CONSISTENCY OF DATA REPLICATION PROTOCOLS IN DATABASE SYSTEMS: A REVIEW

Alireza Souri[1], Saeid Pashazadeh*[2] and Ahmad Habibizad Navin[3]

[1, 3]Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran.
[2]* Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran.

Abstract:

In this paper, a review for consistency of data replication protocols has been investigated. A brief deliberation about consistency models in data replication is shown. Also we debate on propagation techniques such as eager and lazy propagation. Differences of replication protocols from consistency view point are studied. Also the advantages and disadvantages of the replication protocols are shown. We advent into essential technical details and positive comparisons, in order to determine their respective contributions as well as restrictions are made. Finally, some literature research strategies in replication and consistency techniques are reviewed.

Keywords:

Database system, consistency, data replication, update propagation.

1. Introduction

Consistency of Replication models is essential to abstract away execution particulars, and to classify the functionality of a given system. Also a consistency model is a method for come to a joint considerate of each other's rights and responsibilities.

Database system attracts lots of consideration. A large-scale database storage system [1,2] is among the fundamental conveniences in the cloud, unstructured peer-to-peer (P2P) networks [3], grid environment [4] or in similar systems. The system with large-scale database system typically assigns computing replicas near their input data[5]. A good data management develops very important conditions in such a scenario. Data in a distributed database system [6] is replicated for increasing reliability, availability and performance. There are two mechanisms for locations of data replicas such as static and dynamic replicated system[7,8], which regulates replica locations based on session information of requests [9].

DOI : 10.5121/ijit.2014.3402

On the one hand, in these consistency models, its performance suitability for data replication architecture are not specified exactly. On the other hand, a consistency model dos not guarantees the high performance and high scalability for a data replication mechanism [10].

Consistency, accessibility, scalability, security, fault tolerant and performance [11] are areas for system implementation[12]. High accessibility and performance are basics for such a system with large-scale distributed database system. We have to make a tradeoff between consistency and replication. There are dissimilar levels of weak and strong consistency. A distributed database system may deliver levels of consistency weaker than one-copy-serializability but stronger than eventual consistency. Also there are levels of consistency such as data-centric and client-centric models for a data replication mechanism. So the recognizing usage of consistency models in each data replication mechanism is necessary. In Section 2, we introduce consistency models in client view and server view, adapted from the theory of database concurrency control. Then, we depict on consistency protocols in Section 3. With these discussions, we can represent a comparison among eventual consistency and client-centric consistency models. Properties of their implementations can also be deduced accordingly. In section 4, we discuss replication models and propagation techniques. We show all of the replication protocols according to the update propagation and replication mechanisms in Section 5. Section 6 shows a classified review for replication and consistency techniques in some research strategies. Section 7 is the conclusion and describes future work finally.

2. Consistency models

In this section, a series of consistency models are considered. We discuss about differences of consistency models. Variant methods to categorizing the consistency models can be originated from [13] and [14].One of the important properties of a system design is consistency model. This property can typically offered in relations of a state that can be true or false for different implementations. Consistency models are referred to as the contracts between process and data for ensuring correctness of the system. Consistency models are presented through a number of consistency criteria to be satisfied by assessments of operations [15].For standard consistency conditions of the ACID properties [16], there exists some methods for consistency guarantee. In ACID consistency method, database is in a consistent state when a transaction is finished. In the client level there are four component:

- *DS is a storage system.*
- *PA* is the process operation for each read or write by DS.
- *PB* is sovereign of process *PA* that performs each read and write operation from the DS.
- *PC* is sovereign of process *PA* that performs each read and write operation from the DS.
-

In the client level consistency, it is important that how and when an observer is occurred. The *PA, PB* and *PC* processes see updates with a data item in the storage system. There are two consistency types such as Data-Centric consistency and Client-Centric consistency[17].
In Data-Centric consistency there are:

- *Strict consistency*. All of the A, B and C send back the result of update value when the update procedure is completed.

- *Sequential consistency*. The level of sequential consistency is lower than strict consistency. Each read and write operation is performed by all replicas on their data item *x* sequentially. Also each discrete procedure operations execute the identified order.
- *Causal consistency.* This consistency is weaker than strict and sequential consistency[18]. If transaction T_1 is influenced or caused on an earlier transaction T_2, each replica should be first see T_2, and then see T_1.
- *FIFO consistency.* FIFO consistency is relaxed to implement because it is being guaranteed two or more writes from a single source must arrive in the order issued. Basically, this means that with FIFO consistency, all writes generated by different processes are concurrent.

