Errata for "Unifying Consensus and Atomic Commitment for Effective Cloud Data Management"

Sujaya Maiyya*, Faisal Nawab†, Divyakant Agrawal*, Amr El Abbadi*
*UC Santa Barbara, †UC Irvine
*{sujaya_maiyya, divyagrawal, elabbadi}@ucsb.edu, †{nawabf@uci.edu}

ABSTRACT

This errata article discusses and corrects a minor error in our work published in VLDB 2019. The discrepancy specifically pertains to Algorithms 3 and 4. The algorithms presented in the paper are biased towards a *commit* decision in a specific failure scenario. We explain the error using an example before correcting the algorithm.

PVLDB Reference Format:
Sujaya Maiyya, Faisal Nawab, Divyakant Agrawal, Amr El Abbadi. Errata for "Unifying Consensus and Atomic Commitment for Effective Cloud Data Management". PVLDB, 14(7): 1166 - 1166, 2021.
doi:10.14778/3450980.3450985

1 PROBLEM

This article is a response to Chockler and Gotsman [1] who raised a minor error in our work [2]. We explain the problem in Algorithm 3 of [2]. The algorithm elects a leader and identifies the *commit* or *abort* value for a terminating transaction. In both the normal execution as well as failure recovery, every newly elected coordinator executes Algorithm 3 to choose a value to propose. Now consider the following scenario where 5 sites are attempting to terminate a transaction:

(1) Suppose that all initial values are commit. A commit is selected to be proposed by the first leader $\mathcal{L}1$, but it fails after setting AcceptVal = commit at just a single cohort, $C1$.
(2) Another leader, $\mathcal{L}2$, gets elected, but now the cohort $C1$ with AcceptVal = commit stops responding. Leader $\mathcal{L}2$ thus picks abort as its value since no cohort responded with its Decision or AcceptVal set. Leader $\mathcal{L}2$ proceeds to set AcceptVal = abort at a quorum that does not include the cohort $C1$. Then, it manages to decide on abort (i.e., Decision = True and AcceptVal = abort) at exactly one cohort, $C2$, and crashes.
(3) The third leader $\mathcal{L}3$ gets elected, and manages to receive a response from cohort $C1$ with AcceptVal = commit, but not from $C2$ that has both Decision = True and AcceptVal = abort set. For example, it may hear from a quorum which includes one cohort $C3$ with AcceptVal = abort (since that was finalised at a quorum by the second leader) and one cohort $C1$ with AcceptVal = commit (the one set by the first leader). According to the code in Algorithm 3, $\mathcal{L}3$ will adopt commit as its AcceptVal value (line 5) even though cohort $C3$ has a higher AcceptNum than $C1$, and $\mathcal{L}3$ may subsequently commit a transaction that was already aborted.

While Algorithm 3 is used for PAC, the same behavior (and hence the error) exists in Algorithm 4 used for G-PAC. We note that the error does not affect the performance evaluations presented in the paper.

2 SOLUTION

The error is caused mainly due to the condition in Line 5 of Algorithm 3 (and Line 6 of Algorithm 4). To mitigate the above discussed problem, lines 5-7 in Algorithm 3 of the paper must be changed to:

Correction to Algorithm 3

```
5: else if at least one response with AcceptVal≠ ⊥
6: /* Decision is True for none in the received responses. */
   then
7:     AcceptVal ← AcceptVal of the highest AcceptNum
```

Note that this change in condition is similar to the corresponding condition in Paxos. In the example discussed earlier, if the changed pseudocode is used, the third leader $\mathcal{L}3$ will propose AcceptVal = abort. This is because the AcceptNum of cohort $C1$ (with AcceptVal = commit set by the first leader) is strictly lower than the AcceptNum of cohort $C3$ (with AcceptVal = abort set by the second leader). The higher AcceptNum of $C3$ is guaranteed due to quorum intersection: at least one site knows the ballot *B1* of the first leader, and any successful new leader (e.g., $\mathcal{L}2$) must have a ballot $B2 > B1$ (otherwise the overlapping site will reject the second leader's request). By picking the value with the highest AcceptNum, a decided value will not be overridden. In the example, $\mathcal{L}3$ picks abort as the chosen value.

A similar correction is also applied to Algorithm 4.

Correction to Algorithm 4

```
6: else if a majority of replicas of at least one shard respond
   and at least one of them has AcceptVal ≠ ⊥
7: /* Decision is True for none in the SUPER-MAJORITY. */
   then
8:     AcceptVal ← AcceptVal of the highest AcceptNum across
       all the received responses
```

REFERENCES

[1] Gregory Chockler and Alexey Gotsman. [n.d.]. Private Communication. ([n. d.]).
[2] Sujaya Maiyya, Faisal Nawab, Divyakant Agrawal, and Amr El Abbadi. 2019. Unifying consensus and atomic commitment for effective cloud data management. *Proceedings of the VLDB Endowment* 12, 5 (2019), 611–623.

Proceedings of the VLDB Endowment, Vol. 14, No. 7 ISSN 2150-8097.
doi:10.14778/3450980.3450985