A Replication Model for Trading Data Integrity against Availability

Johannes Osrael, Lorenz Froihofer, Karl M. Goeschka

Vienna University of Technology
Argentinierstrasse 8/184-1, 1040 Vienna, Austria
Phone: +43 1 58801 58409, Fax: +43 1 58801 18491
johannes.osrael|lorenz.froihofer|karl.goeschka@tuwien.ac.at

Abstract

Higher availability and better performance of data-centric applications can be achieved by replication of objects or data items. If data integrity, a correctness criterion for such systems, needs to be maintained even during degraded situations (node or link failures) the system soon becomes (partially) unavailable. However, some applications exist (e.g., in control engineering) where data integrity can be relaxed for higher availability during degraded situations. Traditional replication models do not support the balancing of these two properties. In this paper[1]*, we present a novel replication model that allows replicas to diverge if data integrity can be temporarily relaxed and to re-establish both replica consistency and data integrity during repair time.*

1 Data Integrity and Replication

One correctness criterion for data-centric applications are data integrity constraints, such as value constraints, relationship constraints (cardinality, XOR), uniqueness constraints and other predicates. A system is *constraint consistent* if all data integrity constraints are satisfied.

Replication is one of the primary mechanisms to enhance availability and performance of distributed systems. If strict replica and constraint consistency have to be ensured all the time—even in the presence of failures—the system becomes (at least partially) unavailable in degraded scenarios since neither potentially conflicting updates on replicas in different partitions nor updates that possibly violate data integrity constraints are allowed.

On the other hand, some applications (e.g. [1, 2]) exist where consistency can be temporarily relaxed in order to achieve higher availability. Traditional replication models do not support the configuration of this trade-off; thus, a novel adaptive replication model is required that enables the explicit balancing between availability and consistency.

2 Availability/Consistency Balancing Replication Model

The key idea of our *Availability/Consistency Balancing Replication Model* is to enhance availability of traditional models by allowing non-critical operations in degraded situations in *all* partitions, even if replicas might diverge and data integrity constraints are possibly violated (threatened). Different *reconciliation policies* are required to re-establish replica and constraint consistency after nodes rejoin.

Our new replication protocols distinguish three modes of operation: normal mode, degraded mode, and reconciliation mode. The current mode of the replication protocol depends on the system state, as it is locally perceived by each node. Our replication protocols are in the *normal mode* when all nodes are reachable and all constraints are satisfied, i.e., no partitions are present and all repair activities (reconciliation) are finished.

The protocols switch into the *degraded mode* when not all nodes are reachable. Since node and link failures cannot be distinguished until repair time, node failures are treated as network partitions.

The protocols enter *reconciliation mode* when two or more partitions rejoin. The objective of reconciliation is to re-establish replica and constraint consistency of the sys-

[1] This work has been partially funded by the European Community under the FP6 IST project DeDiSys (Dependable Distributed Systems, contract number 004152, http://www.dedisys.org).

tem. However, system-wide consistency can only be re-established if all nodes are reachable. Thus, if partitions rejoin but the merged partition does not contain all nodes, either constraint consistency is re-established within the partition or constraint consistency is ignored and only replica consistency is re-established.

Adaptive Voting [3], which we briefly describe in the next section, and the Primary-per-Partition Protocol [4] are concrete protocols that follow this model.

3 Adaptive Voting - A Concrete Realization of the Model

We enhance availability of traditional voting [5] by allowing non-critical operations even if no quorums exist, i.e., operations are allowed that may violate tradeable constraints but do not affect non-tradeable constraints. Thus our adaptation of quorum consensus for balancing data integrity with availability is called *Adaptive Voting* (AV) [3].

Normal Mode: AV behaves as the traditional voting protocol with the enhancement that invariant constraints are checked in case of write operations: Write operations are performed on a write quorum WQ of replicas and read operations on a read quorum RQ. The quorum conditions $RQ + WQ > N$ and $WQ > \frac{N}{2}$ must be met in order to prevent write-write and read-write conflicts. N is the number of nodes in the system, i.e., we assume all nodes have the same number of votes. Each node hosts a replica of an object.

Degraded Mode: AV allows non-critical operations even if the quorums of the healthy system cannot be acquired. However, within a partition, read-write and write-write conflicts shall be prevented and the tuning of read against write operations shall be supported. Thus, a quorum scheme adapted to the size of the partition is applied. Tentative states are logged.

Reconciliation Mode: No individual node contains the full version history of a partition[2] since updates are performed on a write quorum which is smaller or equal than the number of nodes in the partition. Thus, in order to detect conflicting updates, the version histories need to be calculated based on the (partial) version histories of the nodes. Either *stepwise rollbacks* or *compensation actions* are performed to re-establish data integrity in case of consistency violations. Moreover, the quorums are re-adjusted according to the size of the merged partition and the histories are cleaned up.

4 Related Work

Trading replica consistency for increased availability has been addressed in distributed object systems such as [6, 7]. TACT [8] provides a continuous consistency model based on logical consistency units (*conits*).

Furthermore, different solutions for reconciliation of divergent replicas have been proposed for mobile environments (e.g., Bayou [9]).

All of the above replication and reconciliation approaches have one commonality: In contrast to our approach, they either do not address constraint consistency explicitly or presume strong data integrity.

References

[1] R. Smeikal and K.M. Goeschka. Fault-tolerance in a distributed management system: a case study. In *Proc. 25th Int. Conf. on Software Engineering*, pages 478–483. IEEE CS, 2003.

[2] K. Zagar. Fault tolerance scenarios in control engineering. In P. Cunningham and M. Cunningham, editors, *Innovation and the Knowledge Economy*, pages 1389–1395. IOS Press, 2005.

[3] J. Osrael, L. Froihofer, M. Gladt, and K.M. Goeschka. Adaptive voting for balancing data integrity with availability. In *On the Move to Meaningful Internet Systems: OTM Confederated Int. Workshops Proc.* Springer LNCS, 2006.

[4] J. Osrael, L. Froihofer, K.M. Goeschka, S. Beyer, P. Galdámez, and F. Muñoz. A system architecture for enhanced availability of tightly coupled distributed systems. In *Proc. 1st Int. Conf. on Availability, Reliability, and Security*. IEEE, 2006.

[5] D.K. Gifford. Weighted voting for replicated data. In *Proc. 7th ACM Symp. on Operating Systems Principles*, pages 150–162. ACM Press, 1979.

[6] P. Felber and P. Narasimhan. Reconciling replication and transactions for the end-to-end reliability of corba applications. In *Proc. Confederated Int. Conf. DOA, CoopIS and ODBASE 2002*, pages 737–754. Springer, 2002.

[7] Y. Ren, D.E. Bakken, T. Courtney, M. Cukier, D.A. Karr, P. Rubel, C. Sabnis, W.H. Sanders, R.E. Schantz, and M. Seri. Aqua: An adaptive architecture that provides dependable distributed objects. *IEEE Trans. on Computers*, 52(1):31–50, Jan. 2003.

[8] H. Yu and A. Vahdat. Design and evaluation of a conit-based continuous consistency model for replicated services. *ACM Trans. Comput. Syst.*, 20(3):239–282, 2002.

[9] D. B. Terry, M. M. Theimer, K. Petersen, A. J. Demers, M. J. Spreitzer, and C. H. Hauser. Managing update conflicts in bayou, a weakly connected replicated storage system. In *Proc. 15th ACM Symp. on Operating Systems Principles*, pages 172–182, 1995.

[2] Except a read-one/write-all scheme is applied in a partition.