In Client-Centric consistency models there are:

- *Eventual consistency*. This model guarantees that if a updates are complete to the item eventually [19], then all accesses on this data item send back the previous updated value [20].
- *Monotonic Reads.* In this model if an operation reads the data item x, always each following read operation on data item x send back same value *x* or a more recent value.
- *Monotonic Writes.* In this model if an operation writes on the data item x, always each following write operation on data item x comes after related write operation on the data item *x.*
- *Read-your-Write.* The result of a write operation on the data item x always will be realized by a following read operation on x by the same value.
- *Write follow read.* In this model the effect of a write operation on a data item x following a previous read operation on data item *x* by the same value that is guaranteed to take place on the same or a more recent value of x that was read.

3. Consistency Protocols

In this section, we describe the consistency protocols according to[21]. A consistency protocol explains as an implementation of a specific consistency model. We track the group of our conversation on consistency models[22].

3.1 Primary Replica Based Protocol

In this protocol, All write operations to a data item x is attended by one specific replica that called primary replica. This primary replica [23] is accountable for updating other replicas, the client just cooperates by this primary replica[16].

Two requirements should be happen for this generous of protocol [24]:

- All read and write operations for updating a data item *x* should spread and be executed all replicas at some time.
- These operations should be executed in the same order.

3.2 Replicated Write Protocol:

In this protocol, each write operations are sent to each replica to update procedure. There are two types for replicated write protocols.

3.2.1. Active Replication:

In active replication, each replica contains a concomitant procedure that transports out the update operations. Unlike other protocols, update operations are normally propagated through the write operation. This propagation causes the operation is sent to each replica. Also there is required a total order for all write operations that each replica execute the same order of write commands [25].

3.2.2. Quorum Based:

This protocol specifies that the clients obtain the authorization of several servers before any reading or writing a replicated data item *x* [26]. For example, the write operations only want to be executed on fragment of all replicas before return to the client. It use elections to avoid write-read conflict and write-write conflict [27]:

- ***R*** is the number of replicas of each data item.
- ***Rr*** is number of replicas that a client should contacts by them for reading a value.
- ***Rw*** is number of replicas that a client should contacts by them for writing a value.
- For preventing the Write-Write and Write-Read conflicts, $Rr + Rw > R$ and $Rw + Rw > R$ should be satisfied.

4. Update propagation strategies

Update propagation can be measured in two methods [28]

- The update operations are applied to all replicas as part of the unique contract.
- Each replica is updated by the originating transaction. Update operations send to other replicas asynchronously as a discrete transaction for each node [29].

There are two update propagation methods: Eager techniques and lazy techniques. Normally the eager protocols are identified as read-one/write-any (ROWA) protocols. First, they have not transactional inconsistencies. Second, an update transaction can read a local copy of the data item *x* and be sure that a refresh value is read. Consequently, there is no essential to executing a remote read. Finally, the variations to replicas are completed atomically. When we use to a 2PC execution, the update speed is restricted and it cause that the response time performance of the update transaction is low. When one of the copies is inaccessible, the update transaction cannot terminate meanwhile all the copies updated essentially. Lazy protocol is used to new mechanisms for guaranteeing strong mutual consistency [30]. These mechanisms may be bright to endure some inconsistency between the replicas for better performance[31].

In a distributed database system the requests permit to access data from local and remote databases [32]. Distributed methods spread on the update procedure to the local copy where the update transaction creates, then the updates are broadcasted to the other replica. If distributed techniques are attached by eager propagation approaches, then the distributed concurrency control approaches can sufficiently report the concurrent updates problem [13]. Table 1 is shown the comparisons of update propagation and propagation techniques [33].

