Petr Kuznetsov
Telecom Paris, Institut Polytechnique de Paris
petr.kuznetsov@telecom-paris.fr

04/03/2022

Review
on Ph.D. dissertation authored by

Maciej Kokocinski

entitled:

Correctness of Highly-Available Eventually-Consistent Replicated Systems

1. Problem and its impact

The focus of this thesis is consistency in replicated systems. There is a fundamental conflict between availability of data in large-scale fault-prone distributed systems and the level of consistency the system users can enjoy.

Strong consistency guarantees inherently involve expensive synchronization techniques. In the extreme case when data replicas have to agree on a total order of operations issued by the users (the problem known as *state-machine replication* or, more recently, *blockchain*), a notoriously costly and slow *consensus* protocol must be used. Fortunately, many applications can live with relaxed forms of operation ordering. At the opposite extreme, the application may tolerate arbitrary reordering for unbounded periods of time and may only require *eventual* consistency.

The thesis takes a theoretical viewpoint on a practically inspired question: what is the effect various consistency guarantees may have on the availability of data in a replicated system? The thesis studies a spectrum of “middle-ground” consistency criteria which combine weak but highly-available operations with strong but expensive ones. The thesis describes certain inherent vulnerabilities of existing criteria and proposes several new ones.

2. Contribution

The principal result of the thesis is a streamlined analysis of *mixed* consistency.

Existing academic proposals for consistency criteria that combine strong and weak operations either overlook certain desirable application requirements or are too complicated to be used in practice. Existing attempts to implement this idea in industrial replicated services either allow for unspecified behavior or miss the point of mixing, e.g., by splitting the data into strongly-consistent and weakly-consistent partitions, by only providing strong consistency for updates, or by resorting to strong assumptions about the environment.

The thesis introduces (in Chapter 2) a novel abstraction: *acute cloud types* (ACT) that combine weak and strong operations. Intuitively, in any execution of an ACT implementation, there must exist a total order of all operations S, such that every strong operation “witnesses” S (i.e., its result is consistent with S) and every weak operation witnesses some S’ that diverges from S by only finitely many operations. Interesting, classical consistency properties are typically split into safety(e.g., linearizability) or liveness (e.g., eventual consistency). In contrast, ACT is a mixture of safety and liveness, which matches the intuition of “mixed consistency”.

As a running example of ACT, the thesis uses ACCN (Acute Non-Negative Counter). The ACT abstraction is also used in the context of Bayou, a prominent eventually consistent database, giving rise to AcuteBayou ACT. A model which enables formal reasoning about ACT implementations is given in Chapter 3.

In Chapter 4, the thesis discusses inherent issues incurred by ACT-like systems. It is shown that any system mixing desirable properties of weak and strong operations (ACT included) must exhibit temporary operation reordering, a fascinating phenomenon which intuitively shows that there is a cost to pay for introducing strong operations into an eventually consistent system. Potential consistency relaxations that allow for circumventing this limitation are also discussed.

Chapter 5 studies an interesting categorization of failure models, based on three machine failure types and two network failure types, and explores how these failure models correspond to the system’s consistency.

Chapter 6 studies the phenomenon of *session guarantees* capturing interactions of clients and replicas in a fault-tolerant replicated system. In particular, the chapter studies the interesting question of whether a *stateless* client can be maintained assuming fault-prone replicas. A novel session guarantee, called *context preservation*, is introduced and related to session guarantees from the literature.

Chapter 7 explores the extent to which different kinds of failures can affect progress guarantees of a replicated system and introduces the notion of a failure-aware correctness criteria.

The results of the first three technical chapters of the thesis were presented two brief announcements at the top conference in distributed computing (ACM PODC) and a journal article in IEEE TPDS, one of the major magazines in the domain. The remaining results (Chapters 5-7) are currently under preparation to be submitted.

