

Caching Management of Mobile DBMS

Jenq-Foung Yao
Department of Mathematics and Computer Science
Georgia College & State University
Milledgeville, GA 31061
Email: jfyao@mail.gcsu.edu
Phone: (912) 445-1626
Fax: (912) 445-2602

Margaret H. Dunham
Department of Computer Science and Engineering
Southern Methodist University
Dallas, TX 75275
Email: mhd@seas.smu.edu
Phone: (214) 768-3087
Fax: (214) 768-3085

Abstract

Unlike a traditional client-server network, a mobile computing environment has a very limited bandwidth in a wireless link. Thus, one design goal of caching management in a mobile computing environment is to reduce the use of wireless links. This is the primary objective for this research. Quota data and private data mechanisms are used in our design so that an MU user is able to query and update data from the local DBMS¹ without cache coherence problems. The effect of the two mechanisms is to increase the hit ratio. An agent on an MU along with a program on a base station are used to handle the caching management, including prefetching/hoarding, cache use, cache replacement, and cache-miss handling. The simulation results clearly indicate that our approaches are improvements to the previous research.

Keywords: Caching, Mobile Computing, Mobile DBMS, Mobile Unit (MU), Database, Agent, User Profile, Validation Report (VR)

¹ The local DBMS contains cache data on an MU.

1. INTRODUCTION

For the past ten years, personal computer technology has been progressing at an astonishing rate. The size of a PC is becoming smaller, and the capacity of software and hardware functionality is increasing. Simultaneously, the technologies of cellular communications, satellite services and wireless LAN are rapidly expanding. These state-of-art technologies of PC and wireless have brought about a new breed of technology called mobile computing (MC). Several mobile computing examples have been discussed in [7] and [1].

Most people acknowledge that the mobile environment is an expansion of distributed systems. Mobile units (MUs) and the interfacing devices (base stations that may interact with MUs) are added to the existing distributed systems (see Figure 1). This is a client/server network setting. The servers would be on some fixed hosts or base stations, and the clients could be fixed hosts or mobile units. The mobile units are frequently disconnected for some periods because of the expansive wireless connection, bandwidth competition, and limited battery power. To allow users to access resources at all times no matter which mode they are in, many research issues need to be dealt with. The data caching/replication on an MU is one of the important methods that can help to resolve this problem.

2. PREVIOUS WORKS

In the present research, caching management is handled in two different levels - the file system level and the DBMS level. Issues on the file system level have been addressed widely [11] [22] [12] [21] [17]. Some of the approaches on the file system level have developed real systems that have been used daily, such as Coda [22]. These research efforts on the file system level have some shortcomings. The major one is that all of them explicitly exclude a DBMS. In addition, they use the optimistic replication control. This kind of control allows WRITE operation on different partitions (locations). Committing data in a timely fashion is not important in these systems. Data are allowed to have several different versions on the different partitions, and later will be integrated (and committed). In the academic environment of these approaches, users rarely write to the same file at the same time.

Most of the previous works in mobile DBMS [1] [2] [13] [3] [25] concentrated on the time window "w" and the size of Invalidation Report (IR). These researches impacted the wireless link usage to a certain degree. However, these previous approaches assumed read-only operation on the local cache, just one of the possibilities in which to use cache data on an MU. These previous researches also uplinked to the fixed network for the cache-miss data, which also is just one of the cache-miss handling possibilities.

Only a few researchers at the DBMS level have dealt with the update issue on MUs. Chan, Si, and Leong proposed a mobile caching mechanism based on an object-oriented paradigm [5]. Conceptually their approach is based on an idea that is similar to ours. That is, to cache the frequently accessed database items in MUs to improve performance of database queries and availability of database data items for query processing during disconnection. This is a concept called *hot spot* [2]. This concept states that frequently accessed data are likely to be accessed again in the future.

The other research, which dealt with WRITE operations, is in [24]. In this research, virtual resources are pre-allocated on an MU so that the MU has its own potential share of data. The research is based on a trucking distribution system where each truck is pre-assigned an amount of load. When a truck has