On the Consistability of Storage Systems

Amitanand Aiyer* Eric Anderson Xiaozhou Li Mehul Shah Jay J. Wylie
anand@cs.utexas.edu, {eric.anderson4, xiaozhou.li, mehul.shah, jay.wylie}@hp.com
Hewlett-Packard Laboratories

Modern Internet-scale storage systems typically provide different consistency guarantees under different operating conditions (i.e., failure scenarios). Under the fault-free condition, the system may provide strong guarantees such as atomic consistency. However, under failure conditions (e.g., network partitioning), the system often sacrifices consistency in exchange for higher availability. For example, under network partitioning, the system may only provide eventual consistency. After all, by Brewer's famous theorem, if one has to live with network partitioning, then one has to sacrifice either consistency or availability. The current practice is to only describe the weakest consistency provided by a storage system, although for a considerable amount of time, the system may operate under good conditions and provide stronger consistency. This makes the specification imprecise and focuses exclusively on the worst-case. Such specifications make it hard to compare two systems in detail.

To address this problem, we have introduced the concept of *consistability* [2]. Inspired by and analogous to the notion of performability, consistability tries to capture the fact that a storage system provides different kinds of consistency under different operating conditions. At a high level, the consistability of a system describes what kinds of consistency are achieved by the system under each operating condition. If one can come up with the portion of time each operating condition occurs (by measuring, modeling, or estimating), then one can calculate how often a system achieves a certain consistency. For example, suppose a system provides atomic consistency under fault-free conditions, which is estimated to be around 80% of the time, and provides regular consistency under all faulty conditions, which is estimated to be around 20% of the time. Then the system provides atomic consistency 80% of the time, but regular consistency 100% of the time. (It remains an open problem how to detect the transitions between different consistencies and how to reason about the system's behavior during a transition.) Consistability makes the specification of a system more precise and facilitates the comprehensive comparison of systems. For example, to compare two systems, a user can assign importance to each consistency level and determine which system achieves the more important levels more often.

We are using consistability to understand the design trade-offs in the development of a scalable key-value storage system. This system provides a simple get-put interface, is designed to scale across multiple data centers, and is able to sustain various kinds of failure scenarios such as disk failures, data center failures, and network partitioning. For each failure scenario, we investigate what kinds of consistency are achievable. For example, for disk failures and data center failures, we expect that regular consistency (i.e., a get returns a value most recently put into the system) is achievable. However, for network partitioning, we believe that the strongest consistency achievable is k-regularity: a get returns a value that associates with one of the last k puts [1]. Unfortunately, k may be unbounded because the network can remain partitioned for a long time. In other words, we may only be able to achieve ∞-regularity under network partitioning. We are currently verifying our get-put protocol achieves certain consistency levels for various failure scenarios using Lamport's TLC model checker. We are also investigating how slight variations of the protocol considerably affect the consistencies achieved under some specific failure scenario.

*student

References

[1] Amitanand Aiyer, Lorenzo Alvisi, and Rida A. Bazzi. On the availability of non-strict quorum systems. In *Proc. 19th DISC*, pages 48–62, September 2005.

[2] Amitanand Aiyer, Eric Anderson, Xiaozhou Li, Mehul Shah, and Jay J. Wylie. Consistability: Describing usually consistent systems. In *HotDep 08*, December 2008.