Chapter 3

Data Management in Mobile Environments

Back in the early '90s, the authors of [71] were the first to deal with the problem of data management in distributed mobile environments. It provoked a new research direction with several names such as *pervasive computing, ubiquitous computing, nomad computing, anywhere anytime computing.* At that time, the idea of cell phones becoming personal digital assistants, or pocket computers was still largely a fantasy. Since then, mobile devices have been continuously getting smaller, cheaper and more powerful. Moreover, a large wireless infrastructure exists. WLAN is nowadays a mainstream technology.

This chapter deals with issues pertinent to data management in mobile environments and is organized as follows: Section 3.1 explains the limitations of mobile computing, which despite the impressive progress in recent years, still influence the mobile data management. These limitations play a fundamental role on the design of the *FCLOS* architecture. In Section 3.2, we describe the basics of wireless broadcast systems. Finally, Section 3.3 deals with the research issues of mobile databases. It illustrates the effect of mobility exclusively on relational database systems.

3.1 Limitations

While on the one hand, simple information retrieval such as web browsing or email by users with mobile devices has become a common everyday activity, on the other hand, more complex applications still have to cope with inherent limitations of devices. Some of these limitations tend to vanish. CPU power is not a major issue anymore. Not only can PDAs already easily perform demanding processing tasks, but in the future this limitation can be expected to be completely eliminated. Similarly, storage is not a great issue anymore, since mobile devices can locally store GBs of data.

Nevertheless, there are some inherent shortcomings of mobile computing that

still exist and, more importantly, are not expected to be drastically confronted with in the near future. These are the following:

- **Energy consumption**: Unless a breakthrough in battery technology occurs, a considerable increase of battery resources cannot be expected any time soon. Therefore, the energy efficiency problem profoundly affects mobile applications [46, 145, 155]. Measurements in [152] show that the network interface represents a significant fraction of the energy consumed by the device. This fact directly influences the design of mobile applications.
- **Wireless bandwidth**: WLANs are capable of providing 56 Mbit/s bandwidth, but this is rather a theoretical value, since real applications usually have to live with a lower value. Wireless bandwidth is constantly increasing, but not at a tremendous rate.
- **Small display size**: The limited screen size of mobile devices is a severe limitation, since a considerable increase is not expected. Data analysis becomes extremely problematic using such devices. Therefore, the end user interface design cannot follow the traditional desktop principles [118, 109].

Limitations are not posed only by the devices. The mobile computing environment contributes to the complexity of the domain too. Mobility obviously adds another dimension to the problem. Mobile users change locations. Apart from the fact that devices change their physical location, the application requirements might change as well, since the application semantics might depend on the device's location. Moreover, permanent connections cannot be guaranteed. Protocols and applications have to be designed in such a way, that disconnections are confronted.

3.2 Wireless Information Broadcast

Data broadcast has added another dimension in the area of mobile computing. Data access from wireless channels is a very useful facility because it allows users to get desired data through many computationally enabled devices. In an ideal scenario, assuming an abundance of wireless channels, servers can push all data users might ever need, and users can pull whatever they require. In reality, wireless channels are always fewer than the number required to satisfy user demands. Thus, the task of data dissemination technology is to develop ways for satisfying data demand with limited wireless resources.

Data broadcast is a 1-to-n process, enabling enhanced scalability. The capacity of data transfer from the server to the mobile client (downstream communication) is significantly larger than that of the mobile client to the server (upstream communication). The effectiveness of data dissemination systems is evaluated by its ability to provide users with required data ubiquitously. Data broadcast can be managed with three different modes: on-demand, push-based and hybrid. Before describing the different operation modes, we present the basics of broadcast structures.

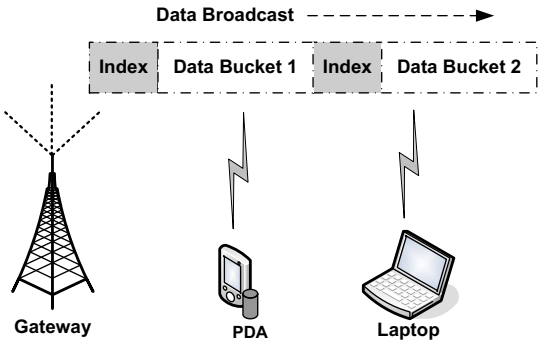

Figure 3.1: Indexed broadcast composition

3.2.1 Broadcast Structure

In broadcast systems, there is no traditional network stack. Data is transmitted in the form of *buckets*. Buckets are also called *data blocks* or *frames*. Practically, buckets reside on top of the wireless Medium Access Control (MAC) protocol. In broadcast systems, mobile clients must wait until the server broadcasts the required information. Therefore, client waiting time is determined by the overall length of broadcast data, which is usually referred to as *broadcast cycle*. Clients must keep listening to the broadcast channel until the arrival of required information. The concept of *selective tuning* is introduced for reducing power consumption. By using selective tuning, mobile clients stay in *doze* mode most of the time and turn into *active* mode only when the requested information is expected to arrive. Indexing techniques are used to implement selective tuning in wireless environments. Indices are broadcast together with data to help mobile clients locate the required information, as shown in Fig. 3.1. In most systems, buckets are classified into *index* and *data buckets*.

