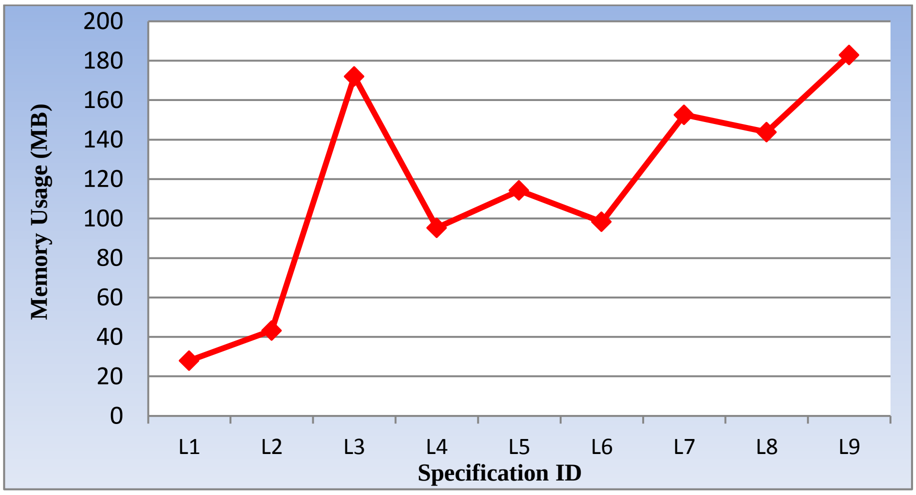

Fig. 12. The memory usage for each specification in the DDRC Kripke model using PAT model checker.

The figure 13 depicts the simulation results for 3, 5, 10, 15, 20, 30 and 50 replicas in the proposed DDRC Kripke model. According to this figure, the propagation time of the DDRC model is lower than the propagation time of the data replication (DR) protocol without multicast and uniform total order protocols in PAT simulation environment.

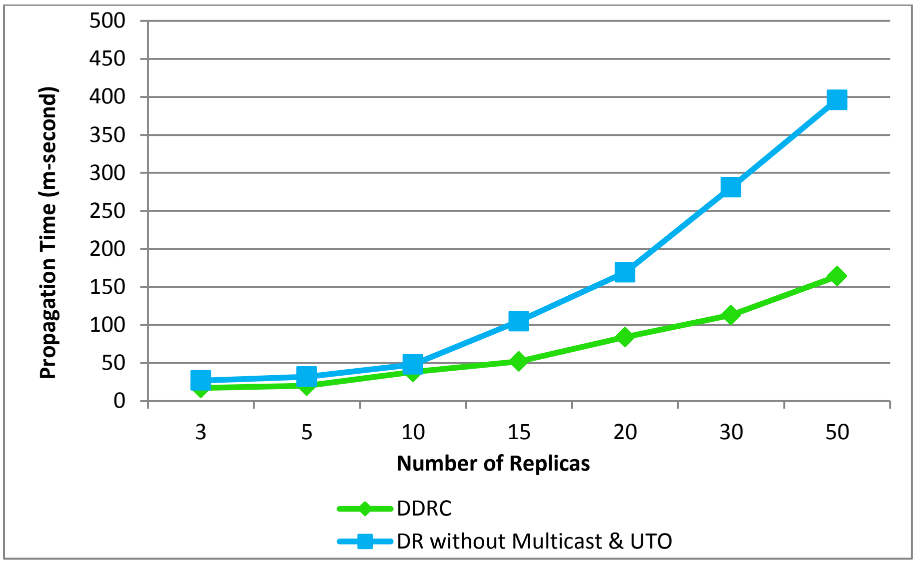

Fig. 13. The propagation time comparison for the DDRC the DR approaches.

5 Conclusion and future work

This paper presented a Dynamic Data Replication with Consistency approach in Data Grids. The proposed approach is based on quorum-based consistency protocol. We take out the probable specification of Dynamic Data Replication with Consistency approach from consistency behavior in the form of LTL formulas. The behavioral model of DDRC had verified using some expected specifications by PAT model checker. The results of verification displayed that the Dynamic Data Replication with Consistency approach can successfully propagates update request of each client to other replicas using multicast and uniform total order protocols. The verification results illustrated the correctness of the data consistency based on quorum consistency protocol and satisfied some expected specifications such as reachability and deadlock free. Finally, a comparison of propagation time for the Dynamic Data Replication with Consistency approach and the data replication protocol without multicast and uniform total order protocols is analyzed using simulation results of PAT model checker. The comparison result shows that the propagation time of the Dynamic Data Replication with Consistency is lower than the traditional data replication protocol without multicast and uniform total order protocols. In the future work, we will try to extend and analyze the data replication approach with other consistency protocols in cloud environments.

References

Aguilera, M., Delporte-Gallet, C., Fauconnier, H., Toueg, S. (2000). Thrifty Generic Broadcast. In M. Herlihy (Ed.), *Distributed Computing* (Vol. 1914, pp. 268-282): Springer Berlin Heidelberg.

Amjad, T., Sher, M., Daud, A. (2012). A survey of dynamic replication strategies for improving data availability in data grids. *Future Generation Computer Systems, 28*(2), 337-349. doi: http://dx.doi.org/10.1016/j.future.2011.06.009

Armendáriz-Iñigo, J. E., Juárez-Rodríguez, J. R., González de Mendívil, J. R., Garitagoitia, J. R., Irún-Briz, L., Muñoz-Escoí, F. D. (2011). A formal characterization of SI-based ROWA replication protocols. *Data & Knowledge Engineering, 70*(1), 21-34. doi: http://dx.doi.org/10.1016/j.datak.2010.07.012

Berthou, G., Quéma, V. (2013). FastCast: A Throughput- and Latency-Efficient Total Order Broadcast Protocol. In D. Eyers and K. Schwan (Eds.), *Middleware 2013: ACM/IFIP/USENIX 14th International Middleware Conference, Beijing, China, December 9-13, 2013, Proceedings* (pp. 1-20). Berlin, Heidelberg: Springer Berlin Heidelberg.