Table 1. Comparisons of update propagation techniques

	Consistency	Updating	Performance	Failure
Eager update	Strong consistency	Up-to-date with High response time	Not transactional inconsistency, Changes are atomic	Restricted update speed, transaction crash and Lower availability
Lazy update	Weak consistency	Out-of-date problem and Low response time	Not fault tolerant, good response time	Dirty read problem, Data inconsistency and transaction inversion
Centralized techniques	-	Up-to-date with Update without synchronization	Appropriate for few master sites	High overload and bottleneck problems
Distributed techniques	-	Up-to-date with Concurrency control methods	Highest system availability	Management problem, Copies need to be synchronized

5. Replication Protocols

[13] presented a categorization for replicas data protocols. This is important that When one of the update propagation mechanisms such as eager or lazy incomes and who should complete updates mechanism such as primary copy or update-everywhere. In eager propagation mechanism, the propagation of updates is contained by the restrictions of a transaction. The client does not receive the notification of commit message up to necessary duplicates have been updated in the system. In the Lazy mechanism, the update procedure of a local copy is committed. the update propagation be accomplished [34]. There is an expensive way for providing response time and message overhead in consistency of eager mechanism. An optimization for prevent from these problems is using Lazy replication approach. However, the update procedure is executed separately, therefore inconsistency conditions might occur. [35]. When the updates are broadcast to replicas in eager or lazy mechanism, two architectures are needed for updates such as centralized and distributed. Table 2 shows the four replication mechanisms such as eager distributed and eager centralized for eager mechanism, lazy distributed and lazy centralized for lazy mechanism [36].

Table 2. Update propagation vs. propagation techniques

	Centralized	**Distributed**
Eager	Eager Primary Copy	Eager Update Everywhere
	One Master by Restricted Transparency	
	One Master with Full Transparency	
Lazy	Lazy Primary Copy	Lazy Update Everywhere
	Single Master with Limited Transparency	

5.1 Eager Centralized Protocol

In eager centralized protocol, there is a site as a master that navigates the read and writes operations on a data item (x). This protocol guarantees strong consistency techniques for update propagation. In update procedure, all updates are applied to a logical data item (x) by using the perspective of the update transaction. This applying is committed by using the 2 Phase Commit protocol. So, when the update procedure is completed in its transaction, all copies return the similar values to the updated data items. The result of this mechanism is one-serializability-replication[37]. The categories of eager centralized include eager primary copy, single master by restricted transparency and single master by full transparency. In eager primary copy, any data item (x_i) has a master. One replica specified as the primary copy. In this case, there is no single master for controlling serializability condition. In the single master by restricted transparency, all of the updates have been sent to the specified master directly. For a read operation, a read lock occurred on data item *x* and the read operation is executed. The result of the operation is returned to the client. Also for a write operation, a write lock occurred on data item *x* and this operation is executed. The result of write operation is returned to the client. In the single master by full transparency, the replica coordination level has been performed by a router. The router sends the entire read and writes operations to the master directly. The master executes each operation and returns the result of execution to the client.

5.2 Eager Distributed Protocol

In eager distributed protocol, first the update applied to the local replica, then the update procedure is propagated to other replicas. The eager update everywhere is a type of eager distributed protocol.

5.3 Lazy Centralized Protocol

Lazy centralized protocol is like to eager centralized protocol. In this protocol, first the updates are applied to a master and then propagated to the clients. The significant alteration is that the propagation procedure does not take place via the update process. However, after the commitment of transactions, if a client executes a read operation (*x*) on its local copy, it may read a non-refresh data, then data item *x* may have been updated at the master, nevertheless the update may not have been propagated to the clients yet. The categories of lazy centralized include lazy primary copy and single master by restricted transparency. In lazy primary copy, each read and write operation sends to a master. All of the updating results have been send back to the client. In single master by restricted transparency, the update procedure is executed to the master directly. When one update has committed, the new transaction is sent to the clients.

5.4 Lazy Distributed Protocol

In Lazy distributed protocol, the update transactions can execute on each replica. Also these updates are propagated to the other replicas lazily. Lazy update everywhere is a type of lazy distributed protocol. In this type, each read and write operation are performed on the local copy and the update transactions commit locally. Comparison of replication protocols about consistency conditions is shown in Table 3.

