IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITP4767

Energy-Efficient Objects Retrieval on Indexed Broadcast Parallel Channels

Angela Maria Muñoz-Avila and Ali R. Hurson
Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802
{Munoz, hurson}@cse.psu.edu

ABSTRACT

In a mobile computing environment, satisfying a "timely and reliable" access to public data in response to requests generated by a large population of clients is a challenge. This is mainly due to the limitations imposed by mobile units and a wireless medium.

The hardware of the mobile unit offers different operational modes that consume different energy level. Along with the architectural and hardware enhancements, efficient algorithms to allocate data objects and appropriate protocols to retrieve data objects from single and multiple indexed parallel channels can be developed to minimize power consumption and improve access efficiency.

This work is an attempt in this direction. It proposes and examines a scheduling method that minimizes both energy consumption and response time when retrieving objects from indexed broadcast parallel channels in the presence of conflicts. A simulation of the scheme is presented to analyze the relationship between response time and power consumption, and the impact of conflicts among requested objects.

1. INTRODUCTION

The conventional notion of timely and reliable access to global information sources is rapidly changing. This is mainly due to the advances in communication and computation technologies, and sophistication of users. In addition, the rapidly expanding technology is making available a wide breadth of devices with different memory, storage, power, and display requirements.

Multidatabase systems (MDBS) were designed to allow timely and reliable access to large amount of heterogeneous/homogeneous data sources in a distributed environment where resources are connected through a wired network. However, the concept of *mobility*, where a user accesses data through a remote connection with a portable device, has introduced additional complexities and restrictions in a multidatabase system.

An MDBS with such additional physical capabilities was referred to as a mobile data access system (MDAS) [2]. Power consumption has proven to be one of the critical issues in MDAS. This is because of the fact that batteries are the main source of the power in mobile devices and the expected increase in the capacity of batteries is no more than 20% over the next 10 years [1, 6].

Within the scope of the MDAS when user's request is directed to the public data, broadcasting has been suggested as an effective mechanism to access data while overcoming the technological limitations of the mobile access devices and wireless communication. Within this framework, any practical solution should attempt to reduce:

- Access latency, and
- Power consumption at the mobile unit.

In response to the limited power of the mobile computing devices, along with architectural enhancements, efficient algorithms can be developed to reduce both network latency and power consumption. This is achieved through proper allocation/organization of data elements (data objects) on single/parallel air channels, and proper retrieval protocols in an attempt to reduce:

- Number of passes over the air channel (s), and
- Active time of the mobile units [1, 3, 7].

This work examines a method to minimize energy consumption and response time when pulling requested objects from an indexed parallel broadcast channels. The paper is organized as follows. Section 2 overviews the background information for this work, this includes a discussion about wireless services, MDAS environment, data broadcasting, and the previous efforts to allocate and retrieve data to/from parallel channels. Section 3 proposes an object retrieval model that aims to reduce energy consumption and response time in an indexed parallel broadcast channels environment. Simulation of the proposed protocol and analysis of the simulated results are the subjects of Section 4. Finally, the conclusions derived from this work and suggestions for future research directions are discussed in Section 5.

2. BACKGROUND

This work is focused on power consumption constraints when pulling public information from an indexed parallel broadcast channels environment. This section presents an overview of a wireless environment and a computational model – Mobile Data Access System (MDAS) – that supports mobile computing and wireless computation. It also introduces the notion of data broadcasting as an efficient means to make public data accessible. Finally, some previous efforts in allocation and recovery of data from broadcast channel(s) will be discussed.

2.1 Wireless Services

The mobile environment consists of two distinct sets of devices; mobile units and network servers. The network servers are enhanced with wireless transceivers, called mobile support stations (MSS) to communicate with mobile units. The mobile units are located within an area reachable by the MSS, called a cell. The MSS is in charge of providing various data services to the mobile units, such as the maintenance of databases and connection arrangements [3, 6].

2.2 MDAS

The main advantage of multi-database systems lies in their flexibility to allow interoperability among independent, heterogeneous, and autonomous preexisting data sources. Mobile Data Access Systems (MDAS) was proposed as an extension to the traditional multi-database systems to allow access to heterogeneous and autonomous data sources through a wireless medium in an *anytime, anywhere* environment [2].

Within the scope of the MDAS environment, broadcasting has been suggested as a possible mechanism to provide efficient access to public data. Previous works as reported in the literature concentrated on the allocation and retrieval of objects from both single and parallel channels with an eye to minimize access latency and/or energy consumption [1, 3].

2.3 Allocation and Retrieval Mechanisms

Several allocation and retrieval techniques to/from the air channel(s) have been proposed and analyzed in the literature [1, 3, 7]. Some schemes have been suggested within the scope of the single broadcast channel and some directed towards the parallel broadcast channels. These solutions are aimed to minimize response time and/or energy consumption. The literature has paid a special attention in application of indexing techniques in order to organize data on the broadcast channel(s). Indexing facilitates object retrieval from the air channel(s) while reducing the power consumption. The application of an index along the air channel, however, incurs space overhead due to the increase in the broadcast length and hence longer response time. To remedy the performance degradation, Boonsiriwattanakul [3] studied object-oriented indexing along parallel air channels. This study showed that allocation of the object on the parallel air channels reduces the broadcast length, access latency, and power consumption.

