1. Summary

This paper studies the synchronization of a local database to improve its freshness and age. A formalized framework is provided to study various synchronization policies. An optimal (resource-allocation) policy is derived and experiments give supports.

In Section 2, the authors present a framework for the synchronization problem. First, 2 metrics (freshness and age) are defined on the level of an element and the whole local database. Then, the change of an element is viewed as a Poisson process on which expected freshness and age could be derived. In addition, evolution model of database is divided into uniform change-frequency one and non-uniform change-frequency one based on assumptions about how uniformly elements in database are being changed.

Section 3 introduces the 4 dimensions of synchronization policies: 1) synchronization frequency, 2) resource allocation, 3) synchronization order, and 4) synchronization points. Synchronization frequency means how frequently the local database is synchronized. Similarly, resource allocation is also about synchronization frequency, but it focuses on elements in the database, and there are uniform and non-uniform allocation policies. Moreover, synchronization order is about the order to synchronize elements in the database, which includes fixed-order, random-order and purely-random styles. Last, synchronization points decide the time-window to synchronize. In this paper it is set uniformly over time.

Then Section 4 and 5 compare different synchronization policies. Discussion on synchronization-order policies shows that fixed-order policy achieves better freshness and age than random-order and purely-random policies under same uniform allocation policy. The mathematical formulas about relations of freshness/age and synchronization-order policies (using change/synchronization frequency ratio as the parameter) are helpful to measure/guarantee the freshness of local database and identify best synchronization policies. And then the topic turns to resource-allocation policies. From a simple and concrete example, it shows under the background of non-uniform change-frequencies of elements, intuitively-better proportional (non-uniform) allocation policy always performs worse for both freshness and age than uniform allocation policy. The optimal resource-allocation policy, which is neither the uniform policy nor the non-uniform policy, is derived to give closed form using the method of Lagrange multipliers.

Finally, in Section 6 analysis on WWW data shows verifies the authors' assumption and observation. The observed change of collected WWW data is subject to Poisson distribution, and different synchronization policies achieve predicted performance.

2. Comments

The paper is strict and reader-friendly. After the introduction of the topic, key concepts including metrics, framework and policies are developed smoothly and broken down carefully. And examples being used are simple and interpretable. Especially, the one of Section 5.2 is helpful to understand how the proportional resource-allocation policy gives less “benefit” than uniform policy when the total synchronization frequency is far smaller than the change frequency.

3. Questions

(1) How can we set the optimal synchronization frequency for the total local database given change frequency? The example of Section 5.3 discusses the optimal allocation of synchronization

frequencies for each element in the local database given the chosen total synchronization frequency, but it does not cover this topic. Maybe details are included in appendixes.