Table 3. Comparison of replication protocols

Replication strategies	Advantages	Disadvantages
Eager Centralized	The coordination do not needs for Update transactions, there is no inconsistencies	Extensive response time, Local copies are can only be Read, Only useful with few updates
Lazy Centralized	The coordination do not needs for Update transactions, there is diminutive response times	Inconsistencies, Local copies are not refresh
Eager Distributed	No inconsistencies	Updates need to be coordinated, Long response times
Lazy Distributed	Shortest response times, No centralized coordination	Inconsistencies, Updates can be lost

6. Comparison of Consistency and replication classification

In this section, some popular and applicable research strategies of replication and consistency techniques in database systems are discussed. However, Amjad, et al. [38] 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.

In a distributed system for providing and handling extremely available service via no single point of failure, Lakshman and Malik [39] proposed a quorum-based protocol. This system replicates data by using in replicated-write group. Also they present three quorum values for guarantee eventual consistency model. [40] presented the design of a highly available key-value storage system (Dynamo) which is supports eventual consistency model via quorum-based protocol hat it allows for better availability in presence of failure. A new dimension of different cloud providers (MetaStorage) based on quorum strategy was presented by [41]. They proposed a new consistency model based on static approach. Dingding, et al. [42] proposed a new I/O model to achieve a good tread-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 [43] that support causal consistency model. Also Aguilera, et al. [44] 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. Sousa, et al. [45] proposed a technique for native clocks and the constancy of network suspensions to decrease the faults in the ordering of cautious deliveries in wide area networks. They present their model based on static replication approach that guarantee strong consistency model. An algorithm that handles replication efficiently in active replication was presented by [46]. Their algorithm is based on static replication approach that focused on strong consistency model. They could not manage their algorithm when the rollback problem is occurred. Xinfeng [47] presented a middleware for using a timestamp-based protocol to maintain the replica consistency. Their algorithm is based on static replication approach that focused on strong consistency model for improving scalability problem.A static distributed data replication mechanism of cloud in Google file system was proposed through[48]. They considered some features when creating conclusions on replicas of data: 1-insertion the new replicas on mass servers by choosing lower-average disk space consumption, 2-limiting the sum of replica establishments on each mass server and 3- spreading replicas of a mass crossways stand. Their algorithm is based on static replication approach that support eventual consistency model.

Wenhao, et al. [49] proposed a novel cost-effective dynamic data replication strategy named CIR in cloud data centers. They applied an incremental replication approach to minimizing the number of replicas while meeting the reliability condition in order to facilitate the cost-effective data replication management goal. Their approach could reduce the data storage cost substantially, especially when the data are only stored for a short duration or have a lower reliability requirement. Also their strategy is based on dynamic replication approach that support causal consistency model.

Qingsong, et al. [50] proposed a dynamic distributed cloud data replication algorithm CDRM to capture the relationship between availability and replica number. They focused on dynamic replication approach that supports a causal consistency model. Ranganathan and Foster [51] presented six different replication strategies for three different access patterns: Best Client, Cascading Replication, No Replication or Caching, Plain Caching, Caching plus Cascading Replication, and Fast Spread. They guarantee the reduction of access latency and bandwidth consumption based on dynamic replication approach. A centralized data replication algorithm (CDRA) for Grid sites was presented by [52]. Their algorithm reduced the total file access time with the consideration of limited storage space of Grid sites. Choi and Youn [53] proposed a dynamic hybrid protocol (DHP) which effectively combines the grid and tree structure. This protocol can detect read-write conflict and write-write collision for consistency maintaining. Their protocol is based on dynamic replication approach that supports an eventual consistency model.

An evolutionary algorithm to find the optimal replication strategy was proposed by [54]. They optimized reliability, latency and storage of the system. Because they considered static replication approach, their protocol did not take total data center energy cost as the primary optimization target. Lloret, et al. [55] presented a protocol for exchanging information, data, services, computing and storage resources between all interconnected clouds. Their protocol is based on static replication approach that guarantees an eventual consistency model.Table 4 summarizes the discussed research strategies and introduces their advantages and disadvantages.