Air indexing is fundamentally different than disk indexing. While disk indices represent a path or an offset in the disk (where accessibility is direct), air indices provide solely a time offset indicating when the pointed item is going to appear in the wireless channel. Therefore, established disk air indexing techniques were accordingly adapted. Most of these indexing schemes are based on three techniques: Index tree [74, 37, 35, 102], signature indexing [97] and hashing [73]. Hybrid indexing schemes have been proposed as well. For example, [66] presents indexing schemes by taking advantages of both index tree and signature indexing. Other approaches can be found in [169, 172, 68, 103]. Figure 3.2 depicts an example of distributed indexing. The broadcast data is partitioned into several data segments. The index tree precedes each data segment in the broadcast. Users first traverse the index tree to obtain the time offset of the requested data item. After that, they switch to doze mode until the data item arrives.

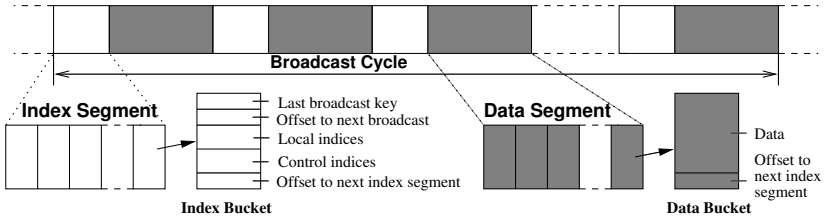

Figure 3.2: Index and data organization of distributed indexing (Source: [169])

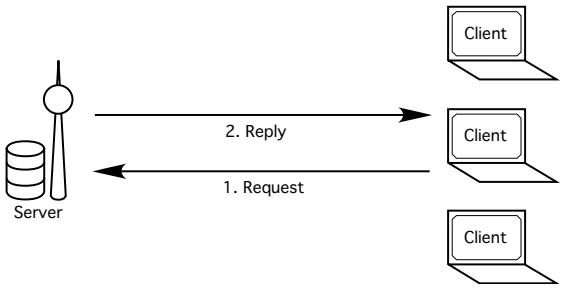

Figure 3.3: On-demand data dissemination

3.2.2 Broadcast Architectures

On-Demand

In on-demand broadcast, clients make explicit requests for data. If multiple clients request the same data at approximately the same time, the server may match these requests and only broadcast the data once. Such an architecture is shown in Fig. 3.3.

On-demand data dissemination is clearly user-oriented. It provides interactive capability to users for accessing the information through query. Users do not have to search in the wireless information space by tuning several channels.

However, this approach has many disadvantages. First of all, it is resource-intensive. Users require a separate channel to send requests to the server. The server, after receiving the request, composes the result and sends it to the user on a back channel (downstream) known to the user. Thus, every pull needs two channels for completing the process.

Moreover, since incoming requests are usually not identical, the server cannot always efficiently group requests in order to exploit the advantages of broadcast. Obviously, this depends on the volume and the context of the incoming workload.

To make things worse, client-server architectures are notoriously not scalable. When the number of incoming requests becomes too high, the server fails to keep up.

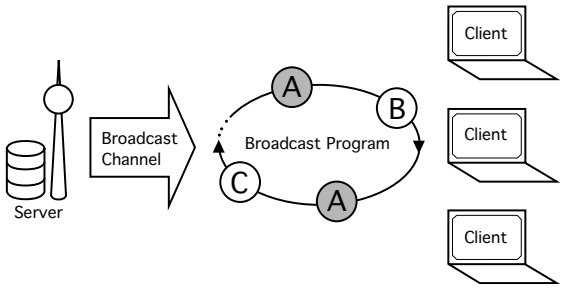

Figure 3.4: Push-based data dissemination

Push-Based

In push-based systems, the server employs point-to-multipoint communication and sends data items in the absence of explicit client requests. In order to achieve that, the server maintains a broadcast schedule, which determines the order and the frequency in which data items are broadcast. Such an architecture is shown in Fig. 3.4. In this trivial example, the scheduler handles three data items (A, B and C), out of which B and C are broadcast with the same frequency and A twice more frequently, resulting in the transmission schedule: $(A, B, A, C, A, B, A, C$ ...).

The major feature of such systems is scalability. Client population does not influence the dissemination process because clients do not issue requests. The addition of new clients does not influence the server's incoming load or the client-perceived access time.

In addition to that, clients need few resources. Mechanisms such as wireless indexing, enable clients to efficiently locate data in the broadcast channel. Moreover, data can be kept actual, since the server can simply broadcast any updates.

The major problem of push-based systems is their lack of self organization and adaptiveness. Since the server does not receive explicit client requests, it remains unaware of possible changes in client population or querying characteristics. This incurs several problems. Bandwidth for instance, can be unnecessarily utilized for a relatively low number of end clients. Apart from that, the push service requires more powerful hardware.

