The Journal of Systems and Software 82 (2009) 2012–2022

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

An energy-efficient mobile transaction processing method using random back-off in wireless broadcast environments

Sunggeun Park [a], Sungwon Jung [b,*]

[a] Core Division, TmaxSoft R&D Center, Seoul, Republic of Korea
[b] Department of Computer Science and Engineering, Sogang University, Seoul 121-742, Republic of Korea

ARTICLE INFO

Article history:
Received 1 December 2007
Received in revised form 2 May 2009
Accepted 28 June 2009
Available online 9 July 2009

Keywords:
Mobile databases
Mobile transactions
Wireless broadcast environments
Concurrency control

ABSTRACT

Broadcast is widely accepted as an efficient technique for disseminating data to a large number of mobile clients over a single or multiple channels. Due to the limited uplink bandwidth from mobile clients to server, conventional concurrency control methods cannot be directly applied. There has been many researches on concurrency control methods for wireless broadcast environments. However, they are mostly for read-only transactions or do not consider exploiting cache. They also suffer from the repetitive aborts and restarts of mobile transactions when the access patterns of mobile transactions are skewed. In this paper, we propose a new optimistic concurrency control method suitable for mobile broadcast environments. To prevent the repetitive aborts and restarts of mobile transactions, we propose a random back-off technique. To exploit the cache on mobile clients, our method keeps the read data set of mobile transactions and prefetches those data items when the mobile transactions are restarted. As other existing optimistic concurrency control methods for mobile broadcast environments does, it works for both read-only and update transactions. Read-only transactions are validated and locally committed without using any uplink bandwidth. Update transactions are validated with forward and backward validation, and committed after final validation consuming a small amount of uplink bandwidth. Our performance analysis shows that it significantly decreases uplink and downlink bandwidth usage compared to other existing methods.

1. Introduction

Due to the rapid advance in computer hardwares (e.g. mobile phone, PDA) and wireless network technologies (e.g. bluetooth, IEEE 802.11), mobile computing has become widely available. In the near future, tens of millions of users will be accessing various services anytime and anywhere with a battery power-based portable computer. However, existing mobile technologies have to face several constraints such as limited network bandwidth, frequent disconnections and insufficient battery power. To cope with these constraints, there has been many studies on data transmission techniques using wireless data broadcasting (Acharya et al., 1995; Datta et al., 1999; Hurson et al., 1998; Juran et al., 2000; Lee et al., 2000, 2002). Generally a mobile client sends requests to the server and receives the response. For sending requests, mobile clients have to consume uplink bandwidth. The response time also can dramatically increase when the server is heavily loaded by the requests from a large number of clients. However, wireless data broadcast models can overcome these problems. For example, Acharya proposed the broadcast disks model (Acharya et al., 1995). The server continuously and repeatedly broadcasts all data in the database using a single or multiple wireless communication channels. Clients wait for the data in need to come up on the channel, and retrieve data from the channel. In this system, the number of mobile clients does not affect their access time. With its good scalability, wireless data broadcast is used in various mobile applications, e.g. auctions, electronic bidding, stock trading, weather information and traffic information broadcasts (Cho, 2003).

The above applications comprise a large proportion of read-only transactions and a small number of update transactions. For example, the number of stock buyers or bidders, (i.e., update transactions) in stock trading and auctions is relatively fewer compared to the number of speculators (i.e., read-only transactions) who read the prices frequently. In these applications, the consistency among data items is likely to be violated by update transactions (Lee et al., 2002, 2004; Pitoura et al., 1999; Shanmugasundaram et al., 1999). Thus, a concurrency control scheme is needed to preserve data currency and consistency for mobile transactions. However, conventional concurrency control methods cannot be directly applied to mobile transaction processing (Lee et al., 2004). In conventional

* Corresponding author. Tel.: +82 2 705 8930.
E-mail addresses: sunggeun.park@gmail.com (S. Park), jungsung@sogang.ac.kr (S. Jung).

0164-1212/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2009.06.053

methods, mobile clients have to communicate with the server by exchanging a number of messages, which consumes the limited uplink bandwidth and much battery power of mobile clients (Bernstein et al., 1987; Haerder, 1984; Kung and Robinson, 1981). Furthermore. as most works are done at server-side, servers can be easily overloaded or interrupted by processing transactions from a lot of clients. For these reasons, conventional concurrency control schemes such as two-phase locking and timestamp ordering are not suitable for mobile transaction processing.

There have been many research efforts reported in the literatures that focus on the concurrency control scheme for wireless broadcast environments (Barbara et al., 1997; Chung et al., 2003; Lee et al., 2002, 2004, 2003, 2006; Pitoura et al., 1999; Shanmugasundaram et al., 1999). Lee et al. proposed a variant of the OCC method called FBOCC (forward and backward OCC) that uses forward validation for update transactions and partial backward validations for read-only transactions (Lee et al., 2004). However, all the concurrency control methods for wireless broadcast environments including FBOCC method are focused on mobile transactions with uniform data access patterns. Existing methods perform poorly when the data access patterns of mobile update transactions are skewed. Mobile update transactions will be frequently aborted and restarted in the final validation stage due to the update conflict of the same data items with high access frequencies. This problem will waste the uplink and downlink communication bandwidth, which in turn cause the wasteful battery power consumption of mobile clients.