Table 4. A collection of research strategies on consistency and replication

Article	Main idea	Consistency method	Replication scheme	Advantages	Disadvantages
[39]	Presenting a distributed system for handling and providing highly available service by no single point of failure.	Eventual	Dynamic	Providing good scalability and supports dynamic control over data layout and format.	The main consistency model is restricted to eventual consistency.
[40]	presenting the design of a highly available key-value storage system (Dynamo)	Eventual	Dynamic	providing a novel interface for developers to using the large e-commerce operations	The response time for replicas not considered
[41]	Presenting a new dimension of different cloud providers (MetaStorage) based on quorum strategy	Eventual	Static	MetaStorage has a highly available and scalable distributed hash table for control consistency-latency	The strategy can only guarantee single consistency model
[42]	proposing a new I/O model to reach a good tread-off between Scalability and Consistency	Eventual	Static	this new model has many advantages over the conventional asynchronous-synchronous model	Limiting consistency maintenance to eventual consistency model
[43]	Ordering the delivery of messages only if needed, based on the semantics of the messages.	Causal	Static	Showing better scalability via optimizing the atomic broadcast protocol with relaxed causal consistency	Static consistency model, semantic of data is difficult to identify without knowing the environment.
[44]	considering the problem of generic broadcast in asynchronous systems with crashes	Causal	Dynamic	By defining a parsimonious approach for the set of messages in generic broad- cast ensures can have optimal scalability	The availability has not considered and the number of replicas are not shown
[45]	Proposing a technique for local clocks and the stability of network delays to reduce the mistakes in the ordering of tentative deliveries in wide area networks	Strong	Static	Improves scalability based on the assumption that data conflict is rarely occurring.	Fixed consistency model, expensive cost process
[46]	Presenting an algorithm that handles replication efficiently in active replication	Strong	Static	Improves scalability	Fixed consistency model
[47]	Presenting a middleware for using a timestamp-based protocol to maintain the replica consistency	Strong	Static	Improves scalability	Fixed consistency model, expensive roll back process.

[48]	propose a static distributed data replication mechanism in cloud	Eventual	Static	insertion the new replicas on mass servers by choosing lower-average disk space consumption,	Fixed replica number is used for all files which may not be the best solution for data.
[49]	proposing a novel cost-effective dynamic data replication strategy named CIR in cloud data centers	Causal	Dynamic	applies an incremental replication approach to minimize the number of replicas and it can reduce the data storage cost substantially	their approach is only based on the reliability parameters and pricing model of Amazon S3 which makes it is not suitable for Google cluster
[50]	Proposing a dynamic distributed cloud data replication algorithm CDRM to capture the relationship between availability and replica number.	Causal	Dynamic	maintains the minimum replica number for a given availability requirement, Improves scalability	The scalability approach is not proposed
[51]	presenting six different replication strategies for three different access patterns	Eventual	Dynamic	Reduction in access latency and bandwidth consumption.	The fixed consistency model and limited number of replica
[52]	presenting a centralized data replication algorithm (CDRA) and designing a distributed caching algorithm wherein Grid sites	Eventual	Dynamic	reduce the total file access time with the consideration of limited storage space of Grid sites	The limitation of the algorithm is that it considers only the access cost.
[53]	proposing a dynamic hybrid protocol (DHP) which effectively combines the grid and tree structure	Eventual	Dynamic	The protocol can detect read/write conflict and write/write collision for consistency maintaining.	The grid and tree structure can only support read-one/write-all mechanism but hybrid protocol can have read-all/write-all
[54]	Presenting an evolutionary algorithm to find the optimal replication strategy	Eventual	Static	optimize latency, storage and reliability of the system	This algorithm cannot take total data center energy cost as the primary optimization target. Also it doesn't take into account the load balancing of the replicas.
[55]	Presenting a protocol for exchanging information, data, services, computing and storage resources between all interconnected clouds	Eventual	Static	highly scalable and load balancing approaches	The resource cost is not considered in replicas

Table 5 displays a summarized form of structures of all research strategies studied in above. These structures include availability, scalability, reliability, response time, bandwidth, load balancing, number of replicas and storage cost.