Hybrid data dissemination

As the name suggests, hybrid data dissemination is a combination of on-demand and push-based approaches. Data items are classified into popular and unpopular (respectively, hot and cold). Popular data items are delivered via push-based channels, while unpopular data items are disseminated via on-demand channels. Such an architecture is shown in Fig. 3.5.

Typical design issues in hybrid systems are channel allocation (number of push and on-demand channels), data classification (cold or hot) and item scheduling

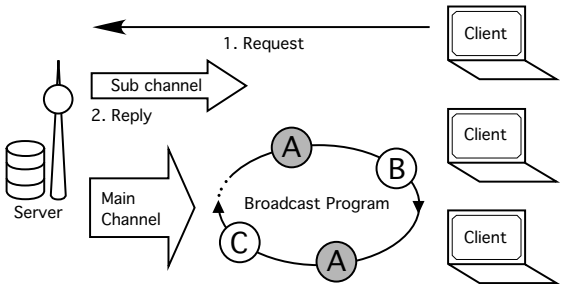

Figure 3.5: Hybrid data dissemination

(both on on-demand and push channel).

As stated in [25], document classification and bandwidth division are interrelated issues, simply because a given bandwidth division determines the performance of a document classification choice and, conversely, a given document classification determines a bandwidth split that optimizes performance. In turn, both document classification and bandwidth division depend on the popularity of data items because download latency is smaller when hot items are assigned to multicast push, cold items to unicast pull, and the bandwidth is divided appropriately between the two channels.

3.3 Mobile Databases

Mobility and portability pose new challenges to mobile database management and distributed computing [72]. In conventional database systems, there is one common characteristic: All components, especially the processing units, are stationary. The first research efforts for mobile databases concentrated on relational databases. The presence of personal and terminal mobility incurs several problems related to the maintenance of the ACID (Atomicity, Consistency, Isolation, Durability) properties. Naturally, the ACID properties of a transaction must be maintained in all data management activities.

Since this thesis deals with OLAP and not transactional data, the following paragraphs provide a brief overview of the area's major research issues. However, some presented issues are similar in the case of OLAP data.

3.3.1 Transaction Processing

Transactions models for mobile environments [106, 123, 34, 62, 122] are different than those used in centralized or distributed databases in the following ways [107]:

- Computation and communication have to be supported by stationary hosts.
- The transactions are prolonged due to the mobility of both data and users, and due to frequent disconnections.

- The models should support and handle concurrency, recovery, disconnection and mutual consistency of the replicated data objects.
- As mobile hosts move from one cell to another, the states of transaction and accessed data objects, and the location information also change.
- Computations might have to be split into sets of operations executed on mobile and stationary hosts.

3.3.2 Query Processing

Query optimization techniques have to consider the effects of mobility. Query processing in mobile environments can be divided into queries that involve only the content of the database, and location based queries. Mobility has several effects on the ACID properties. Location data may involve location based queries or location aware queries. Due to fast changing location data, queries may be answered in an approximate way [108]. Another major issue is querying the broadcast data on the air, as already mentioned in Section 3.2, but under the premise of transactional data. Other typical issues are finding the best execution plan for a query that involves data broadcast on different channels, and defining the organization of the broadcast data so that the consumed energy is minimized.

3.3.3 Caching

Cache management plays an important role in mobile computing because of its potential to alleviate the performance and availability limitations during weak connections and disconnections [23, 166]. It can reduce contention on limited bandwidth networks. This improves query response time and supports disconnected or weakly connected operations. If a mobile user has cached a portion of the shared data, different levels of cache consistency may be requested. In a strongly connected mode, the user may want the current values of the database items belonging to its cache. During weak connections, the user may require weak consistency when the cached copy is a quasi-copy of the database items. Each type of connection may have a different degree of cache consistency associated with it, namely weak connection corresponds to weaker level of consistency. Cache consistency is severely hampered by both disconnections and mobility, since a server may be unaware of the current locations and connection status of clients [128].

3.3.4 Replication

The ability to replicate data objects is essential in mobile computing in order to increase availability and performance [54]. Shared data items have different synchronization constraints depending on their semantics and particular use. These constraints should be enforced on an individual basis. Replicated systems need to provide support for disconnected mode, data divergence, application defined reconciliation procedures, and optimistic concurrency control.

3.4 Summary

This chapter provides an overview of the major issues related to mobile data management in infrastructure based networks. We believe that the current limitations of mobile computing should not be considered only as such. Instead, it should be considered how these are expected to evolve in the (near) future. The limitations not expected to vanish any time soon, should have a prioritized influence on the system design. In this context, we present the fundamentals of broadcast architectures because we believe that such architectures are capable of coping with these limitations. Finally, due to the fact that DWs are specialized databases, we present the major issues concerning mobile data management of relational databases. However, since the focus lies on the maintenance of the ACID properties, they are not so important for the mOLAP domain.