Rule-Based Data Mining Cache Replacement Strategy

Ramzi A. Haraty, Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon

Joe Zeitouny, Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon

ABSTRACT

Caching in mobile environments is used for improving bandwidth usage and reducing query delays among resource constrained devices. It is used in multitudes of approaches. Some rely on the server occasionally broadcasting reports to inform clients of updated data while others permit clients to access data whenever they need it. Caching can be constrained by size and bandwidth; thus, coming up with a good invalidation scheme is critical in improving cache consistency solutions. In this work, the authors propose a new hybrid cache invalidation scheme that makes use of data mining concepts. They also compare their scheme with other related strategies.

Keywords: *Cache Management, Cache Replacement Strategy, Data Mining, Invalidation Reports, Mobile Environments*

INTRODUCTION

Caching in a mobile environment can leverage the relatively low bandwidth wireless channel constraint imposed by wireless and mobile computing. However, developing caching techniques has been a challenging task due to several factors such as the need to consider users' mobility and client disconnected modes, which may show as a result of limited power availability or unpredictable disconnection circumstances. Thus, it is not enough to have a copy of the requested data reside in cache because data may have been changed over time and may not be valid anymore. Caching aims at providing users with the latest version of the data when it is requested.

Two primary models for data dissemination have been proposed: the push-based model and the pull-based model (Waluyo et al., 2005a, 2005b). In the first model, clients query for some data objects from the server which replies back with the requested information. In the second model, information is sent from the server to connected clients even though the clients have not requested it. The timing and the data chosen for broadcast is up to the server to decide on.

DOI: 10.4018/jdwm.2013010103

Data dissemination through broadcasting turned out to be efficient (Cao, 2002; Minh & Bich, 2010; Waluyo et al., 2003, 2004; Ilarri et al., 2012; Nicopolitidis et al., 2011; Lin et al., 2010). Broadcasting the data changes from the server in a periodic manner (either by sending data or by using invalidation reports), allows the server to share data information and status updates with an arbitrary number of mobile devices residing in the same network while minimizing the bandwidth consumption.

A multitude of methods that tackle data dissemination were suggested to ensure consistency among cache in mobile computing environments. Several of these methods consisted of stateless servers (Cao, 2003; Chand et al., 2009); whereas others used stateful servers in their design (Kahol et al., 2001; Chavan & Sane, 2011). Stateless servers do not store information about data objects that are stored in each mobile device; rather, they periodically transmit data discrepancy reports. Stateful servers maintain every mobile unit's status and only transmit discrepancy reports in case data objects were updated or accessed. Combining the benefits of both approaches, a hybrid scheme was proposed – SACCS. SACCS or Scalable Asynchronous Cache Consistency Scheme uses as a cache invalidation scheme for a Least-Recently-Used (LRU) algorithm which takes into consideration the least recently used data object when replacing an old entry in the cache (Wang et al., 2004).

In this paper, we examine a multitude of cache invalidation schemes that were adapted to mobile environments. We also propose a new caching policy using data mining concepts, which we call apriori, and evaluate it against previous replacement strategies. Our proposal consists of using an association based rule mining algorithm in SACCS–based cache replacement system. The main reason behind choosing this type of mining is because data objects requested over a certain period are, most of the time related. Hence, any substitution of an item in the cache cannot be done without taking into consideration other related items. Thus, any substitution for a data item that is greatly related to a group of data items in the cache can lead to probable cache misses in the subsequent client queries.

RELATED WORK

Applying caching techniques to mobile environments has proven to reduce network traffic (Seifert & Scholl, 1994; Ding & Chen, 2007; Gaddah & Kunz, 2010). Caching works basically by keeping a replica for some data near the client storage space so that data access rate is improved and network traffic consumption is reduced (Goel et al., 2005). Caching in a mobile environment faces different challenges: how data is to be distributed from mobile support stations to clients - taking into consideration energy usage, limited bandwidth and unexpected clients connection interruption, and how to keep the cached data consistent and valid for each client.

Due to the limited network bandwidth constraint in mobile environments, maintaining cache consistency using validation checks is very inefficient. Instead, mobile support stations may send invalidation reports which notify connected clients about changes in the cached data. These invalidation reports transmit only important information to clients. Sometimes a client may miss an invalidation reports (e.g., when a client gets disconnected during an invalidation report broadcast). Many cache invalidation strategies have been tailored to answer these specific needs. These include the use of synchronous algorithms, invalidation reports and stateful servers. In all of these methods the server often broadcasts invalidation reports (which reflect data item changes) within a fixed window frame. Mobile units listen to these broadcast reports in order to update their cache accordingly.

Other work such as Wu et al. (1996) suggested that the mobile unit needs to reply back to server with IDs of all the data objects cached with their corresponding timestamps when they get reconnected following a substantial amount of disconnection, and then after that the server

Related Content