Table 5. The popular factors of replication and consistency techniques

Article	Availability	Scalability	Reliability	Response time	Bandwidth consumption	Load balancing	Optimal number of replicas	Storage cost
[39]	✗	✓	✓	✗	✗	✗	✓	✗
[40]	✓	✓	✓	✗	✗	✗	✗	✗
[41]	✓	✓	✗	✗	✗	✗	✗	✓
[42]	✗	✓	✗	✗	✗	✗	✗	✓
[43]	✓	✗	✓	✗	✗	✗	✗	✗
[44]	✗	✓	✗	✗	✗	✗	✗	✗
[45]	✗	✓	✓	✓	✗	✗	✗	✗
[46]	✓	✗	✓	✗	✗	✗	✗	✗
[47]	✗	✗	✓	✓	✗	✗	✗	✓
[48]	✓	✗	✓	✓	✓	✓	✗	✗
[49]	✓	✗	✓	✗	✗	✗	✓	✗
[50]	✓	✗	✗	✗	✓	✓	✓	✓
[51]	✓	✗	✓	✓	✓	✓	✗	✗
[52]	✗	✗	✓	✓	✗	✗	✗	✗
[53]	✓	✗	✗	✓	✗	✓	✓	✗
[54]	✗	✗	✓	✓	✓	✗	✗	✓
[55]	✓	✓	✗	✗	✗	✓	✗	✗

7. Conclusion

This paper presents a review for data replication protocols in the database systems. Also it discusses consistency models of replication mechanisms in different update propagations. By comparing propagation approaches we can use to type of consistency methods for implementing various data replication mechanisms By notice to comparison of replication protocols, a consistent replication protocol have important issue in managing and implementing database systems. In future work, we discuss efficient factors of consistency protocols in distributed databases that extended in distributed database systems.

References

[1] S. Çokpınar and T. İ. Gündem, "Positive and negative association rule mining on XML data streams in database as a service concept," Expert Systems with Applications, vol. 39, pp. 7503-7511, 6/15/ 2012.

[2] X. Wang, X. Zhou, and S. Wang, "Summarizing Large-Scale Database Schema Using Community Detection," Journal of Computer Science and Technology, vol. 27, pp. 515-526, 2012/01/01 2012.

[3] G. Gao, R. Li, K. Wen, and X. Gu, "Proactive replication for rare objects in unstructured peer-to-peer networks," Journal of Network and Computer Applications, vol. 35, pp. 85-96, 1// 2012.

[4] N. Saadat and A. M. Rahmani, "PDDRA: A new pre-fetching based dynamic data replication algorithm in data grids," Future Generation Computer Systems, vol. 28, pp. 666-681, 4// 2012.
[5] A.-H. Wu, Z.-J. Tan, and W. Wang, "Annotation Based Query Answer over Inconsistent Database," Journal of Computer Science and Technology, vol. 25, pp. 469-481, 2010/05/01 2010.
[6] F.-C. Tseng, "Mining frequent itemsets in large databases: The hierarchical partitioning approach," Expert Systems with Applications, vol. 40, pp. 1654-1661, 4// 2013.
[7] N. Mansouri, G. H. Dastghaibyfard, and E. Mansouri, "Combination of data replication and scheduling algorithm for improving data availability in Data Grids," Journal of Network and Computer Applications, vol. 36, pp. 711-722, 3// 2013.
[8] N. Mansouri and G. H. Dastghaibyfard, "A dynamic replica management strategy in data grid," Journal of Network and Computer Applications, vol. 35, pp. 1297-1303, 7// 2012.
[9] M. Tang, B.-S. Lee, X. Tang, and C.-K. Yeo, "The impact of data replication on job scheduling performance in the Data Grid," Future Generation Computer Systems, vol. 22, pp. 254-268, 2// 2006.
[10] X. Liao, H. Jin, and L. Yu, "A novel data replication mechanism in P2P VoD system," Future Generation Computer Systems, vol. 28, pp. 930-939, 6// 2012.
[11] J. E. Westfall, C. M. Kim, and A. Y. Ma, "Locking the virtual filing cabinet: A researcher's guide to Internet data security," International Journal of Information Management, vol. 32, pp. 419-430, 10// 2012.
[12] R. Garcia, R. Rodrigues, N. Pregui, and #231, "Efficient middleware for byzantine fault tolerant database replication," presented at the Proceedings of the sixth conference on Computer systems, Salzburg, Austria, 2011.
[13] J. Gray, P. Helland, P. O'Neil, and D. Shasha, "The dangers of replication and a solution," SIGMOD Rec., vol. 25, pp. 173-182, 1996.
[14] M. Wiesmann, F. Pedone, A. Schiper, B. Kemme, and G. Alonso, "Understanding replication in databases and distributed systems," in Distributed Computing Systems, 2000. Proceedings. 20th International Conference on, 2000, pp. 464-474.
[15] W. He and I.-R. Chen, "A proxy-based integrated cache consistency and mobility management scheme for client–server applications in Mobile IP systems," Journal of Parallel and Distributed Computing, vol. 69, pp. 559-572, 6// 2009.
[16] A. S. Tanenbaum and R. V. Renesse, "Distributed operating systems," ACM Comput. Surv., vol. 17, pp. 419-470, 1985.
[17] D. B. Terry, A. J. Demers, K. Petersen, M. Spreitzer, M. Theimer, and B. W. Welch, "Session Guarantees for Weakly Consistent Replicated Data," presented at the Proceedings of the Third International Conference on Parallel and Distributed Information Systems, 1994.
[18] P. W. Hutto and M. Ahamad, "Slow memory: weakening consistency to enhance concurrency in distributed shared memories," in Distributed Computing Systems, 1990. Proceedings., 10th International Conference on, 1990, pp. 302-309.
[19] Y. Zhu and J. Wang, "Client-centric consistency formalization and verification for system with large-scale distributed data storage," Future Generation Computer Systems, vol. 26, pp. 1180-1188, 10// 2010.
[20] T. Chen, R. Bahsoon, and A.-R. H. Tawil, "Scalable service-oriented replication with flexible consistency guarantee in the cloud," Information Sciences, 2014.
[21] D. Imbs and M. Raynal, "Virtual world consistency: A condition for STM systems (with a versatile protocol with invisible read operations)," Theoretical Computer Science, vol. 444, pp. 113-127, 7/27/ 2012.
[22] H. Yu and A. Vahdat, "Efficient Numerical Error Bounding for Replicated Network Services," presented at the Proceedings of the 26th International Conference on Very Large Data Bases, 2000.
[23] N. Budhiraja and K. Marzullo, "Tradeoffs in implementing primary-backup protocols," presented at the Proceedings of the 7th IEEE Symposium on Parallel and Distributeed Processing, 1995.
[24] N. Budhiraja, K. Marzullo, F. B. Schneider, and S. Toueg, "The primary-backup approach," in Distributed systems (2nd Ed.), ed: ACM Press/Addison-Wesley Publishing Co., 1993, pp. 199-216.