Brzezinski, J., Sobaniec, C., Wawrzyniak, D. (2004, 11-13 Feb. 2004). *From session causality to causal consistency.* Paper presented at the Parallel, Distributed and Network-Based Processing, 2004. Proceedings. 12th Euromicro Conference on.

Chen, T., Bahsoon, R., Tawil, A.-R. H. (2014). Scalable service-oriented replication with flexible consistency guarantee in the cloud. *Information Sciences, 264*, 349-370. doi: http://dx.doi.org/10.1016/j.ins.2013.11.024

Clarke, E. M., Grumberg, O., Peled, D. A. (1999). *Model checking*: MIT press.

Dingding, L., Xiaofei, L., Hai, J., Bingbing, Z., Qi, Z. (2013). A New Disk I/O Model of Virtualized Cloud Environment. *Parallel and Distributed Systems, IEEE Transactions on, 24*(6), 1129-1138. doi: 10.1109/TPDS.2012.321

García-García, J., Ordonez, C., Tosic, P. (2013). Efficiently repairing and measuring replica consistency in distributed databases. *Distributed and Parallel Databases, 31*(3), 377-411. doi: 10.1007/s10619-012-7116-0

Gray, J., Reuter, A. (1992). *Transaction Processing: Concepts and Techniques*: Morgan Kaufmann Publishers Inc.

Guerraoui, R., Levy, R. R., Pochon, B., Quema, V. (2006, 25-28 June 2006). *High Throughput Total Order Broadcast for Cluster Environments.* Paper presented at the Dependable Systems and Networks, 2006. DSN 2006. International Conference on.

Hansen, H., Virtanen, H., Valmari, A. (2003). *Merging State-Based and Action-Based Verification*. Paper presented at the Proceedings of the Third International Conference on Application of Concurrency to System Design.

Jafari Navimipour, N., Habibizad Navin, A., Rahmani, A. M., Hosseinzadeh, M. (2015). Behavioral modeling and automated verification of a Cloud-based framework to share the knowledge and skills of human resources. *Computers in Industry, 68*, 65-77. doi: http://dx.doi.org/10.1016/j.compind.2014.12.007

Luo, C., Okamura, H., Dohi, T. (2014). Performance evaluation of snapshot isolation in distributed database system under failure-prone environment. *The Journal of Supercomputing*, 1-24. doi: 10.1007/s11227-014-1162-5

Ozsu, M. T. (2007). *Principles of Distributed Database Systems*: Prentice Hall Press.

Pedone, F., Schiper, A. (1999). Generic Broadcast. In P. Jayanti (Ed.), *Distributed Computing* (Vol. 1693, pp. 94-106): Springer Berlin Heidelberg.

Powell, D. (1994). *Distributed Fault Tolerance - Lessons Learned from Delta-4*. Paper presented at the Revised Papers from a Workshop on Hardware and Software Architectures for Fault Tolerance.

Safarkhanlou, A., Souri, A., Norouzi, M., Sardroud, S. E. H. (2015). Formalizing and Verification of an Antivirus Protection Service using Model Checking. *Procedia Computer Science, 57*, 1324-1331. doi: http://dx.doi.org/10.1016/j.procs.2015.07.443

Schneider, F. B. (1990). Implementing fault-tolerant services using the state machine approach: a tutorial. *ACM Comput. Surv., 22*(4), 299-319. doi: 10.1145/98163.98167

Souri, A., Jafari Navimipour, N. (2014). Behavioral modeling and formal verification of a resource discovery approach in Grid computing. *Expert Systems with Applications, 41*(8), 3831-3849. doi: http://dx.doi.org/10.1016/j.eswa.2013.11.042

Souri, A., Norouzi, M. (2015, 23-25 Sept. 2015). *A new probable decision making approach for verification of probabilistic real-time systems.* Paper presented at the Software Engineering and Service Science (ICSESS), 2015 6th IEEE International Conference on.

Souri, A., Pashzadeh, S. (2014). CONSISTENCY OF Data Replication protocols in database Systems: A review. *International Journal on Information Theory (IJIT), 3*(4), 19-32.

Souri, A., Rahmani, A. M. (2014). A survey for replica placement techniques in data grid environment. *International Journal of Modern Education and Computer Science, 6*(5), 46.

Tanenbaum, A. S., Steen, M. v. (2006). *Distributed Systems: Principles and Paradigms (2nd Edition)*: Prentice-Hall, Inc.

Wiesmann, M., Pedone, F., Schiper, A., Kemme, B., Alonso, G. (2000, 2000). *Understanding replication in databases and distributed systems.* Paper presented at the Distributed Computing Systems, 2000. Proceedings. 20th International Conference on.

Zhou, W., Wang, L., Jia, W. (2004). An analysis of update ordering in distributed replication systems. *Future Generation Computer Systems, 20*(4), 565-590. doi: http://dx.doi.org/10.1016/S0167-739X(03)00174-2

Zhu, Y., Wang, J. (2010). Client-centric consistency formalization and verification for system with large-scale distributed data storage. *Future Generation Computer Systems, 26*(8), 1180-1188. doi: http://dx.doi.org/10.1016/j.future.2010.06.006

Received December 3, 2015, revised February 27, 2016, accepted May 4, 2016