In addition to the above skewed data access pattern problem, all the mobile concurrency control methods including FBOCC do not utilize caches of mobile clients for their concurrency control. When a mobile client has cache, its access time can be further reduced by directly accessing data items from its cache. Lee et al. proposed a method to reduce the response time of mobile transactions using data prefetching (Lee et al., 2003, 2006). There are two ways to identify the data items for prefetching. The first is that transactions can declare read data set when they starts. The second is that a preprocessor can search for all possible necessary data items in the transaction before it starts. However, the first method is not so practical, and the second method consumes more resources by reading more data than actually required.

In this paper, we propose an efficient mobile concurrency control method for wireless broadcast environments. Our method uses a random back-off technique to overcome the repetitive aborts and restarts problem of mobile update transactions caused by the skewed data access patterns of mobile update transactions. Our method also prefetch data based on the access invariance proposed by Franaszek et al. (1990). The access invariance means that a restarting transaction reads the same data set as the one used in its previous failed execution. As a result, our method does not have much computational overhead for identifying the set of data items for prefetching as in the methods proposed by Lee et al. (2003, 2006). Our method also supports both read-only and update transactions and guarantees the global serializability to keep the database consistent. In our method, by using forward and backward validation, read-only transactions can be locally committed without using any uplink bandwidth and update transactions consume less uplink bandwidth for their validations. Overall, our proposed method has two contributions. The first contribution is that by using the random back-off technique, mobile clients can avoid repetitive aborts of mobile update transactions, resulting in the reduced uplink bandwidth. The second contribution is that by exploiting client's cache, mobile clients can improve the response times of restarted transactions by first reading the prefetched data in their cache.

The rest of the paper is organized as follows: in Section 2, we discuss the related works. We describe the basic idea of our proposed AOCCRB (Adaptive Optimistic Concurrency Control with Random Back-off) method in Section 3. Section 4 gives the formal description of AOCCRB method. We discuss the performance analysis of AOCCRB method in Section 5. Finally, Section 5 provides the concluding remarks.

2. Related works

Barbara proposed a modified version of the conventional optimistic concurrency control (OCC) protocol to support both read-only and update transactions at the mobile clients (Barbara et al., 1997). However, his protocol gives a significant delay to aborted mobile transactions. Shanmugasundaram et al. proposed a weaker correctness criterion called *update consistency* (Shanmugasundaram et al., 1999). Update consistency allows read-only transactions to read current and consistent data in wireless broadcast environments without contacting the server. However, the serializability is not maintained which is very important in mobile commerce applications such as mobile stocks trading. Lee et al. proposed a method to reduce the response time of mobile read-only transactions by using a predeclaration technique (Lee et al., 2003, 2006). However, the predeclaration technique does not support update transactions and the predeclaration process is computationally expensive.

Pitoura and Chrysanthis proposed three broadcast methods to guarantee the correctness of read-only transactions (Pitoura et al., 1999). However, their three methods have the following problems. The *multi-version broadcast* method increases the size of the broadcast cycle resulting in the increased response time. The *invalidation-only broadcast* method has a low concurrency problem. The *conflict serializability* method incurs high overheads to maintain the serialization graph. Furthermore, these three methods do not support update transactions. Chung et al. proposed a transaction processing protocol that increases the autonomy of clients based on the dependency relation among updated data items (Chung et al., 2003). Each mobile client uses the lists of dependents to build partial serialization graphs. Utilizing the graphs, mobile clients can autonomously verify serializability of locally executed read-only transactions. This information also allows mobile clients to detect the update transactions that need to be aborted in their early stages. Since the protocol assumes the asynchronous broadcast, mobile clients have the problem that they must listen to the downstream communication channel continuously.

Lee et al. proposed a concurrency control method using a dynamic adjustment of timestamp ordering to reduce the number of restarting transactions (Lee et al., 2002). As their method detects the data conflicts at an early stage, it prevents the wasteful resource consumption. However, their method requires read-only transactions to communicate with a server for their validation using uplink communication bandwidth. Since most mobile transactions are read-only in wireless broadcasting environments, their method results in the serious uplink bandwidth and battery resource consumptions. To overcome this problem, Lee et al. proposed a variant of the OCC method called FBOCC(Forward and Backward OCC) that uses forward validation for update transactions and partial backward validations for read-only transactions (Lee et al., 2004). FBOCC method allows the local validation and commitment of read-only mobile transactions.

3. Basic idea of AOCCRB

In this section, we describe the basic idea of our efficient concurrency control method AOCCRB (Adaptive Optimistic Concurrency Control with Random Back-off) for transaction processing