[25] L. E. T. Rodrigues, H. Fonseca, and P. Verissimo, "Totally ordered multicast in large-scale systems," in Distributed Computing Systems, 1996., Proceedings of the 16th International Conference on, 1996, pp. 503-510.
[26] D. K. Gifford, "Weighted voting for replicated data," presented at the Proceedings of the seventh ACM symposium on Operating systems principles, Pacific Grove, California, USA, 1979.
[27] P. Jalote, Fault tolerance in distributed systems: Prentice-Hall, Inc., 1994.
[28] H. Garcia-Molina, "Performance of update algorithms for replicated data in a distributed database," Stanford University, 1979.
[29] W. Zhou, L. Wang, and W. Jia, "An analysis of update ordering in distributed replication systems," Future Gener. Comput. Syst., vol. 20, pp. 565-590, 2004.
[30] E. A. Boiten and J. Derrick, "From ODP viewpoint consistency to Integrated Formal Methods," Computer Standards & Interfaces, vol. 35, pp. 269-276, 3// 2013.
[31] Y. Saito and M. Shapiro, "Optimistic replication," ACM Comput. Surv., vol. 37, pp. 42-81, 2005.
[32] S. Goswami and C. Kundu, "XML based advanced distributed database: implemented on library system," International Journal of Information Management, vol. 33, pp. 28-31, 2// 2013.
[33] M. T. Ozsu, Principles of Distributed Database Systems: Prentice Hall Press, 2007.
[34] J. Ma, W. Liu, and T. Glatard, "A classification of file placement and replication methods on grids," Future Generation Computer Systems, vol. 29, pp. 1395-1406, 8// 2013.
[35] B. Kemme, R. J. Peris, and M. Patio-Martnez, Database Replication: Morgan and Claypool Publishers, 2010.
[36] E. Atsan and Ö. Özkasap, "SCALAR: Scalable data lookup and replication protocol for mobile ad hoc networks," Computer Networks, vol. 57, pp. 3654-3672, 12/9/ 2013.
[37] B. Kemme and G. Alonso, "A new approach to developing and implementing eager database replication protocols," ACM Trans. Database Syst., vol. 25, pp. 333-379, 2000.
[38] T. Amjad, M. Sher, and A. Daud, "A survey of dynamic replication strategies for improving data availability in data grids," Future Generation Computer Systems, vol. 28, pp. 337-349, 2// 2012.
[39] A. Lakshman and P. Malik, "Cassandra: a decentralized structured storage system," SIGOPS Oper. Syst. Rev., vol. 44, pp. 35-40, 2010.
[40] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, et al., "Dynamo: amazon's highly available key-value store," SIGOPS Oper. Syst. Rev., vol. 41, pp. 205-220, 2007.
[41] D. Bermbach, M. Klems, S. Tai, and M. Menzel, "MetaStorage: A Federated Cloud Storage System to Manage Consistency-Latency Tradeoffs," in Cloud Computing (CLOUD), 2011 IEEE International Conference on, 2011, pp. 452-459.
[42] L. Dingding, L. Xiaofei, J. Hai, Z. Bingbing, and Z. Qi, "A New Disk I/O Model of Virtualized Cloud Environment," Parallel and Distributed Systems, IEEE Transactions on, vol. 24, pp. 1129-1138, 2013.
[43] F. Pedone and A. Schiper, "Generic Broadcast," in Distributed Computing. vol. 1693, P. Jayanti, Ed., ed: Springer Berlin Heidelberg, 1999, pp. 94-106.
[44] M. Aguilera, C. Delporte-Gallet, H. Fauconnier, and S. Toueg, "Thrifty Generic Broadcast," in Distributed Computing. vol. 1914, M. Herlihy, Ed., ed: Springer Berlin Heidelberg, 2000, pp. 268-282.
[45] A. Sousa, J. Pereira, F. Moura, and R. Oliveira, "Optimistic total order in wide area networks," in Reliable Distributed Systems, 2002. Proceedings. 21st IEEE Symposium on, 2002, pp. 190-199.
[46] P. Felber and A. Schiper, "Optimistic active replication," in Distributed Computing Systems, 2001. 21st International Conference on., 2001, pp. 333-341.
[47] Y. Xinfeng, "Providing Reliable Web Services through Active Replication," in Computer and Information Science, 2007. ICIS 2007. 6th IEEE/ACIS International Conference on, 2007, pp. 1111-1116.
[48] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," SIGOPS Oper. Syst. Rev., vol. 37, pp. 29-43, 2003.
[49] L. Wenhao, Y. Yun, and Y. Dong, "A Novel Cost-Effective Dynamic Data Replication Strategy for Reliability in Cloud Data Centres," in Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on, 2011, pp. 496-502.

