Probabilistic Data Consistency for Wide-Area Applications

Hengming Zou[1], Nandit Soparkar[2], Farnam Jahanian[1]
Department of EECS, University of Michigan, Ann Arbor, MI 48109
{*zou, soparkar, farnam*}*@eecs.umich.edu*

1. Introduction

Wide-area applications, such as electronic commerce, groupware, directory service etc., rely on distributed data management services which often employ data replication to meet availability and response time constraints imposed by the target domains. Since many of these applications can tolerate some staleness in the data they access, the constraints may be better met by exploiting relaxed data consistency among copies in the system. However, in the current Internet architecture, it is nearly impossible to provide deterministic guarantees for the bounds on data inconsistency among copies. We present an approach for providing *probabilistic data access* guarantees while maintaining availability and timing predictability. Our model makes few assumptions regarding the underlying infrastructure. In fact, a novel feature of our approach is that it is independent of the underlying replication scheme. We focus on modeling the update patterns of the servers and modifying the data access mechanisms. Our model guarantees, with a certain probability P, that if a client requests the value of an object, the value returned by the system, if any, will be temporally consistent with the newest copy of the same object in the system. That is, from the perspective of a client, the system provides a probabilistic data consistency guarantee.

2. Summary of results

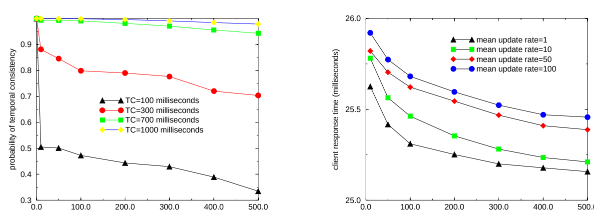
The key in our approach is that each server must be able to independently compute its probability of data consistency. This computation should not depend on the patterns and probabilities of fault occurrences, or any bounds on the communication delays, in the system. Instead, the probability only depends on the update patterns of the servers. Let t_u^i denote the timestamp of the last update for object i on server S before time t, and δ denote the temporal constraint that must be met between any two copies, we found that the probability that the copy of object i on server S is temporally inconsistent at time t is:

$$P_i(t) = \begin{cases} 1 & t \le t_u^i + \delta \\ 0 & t \ge t_u^i + p_i + \delta \\ 1 - p_{u,-s}^i & \text{o.t. \& random update} \\ 1 - (t - \delta - t_u^i)/p_i & \text{o.t. \& periodic} \end{cases}$$

where p_i is the update period for object i in the system (for periodic updates), $p_{u,-s}^i$ is the probability that server 1 made at least one update, or server 2 made at least one update, or ... server $s-1$ made at least one update, or server $s+1$ made at least one update ..., or server n made an update during the concerned time interval for the random update case. The value of $p_{u,-s}^i$ is computed based on the waiting time random sequence model. We omit the detail due to space limit.

3. Experiments and conclusion

We chose to write our own replication protocols in the implementation to take advantage of the temporal consistency concept. Our prototype replication uses the imposed temporal constraint to influence the update rate of objects, and thereby, facilitates better real-time performance with respect to the date access.



(a) Response consistency (b) Response time

Figure 1: Performance under 100 ms roud-time

The performance results presented above show that our system is fast in responding to client requests, keeps a small temporal separation among servers, maintains a high probability of query consistency, and a reasonably low answer to query miss ratio.