Correctness Criteria for Replicated Database Systems with Snapshot Isolation Replicas*

J.E. Armendáriz-Iñigo, J.R. Juárez-Rodriguez, and J.R. González de Mendivil
Universidad Pública de Navarra, Spain
{enrique.armendariz, jr.juarez, mendivil}@unavarra.es

F.D. Muñoz-Escoí
Instituto Tecnológico de Informática, Spain
fmunyoz@iti.upv.es

ABSTRACT
In this work, we present the correctness criteria that ensures a replicated database behaves like a single copy where transactions see a weaker form of SI, called Generalized-SI, with deferred update protocols in a crash failure scenario.

Categories and Subject Descriptors: C.2.4 [Distributed Systems]: Distributed databases; H.2.4 [Systems]: Transaction processing

General Terms: Algorithms, Theory, Verification

Database replication with Snapshot Isolation (SI) replicas has become very popular to enhance system performance (specially for read-only transactions that never block) and afford site failures. However, there must be some coordination mechanism so that each transaction is committed at each database replica in some consistent order which is done by a replication protocol. Several replication protocol proposals [1, 3] have been presented so far. These replication protocols are a sort of distributed algorithm, though no previous work uses a tool that formally specifies the correctness criteria under the simplest failure model scenario such as the crash one. In this brief announcement, due to space limitations, we outline the proposal of using the I/O Automaton Model [4] to provide a new insight in the specification of a database replication system, i.e. formalizing replication protocols with SI replicas and crash failures. Hence, it represents a case study that shows how to reason about a system in an abstract and compositional way. This has been thoroughly explained in [2].

The replicated database system is composed of |N| sites. We assume that at most f (|N| > f) sites may crash. At each site n ∈ N there is an extended database module denoted EDBn. The system is shown as the composition of an abstract replication tool (RP) and the set of databases: RDBS = RP × (Πn∈N EDBn). Let T be the set of system transactions. Each transaction t ∈ T is sent to a unique site where it is firstly executed (local at that one and remote otherwise). The signature of the EDB module is: \( \text{in}(EDB) = \{ \text{commit}(t, ws), \text{apply}(t, ws) : t ∈ T, ws ∈ 2^V \} \) and, \( \text{out}(EDB) = \{ \text{begin}, \text{committed}, \text{aborted} : t ∈ T \} \cup \{ \text{deliverws}(t, ws) : t ∈ T, ws ∈ 2^V \} \). It works in collaboration with RP and provides the action \( \text{apply}(t, ws) \) to the latter as a facility for committing a remote update transaction t on behalf of RP; thus, ensuring t will eventually commit. On the other hand, RP is responsible for guaranteeing the correctness criteria in the whole system and its signature is: \( \text{in}(RP) = \bigcup_{n ∈ N} \{ \text{out}(EDB_n) \} \cup \{ \text{crash}_n \} \); and, \( \text{out}(RP) = \bigcup_{n ∈ N} \{ \text{commit}_n(t), \text{apply}_n(t, ws) : n ∈ N, t ∈ T, ws ∈ 2^V \} \).

These criteria ensure that: it is respected the behavior of each database replica; transactions are applied in the very same order to generate the same global set of snapshots; transactions are either committed, or aborted, at all replicas or none; transactions must be allowed to progress at correct replicas. We formalize them as axioms for every behavior \( \beta \) of RDBS (the predicate local(\( (n, \beta) \equiv \text{begin}_n(t) \leq \text{acts}(EDB_n, t) \) used to denote that t is local at n in \( \beta \)).

Well-formedness Conditions: (a) \( EDB_n \in \text{beh}(EDB_n) \); (b) local(\( (t, n, \beta) \land \text{local}(t, n', \beta) \Rightarrow n = n' = \text{site}(t) \)); and, (c) \( \pi_i = \text{apply}_n(t, ws) \Rightarrow \exists k : i < k : \pi_k = \text{deliverws}_{\text{site}(t)}(t, ws) \land n \neq \text{site}(t) \).

Conflict Serializable. (a) \( \pi_i = \text{apply}_n(t, ws), \text{commit}(t, ws') \land \pi_j = \text{apply}_n(t', ws') \land i < j \land ws \cap ws' \neq \emptyset \Rightarrow \exists k : i < k < j : \exists \pi_k \in \{ \text{committed}_n(t), \text{aborted}_n(t) \} \land, (b) \pi_i = \{ \text{apply}_n(t, ws), \text{commit}_n(t, ws) \} \land \pi_{j_1} = \text{begin}_n(t') \land \pi_{j_2} = \text{commit}_n(t', ws) \land i < j_2 \land ws \cap ws' \neq \emptyset \Rightarrow \exists k : i < j_2 \land j_1 : \exists \pi_k \in \{ \text{committed}_n(t), \text{aborted}_n(t) \} \).

Uniform Prefix Order Database Consistency. For every finite prefix \( \beta' \) of \( \beta : \text{log}(\beta')(EDB_n) \leq \text{log}(\beta')(EDB_n) \) or vice versa.

Uniform Decision. (a) \( \pi_i \in \{ \text{committed}_n(t) \Rightarrow \forall n' \in N \land (\exists k : \exists \pi_k \in \{ \text{committed}_n(t), \text{aborted}_n(t) \} \land, (b) \pi_i = \text{aborted}_{\text{site}(t)}(t) = \{ \beta(\text{apply}_n(t, ws) : n \in N, ws \in 2^V \} \).

Local Transaction Progress. \( \pi_i = \text{deliverws}_{\text{site}(t)}(t, ws) \Rightarrow \exists k : k > i : \pi_k \in \{ \text{crash}_{\text{site}(t)}, \text{commit}_{\text{site}(t)}(t, ws) \} \land \pi_k \in \{ \text{commit}_{\text{site}(t)}(t', ws'), \text{deliverws}_{\text{site}(t)}(t', ws') : ws \cap ws' \neq \emptyset \} \).

Up to our knowledge, this has been the first correctness criteria proposal for SI replicas. Moreover, as proven in [2], the last criterion is important for ensuring liveness in real implementations [3] and should be taken into account.

1. REFERENCES

Copyright is held by the author/owner(s).
PODC’08, August 18–21, 2008, Toronto, Ontario, Canada.
ACM 978-1-59593-989-0/08.

*This work has been supported by the Spanish Government under research grant TIN2006-14738-C02.