[50] W. Qingsong, B. Veeravalli, G. Bozhao, Z. Lingfang, and F. Dan, "CDRM: A Cost-Effective Dynamic Replication Management Scheme for Cloud Storage Cluster," in Cluster Computing (CLUSTER), 2010 IEEE International Conference on, 2010, pp. 188-196.
[51] K. Ranganathan and I. Foster, "Identifying Dynamic Replication Strategies for a High-Performance Data Grid," in Grid Computing — GRID 2001. vol. 2242, C. Lee, Ed., ed: Springer Berlin Heidelberg, 2001, pp. 75-86.
[52] D. T. Nukarapu, T. Bin, W. Liqiang, and L. Shiyong, "Data Replication in Data Intensive Scientific Applications with Performance Guarantee," Parallel and Distributed Systems, IEEE Transactions on, vol. 22, pp. 1299-1306, 2011.
[53] S. Choi and H. Youn, "Dynamic hybrid replication effectively combining tree and grid topology," The Journal of Supercomputing, vol. 59, pp. 1289-1311, 2012/03/01 2012.
[54] O. Al-Haj Hassan, L. Ramaswamy, J. Miller, K. Rasheed, and E. R. Canfield, "Replication in Overlay Networks: A Multi-objective Optimization Approach," in Collaborative Computing: Networking, Applications and Worksharing. vol. 10, E. Bertino and J. D. Joshi, Eds., ed: Springer Berlin Heidelberg, 2009, pp. 512-528.
[55] J. Lloret, M. Garcia, J. Tomas, and J. J. P. C. Rodrigues, "Architecture and protocol for intercloud communication," Information Sciences, vol. 258, pp. 434-451, 2/10/ 2014.