

Understanding Eventual Consistency

Summary of the presentation held on June 17th, 2014

Horst Schneider, Patrick Beedgen

June 17th, 2014

Modern large-scale, geo-replicated Web Applications depend on databases that are highly available and can be scaled largerly. Following the CAP theorem, this can not be achieved without constraints in consistency. Geo-replicated databases like Amazon DynamoDB, MongoDB, CouchDB, and many others offer consistency models that are commonly described with the term *eventual consistency*. While some of these databases offer similar features, it is not possible to compare features of different systems semantically, as different formalisms are used, weak guarantees are being made and handling of conflict resolution varies greatly.

Based on [BGY13], this presentation is going to suggest a rigorous way to specify the semantics of geo-replicated databases. After presenting a function that is general enough to specify the semantics of any replicated data type, different conflict resolution strategies from real-world applications are identified. To show the validity of the replicated data type specification, examples are given to show how those strategies can be specified using the presented formalism. In a next step, the semantics of individual objects are extended to the whole database by defining sessions and a database history that provide a context allowing us to formalize features and guarantees of the whole database, considering interactions in different sessions and on different replicas.

With the help of these defined datatypes and their semantics in a so-called *operation context*, different levels of eventual consistency are being formalized. Those consistency levels will be grouped into three forms:

- basic Eventual Consistency
- session guarantees
- causal consistency

Each form is defined by multiple consistency axioms that provide stronger consistency with every axiom chosen. Every form builds upon the latter one, meaning, that session guarantees can't be achieved without providing basic Eventual Consistency. From every form there will be picked selected axioms

to show the issues of available and partition-tolerant databases that are being addressed by them.

The final part of the presentation will revolve around the idea of strengthening consistency on-demand. A real world example will be used to demonstrate the advantages of this concept.

After emphasizing its uses, the two concepts mentioned in [BGY13] of how to strengthen consistency on demand are being elaborated upon.

The presentation will be finalized by showing advantages and disadvantages of the shown concepts and by discussing them with the plenum.

Literatur

- [BGY13] Sebastian Burckhardt, Alexey Gotsman, and Hongseok Yang. Understanding eventual consistency. Technical Report MSR-TR-2013-39, Microsoft Research, One Microsoft Way, Redmond, WA 98052, 2013.
- [dyn12] Amazon DynamoDB Developer Guide. <http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/APISummary.html>, 2012. [Online; accessed 16-June-2014].
- [mon10] On Distributed Consistency. <http://blog.mongodb.org/post/498145601/on-distributed-consistency-part-2-some-eventual>, 2010. [Online; accessed 16-June-2014].
- [Vog09] Werner Vogels. Eventually consistent. *Commun. ACM*, 52(1):40–44, January 2009.