I found the proposed notion of ACT elegant. The formal framework proposed for ACT analysis enable the discovery of the fascinating phenomenon of paying the price of temporary operation reordering in replicated systems that combine strong (*linearizable*) and weak (*eventually consistent*) operations. The discussion on client sessions, and the effect faults can have on safety and liveness of a replicated system is illuminating.

3. Correctness

The results are properly stated, and apparently correct proofs are provided. The intuition prefacing the formal claims and proofs is convincing. The thesis results are purely theoretical, the findings are not supported by implementations and performance analyses.

4. Knowledge of the candidate

The introduction gives a concise but comprehensive overview of the major challenges faced by large-scale replicated systems. In particular, the thesis creates a convincing case for mixed consistency that combines features of weak, eventually consistent operations and strong, consensus-based ones. The thesis in general gives the impression that the candidate masters his topic of research: consistency and availability of replicated data.

The related work is adequately discussed and all relevant articles in the domain are referenced.

I am also aware of the work the candidate has been involved in outside the scope of this thesis: deferred update in transactional systems and concurrent skip lists. Overall, I am convinced that Maciej matches the profile of an expert in Information and Communication Technology.

5. Other remarks

I only have minor remarks that do not have any major effect on my evaluation of this thesis.

- The introduction could be slightly better in creating a "big picture", unifying all the diverse results of the thesis. The choice of topics of Chapters 5-7 appears random. In the context of the thesis, is there a particular motivation of focusing in session guarantees? On failure-aware correctness? A unifying perspective would also help at the defence.
- In the same vein, one of the most interesting features of a PhD thesis is "future work". What did we learn from the thesis, what are the most interesting and challenging question it rises? The last paragraph of the conclusion is a bit too short and too vague. For example, a major theme of the thesis is the relationship between model assumptions (faults or client profiles) and the consistency criteria. This raises the question of precise *characterization* of the model in which a given consistency criterion can be ensured.
- Algorithm 1: One should specify the initial values of weak/strong Add/Sub variables.
- Before jumping into details of Bayou (Section 2.2), it would make sense to specify first what the system is supposed to implement. What is the system interface and what properties Bayou designers intended to guarantee?
- We should be more careful with the use of terms "wait-freedom" and "bounded wait-freedom" in the model considered in this thesis (Section 2.2.5). Wait-freedom was introduced by Herlihy for the shared-memory context and it literally means that a process may make progress *without waiting* for other processes to move. In our case, this is impossible: normally, a process must wait until a quorum of replicas react.
- Chapter 7: I would not insist that all practical types are non-trivial, as defined here. There are types that do not separate operations in read-only and updates. For example, conditional types such as TAS and CAS only have updates, but some of them fail in some states. I guess your results would hold for such types too.

6. Conclusion

Taking into account what I have presented above and the requirements imposed by Article 13 of *the Act of 14 March 2003 of the Polish Parliament on the Academic Degrees and the Academic Title* (with amendments)[1], my evaluation of the dissertation according to the three basic criteria is the following:

A. Does the dissertation present an original solution to a scientific problem? (the selected option is marked with **X**)

☒	□	□	□	□
Definitely YES	*Rather yes*	*Hard to say*	*Rather no*	*Definitely NO*

B. After reading the dissertation, would you agree that the candidate has general theoretical knowledge and understanding of the discipline of **Information and Communication Technology**, and particularly the area of **theory of distributed computing?**

☒	□	□	□	□
Definitely YES	*Rather yes*	*Hard to say*	*Rather no*	*Definitely NO*

C. Does the dissertation support the claim that the candidate is able to conduct scientific work?

☒	□	□	□	□
Definitely YES	*Rather yes*	*Hard to say*	*Rather no*	*Definitely NO*

Moreover, given the soundness of the results and the elegance of their presentation, I **recommend to distinguish** the dissertation for its quality.

Signature

[1] http://www.nauka.gov.pl/g2/oryginal/2013_05/b26ba540a5785d48bee41aec63403b2c.pdf