Chapter 7.31
Energy-Efficient Cache Invalidation in Wireless Mobile Environment

R. C. Joshi
Indian Institute of Technology Roorkee, India

Manoj Misra
Indian Institute of Technology Roorkee, India

Narottam Chand
Indian Institute of Technology Roorkee, India

ABSTRACT

Caching at the mobile client is a potential technique that can reduce the number of uplink requests, lighten the server load, shorten the query latency and increase the data availability. A cache invalidation strategy ensures that any data item cached at a mobile client has same value as on the origin server. Traditional cache invalidation strategies make use of periodic broadcasting of invalidation reports (IRs) by the server. The IR approach suffers from long query latency, larger tuning time and poor utilization of bandwidth. Using updated invalidation report (UIR) method that replaces a small fraction of the recent updates, the query latency can be reduced. To improve upon the IR and UIR based strategies, this chapter presents a synchronous stateful cache maintenance technique called Update Report (UR). The proposed strategy outperforms the IR and UIR strategies by reducing the query latency, minimizing the disconnection overheads, optimizing the use of wireless channel and conserving the client energy.

INTRODUCTION

The tremendous growth in mobile hardware technology and wireless communication has increased the number of clients that access data remotely. Efficient data access in mobile computing is a

field of increasing importance for a wide range of mobile applications. Users of mobile devices wish to access dynamic data, such as stock quotes, news items, current traffic conditions, weather reports, e-mail, and video clips via wireless networks. However, limited battery power of mobile client and scarce wireless bandwidth hinder the full realization of ubiquitous data access in mobile computing. Caching at the mobile client can relieve bandwidth constraints imposed on wireless mobile computing. Copies of remote data can be kept in the local memory of the mobile client to substantially reduce user requests for retrieval of data from the origin server. This not only reduces the uplink and downlink bandwidth consumption, but also the average query latency. Caching frequently accessed data by a mobile client can also save its power used to retrieve the repeatedly requested data.

Cache invalidation strategy is used to ensure that the data items cached at a mobile client are consistent with those stored on the server. Depending on whether or not the server maintains the state of the mobile client's cache, the invalidation strategies are divided into two categories: the *stateful* server approach and the *stateless* server approach (Barbara & Imielinski, 1994; Tan, Cai, & Ooi, 2001). Barbara and Imielinski (1994) provide a solution where the server periodically broadcasts an invalidation report (IR) in which the changed data items are indicated. Rather than querying the server directly regarding the validation of cached copies, the clients can listen to these IRs over the wireless channel and use them to validate their local cache. The IR-based invalidation may be of two types: *synchronous* and *asynchronous*. In the synchronous method, the invalidation reports are broadcast periodically, whereas in the asynchronous method, the server broadcasts the reports only when some data changes. Because of the nature of periodic broadcast, synchronous methods provide a bound on the waiting time of the next report, whereas in an asynchronous invalidation report, there is no guarantee on how long the client must wait.

Clients use IRs to keep their cache consistent by discarding any obsolete data. If a query cannot be served locally—that is, a cache miss—the client issues an uplink query request for the data items. The IR-based solution is attractive because of its scalability, as the size of IR is independent of the number of clients. It is also energy efficient, as clients can exploit the periodicity of server broadcast to save power, in that mobile devices can operate in doze mode most of the time and only activate during broadcast. However, the solution suffers from the problem of long query latency since a client must listen to the next IR before answering a query. The problem has been tackled with the addition of *updated invalidation report (UIR)* by broadcasting a number of smaller reports (UIRs) between successive IRs (Cao, 2001, 2002a, 2002b, 2003). Each UIR contains information about most recently updated data items since the last IR. In case of cache hit, there is no need to wait for the next IR and hence the query latency is reduced. However, if there is a cache miss, the client still needs to wait for the data to be delivered. Thus, due to cache miss, the UIR strategy has the same query latency as IR strategy.

In IR strategy, if the disconnection time of a client is longer than a fixed period, the client should discard its entire cache even if some of the cached data may still be valid. This issue is addressed in Cao (2002a, 2002b), and Jing, Elmagarmid, Helal, and Alonso (1997). Chand, Joshi, and Misra (2005) have demonstrated more efficient handling of arbitrarily long client disconnection.

To overcome the limitations of existing cache invalidation strategies, we present a synchronous stateful caching strategy where cache consistency is maintained by periodically broadcasting *update reports (URs)* and *request reports (RRs).* The central design of our strategy includes reducing the query latency, improving the cache hit ratio, minimizing the client disconnection overheads, utilizing the wireless channel better, and conserving the client energy. The track of cached