Chapter 6
Selective Data Consistency Model in No-SQL Data Store

Shraddha Pankaj Phansalkar
Symbiosis International University, India

Ajay Dani
G. H. Raisoni Institute of Technology, India

ABSTRACT
Contemporary web-applications are deployed on the cloud data-stores for realizing requirements like low latency and high scalability. Although cloud-based database applications exhibit high performance with these features, they compromise on the weaker consistency levels. Rationing the consistency guarantees of an application is a necessity to achieve the augmented metrics of application performance. The proposed work is a paradigm shift from monotonic transaction consistency to selective data consistency in web database applications. The selective data consistency model leverages consistency of critical data-objects and leaves consistency of non-critical data-objects to underlying cloud data-store; it is called selective consistency and it results in better performance of the cloud-based applications. The consistency of the underlying data-object is defined from user-perspective with a user-friendly consistency metric called Consistency Index (CI). The selective data consistency model is implemented on a cloud data-store with OLTP workload and the performance is gauged.

INTRODUCTION
Commercial big data stores and big data analytics have become popular among enterprises because of their assurance of providing unlimited scalability and availability. Their application is from areas like emotion and sentiment analysis (Alvandi, 2011) to daily monitoring of urban traffic (Meier and Lee, 2009). Database applications in the cloud environment are required to be available 24x7 to serve the demands of the users across the globe. The databases of such applications are simulated across the world to guarantee high accessibility of data, low latency time and cost of accessibility. The CAP theorem (Brewer, 2000) states that the provision of strong consistency plus high scalability simultaneously, is improbable in the existence of network partitioning. The state-of-the-art web applications compromise
Selective Data Consistency Model in No-SQL Data Store

data consistencies to a definite level and thus attain high performance and availability. The design of the databases of these web applications can be considered as an optimization problem (Redding et al., 2009), where the desired performance level is achieved with optimized levels of consistency. This work introduces user perspective of data consistency and measures it by a comprehensive quantitative metric known as Consistency Index (CI). Any user can realize his consistency requirements objectively, if consistency is quantitatively measured and indexed (by some index measure). It must be a simple, flexible metric that is independent of application. It is supposed to be such, which can have applicability to database objects with different granularity from attributes (i.e. database fields) and database rows (objects) to database tables (collection). This metric realizes the user perspective of consistency. CI can be used to develop intelligent transaction scheduling strategies which can optimize the performance of applications.

No-SQL data stores offer the augmented metrics of performance with different levels of consistency and lower isolation levels. Table 1 shows the summary of the popular No-SQL data stores with respect to consistency, replication, availability, fault tolerance and scalability guarantees.

Consistency of data is the consensus of the data across multiple copies (replicas) in the system in the replicated databases (Padhye, 2014). This is the database perspective of the data consistency problem. However, in contemporary web applications, maintaining strict consistency throughout the database is difficult, if requirements like low query response time and high availability of data are to be achieved with higher priority. Besides these, security and privacy of huge data sets against intrusive threats is elucidated in works by Mohanpurkar and Joshi (2016), Fouad et al. (2016) and Odella (2016) which poses a new challenge to big data world. The articles like Curran et al. (2014) also emphasize on related techniques and tools to carry location based tracking of objects in different environments.

CAP theorem by Brewer (2000) states that consistency and availability are inversely proportional in a partitioned system. Hence consistency needs to be rationed to weaker levels in case of web applications with high availability requirements. The positive impact of consistency rationing (accepting weaker levels of consistency) on the system performance has been studied in Koroush (1995).

The idea of tradeoff between consistency, response time and high availability has been proposed in (Redding et al., 2009). All the levels of consistency like sequential, serializable, eventual, release consistency are discussed for replicated distributed database systems in transactional as well as non-transactional context. The researchers have planned different approaches to ration consistency and progress the performance of web-based functions in transactional or non-transactional circumstance.

Use of replicated data stores is mandatory for improving application performance with respect to availability and reliability. Consistency rationing is thus highly desirable in replicated data stores.

In Olston and Widom (2000), the authors have proposed a replicated system which works on a combination of thresholds in cache and server data to deliver an answer in tolerable limit. Here a transaction query is attributed with the bounds of staleness to which the system must abide.

Haifeng and Vahdat, in 2002 developed a consistency model to describe data consistency in replicated objects using a new metric called conit which is a combination of three measures viz value error, sequence error and correctness for the replicated data. In this work, a middleware layer that managed consistency bounds among replicated data objects was built with these metrics.

Zhang and Zhang (2003) propose Window Based Quantitative Model, which expresses inconsistency as the proportion of non-consensus in writes. The system specifies the tolerable number of missed updates as the consistency rationing mechanism.

A probabilistic approach uses subset of replicas to find threshold limit of inconsistency with respect to both version number of the replicas and physical time, this is provided by the theory of Probabilisti-