Chapter XXVIII
Caching, Hoarding, and Replication in Client/Server Information Systems with Mobile Clients

Hagen Höpfner
International University in Germany, Germany

INTRODUCTION

Redundant data management is a must in client server information systems with mobile clients. Based on the level of autonomy of mobile devices/ users techniques for handling such data can be divided into caching, hoarding, and replication. These three terms are often used incorrectly in the literature. To our knowledge the exact definition of the terms has never been published in an international book or journal. We fill this gap with this article. We furthermore explain the terms cache replacement, cache invalidation, cache maintenance, automated hoarding, and synchronization of replicated data.

BACKGROUND

Many mobile information systems are client/server information systems with mobile, lightweight, portable clients that access data and information over wireless links. This scenario implies various restrictions that must be considered:

- Mobile clients must be mobile and therefore need to use rechargeable batteries for power supply.
- The availability of wireless links is uncertain and unstable, very often slow, and/or expensive.
- Wireless data transmission is energy intensive and reduces the uptime of mobile clients.

The only possibility to react to these limitations is to store and reuse data on the mobile device. Unfortunately, redundant data management also implies a high inconsistency risk. Updates to local stored data items and updates to server side data must be harmonized.

MAIN THRUST: MANAGING REDUNDANT DATA

The literature describes three major classes of approaches for managing redundant data in a client server manner if clients are mobile: caching, hoarding, and replication. The following three general questions help to understand the differences: Is the data on the mobile device read only or can the user update it? Who decides about what data is stored on the mobile device - the user or the system? Is the decision for storing data on the device dynamic or static? Figure 1 illustrates various important aspects in which the three classes differ from each other:

- The grade of possible data manipulations: *Are updates allowed on the mobile device?*
- The possibility to work offline: *Is all required data guaranteed to be on the mobile device?*
- The potential dynamic of data: *How often does the data, which are on the mobile device, change?*
- The required resources: *Which kind of software is required for a certain technique?*
- The influence on local data: *Is the user able to specify the data that he or she needs on the mobile device?*

In the following we discuss these aspects per technique in more detail.

Caching

The easiest way to provide a mobile user with data locally is to cache data once it has been received. So, data is requested implicitly. Web and WAP browsers, for example, automatically cache WML or HTML pages and pictures. We know caching approaches from network based desktop applications as well as from database based information systems. The research on caching in the context of mobile information system focuses on semantic caching instead of caching pages/blocks or objects. Thus, query results are stored as they are but indexed using the corresponding query (Keller and Basu, 1996, Lee et al., 1999, Ren and Dunham, 1999). Because of this, it is possible to analyze whether new queries can be answered without communicating with the server. To a certain degree, it is also possible to recombine cached data with additional data to answer a query (Godfrey and Gryz, 1999, Höpfner and Sattler, 2003, Ren and Dunham, 2003).

Unfortunately, the algorithms are complex and suffer from strict limitations of the underlying query containment problem (Chandra and Merlin, 1977, Chekuri and Rajaraman, 1997, Klug, 1988, Guo et al., 1996, v. d. Meyden, 1997). The availability of data in the cache cannot be guaranteed as it depends on the used replacement strategy. If the cache is full, these algorithms decide on removing (hopefully) data no longer required. However, that decision is typically based on the time a data item is already in the cache and/or on the number of accesses of this data item in the cache might be incorrect. Semantic caches tend to use semantic correlations between different cached items as replacement criteria (Dar et al., 1996).

Caches are primarily used for reducing costs and delays of data transmissions. However, they do not provide full offline work with the information system. On the other hand caches are easy realizable and do not require additional functionalities such as synchronization or data mining that are

Related Content

HyTM-AP Hybrid Transactional Memory Scheme Using Abort Prediction and Adaptive Retry Policy for Multi-Core In-Memory Databases
Hyeong-Jin Kim, Hyun-Jo Lee, Yong-Ki Kimand Jae-Woo Chang (2022). *Journal of Database Management (pp. 1-22).*
www.irma-international.org/article/hytm-ap-hybrid-transactional-memory-scheme-using-abort-prediction-and-adaptive-retry-policy-for-multi-core-in-memory-databases/299555

Data Clustering for Effective Mapping of Object Models to Relational Models
Narasimha Bollojuand Kranti Toraskar (1997). *Journal of Database Management (pp. 16-24).*
www.irma-international.org/article/data-clustering-effective-mapping-object/51185

Customer Investigation Process at Credit Suisse: Meeting the Rising Demands of Regulators
Daniel Maier, Thomas Muegeliand Andrea Krejza (2009). *Database Technologies: Concepts, Methodologies, Tools, and Applications (pp. 1787-1807).*
www.irma-international.org/chapter/customer-investigation-process-credit-suisse/8005

Multi-Label Classification: An Overview
Grigorios Tsoumakasand Ioannis Katakis (2009). *Database Technologies: Concepts, Methodologies, Tools, and Applications (pp. 309-319).*
www.irma-international.org/chapter/multi-label-classification/7918

Versioning Approach for Database Evolution
Hassina Bounif (2009). *Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends (pp. 91-101).*
www.irma-international.org/chapter/versioning-approach-database-evolution/20692