2.4 Conflicts in parallel air channels

One of the main problems in a parallel channel environment is the possibility of conflicts while pulling objects from the air channels. A conflict occurs when more than one requested object is transmitted at the same point of time on different parallel channels. Since mobile units can only tune into one channel at a time, the retrieval process has to wait for the next broadcast cycle (s) to download the remaining requested objects [4].

A single broadcast can be modeled as an N x M grid, where N is the number of pages per broadcast, and M is the number of channels. In this grid, K objects ($0 < K \leq MN$) are randomly distributed throughout the MN positions of the grid. The mobile unit can only tune into one channel at a time. The mobile host can switch channels, but it takes time to do this. Based on the common page size and the network speed, the time required to switch from one channel to another is equivalent to the time it takes for one page to pass in the broadcast. Thus, it is impossible for the mobile unit to retrieve both the i^{th} page on channel A and $(i+1)^{th}$ page on channel B ($A \neq B$).

The main problem with conflicts is that they affect the access latency and hence the response time and power consumption. In our previous efforts, a set of heuristics was used to estimate the effect of conflict on access latency. Two heuristics namely, *Next Object* and *Row Scan* heuristics were used to analyze the effect of conflicts on object retrieval [4]. It was shown that when the number of requested objects exceeds 45% of the total objects, *Row Scan* heuristic provides a better solution; moreover, it reduces the delay associated with switching between channels.

3. ENERGY-EFFICIENT OBJECT RETRIEVAL SCHEME

3.1 Retrieval of Objects from Indexed Broadcast Parallel Channels

In an indexed broadcast parallel channels an access protocol is composed of three steps:

1) ***Initial probe***: The client tunes into the broadcast channel to determine when the next index is broadcast.
2) ***Search***: The client accesses the index and determines the offset for the requested objects.
3) ***Retrieve***: The client tunes into the channel and downloads all the required data objects.

The retrieval protocol proposed in this paper focuses on the last two steps of the general access protocol (steps 2 and 3). The idea is to produce an ordered access list of requested objects that reduces:

- The number of passes over the air channels, and
- The number of channel switching.

During the search step, the index is accessed to determine the offset and the channel of the requested objects. With this information available, a sequence of access patterns to pull objects from the air channels is generated using the retrieval scheme proposed in subsection 3.2. Finally the retrieval step is performed following the generated access patterns. Figure 1 shows more detailed operational sequence:

Figure 1: Flow diagram of the proposed retrieval scheme.

```
1. Probe the channel and retrieve offset to the next index.
2. Access the next index
3. Do {Search the index for the requested object
          Calculate the offset of the object
          Get the channel on which the object will be broadcast
4. } while there is an unprocessed requested object
5. Generate access patterns for the requested objects (using retrieval scheme)
6. Do {Wait for the next broadcast cycle
      Do {Reach the first object as indicated by the access pattern
          Retrieve the object
          } while there is an un-retrieved object in the access pattern
      } while there are an unprocessed access pattern
```

3.2 Proposed Retrieval Scheme

The retrieval scheme based on a set of heuristics determines the sequence of objects to be retrieved in each broadcast cycle. It attempts to minimize the energy consumption of the mobile unit and the total number of passes required to satisfy the user query. The schedule is determined based on the following three prioritized conditions:

1) Eliminate the number of conflicts.
2) Retrieve the maximum number of objects.
3) Minimize the number of channel switching.

The scheme determines the order of retrieval utilizing a forest – *access forest*. Access forest is a collection of trees (*access trees*), where each *access tree* represents an access pattern during a broadcast cycle.

An *access tree* is composed of two elements; nodes and arcs.

- **Node:** A node represents a requested object. The nodes are labeled to indicate its conflict status. Each access tree in the access forest has a different node as a root – the root is the first accessible requested object on a broadcast cycle. This simply implies that an access forest can have at most n trees where n is the number of the broadcast channels.
- **Arcs**: The arcs of the trees are weighted arcs. A weight denotes whether or not a channel switching is required in order to retrieve the next scheduled object in the access pattern. A branch in a tree represents a possible access pattern of objects during a broadcast cycle with no conflicts. Starting from the root, the total number of branches in the tree represents all possible access patterns during a broadcast cycle.

This scheme allows one to generate all possible non-conflicting weighted access patterns from all channels. The generated access patterns are ranked based on their weights – a weight is set based on the number of channel switching – and then selects the one(s) that allows maximum number of object retrieval with minimum number of channel switching. It should be noted that, the time needed to build and traverse the access forest is a critical factor that must be taken into account to justify the validity of this approach. This issue will be addressed later as part of our simulation.

3.3 Timing Analysis

Response time, active time, number of pages accessed, number of channel switches, and power consumption as the performance metrics to evaluate the effectiveness of our proposal. The channel switching process requires special consideration in the timing analysis – during channel switching the mobile unit cannot retrieve data from the air channel. This brings up the *overlapped page range* issue. An overlapped page of N means that an object is in conflict with any other object on different channels in the same page or the next $N-1$ pages. In this study, the overlapped page range is 2 [3].

The power is the amount of energy consumed per unit time; as a result the energy consumption is calculated as:

*Energy Consumption = (AccessTime - Tune-inTime)*DozeModePower +Tune-inTime *ActiveModePower +TheNumberOfSwitching %10 *ActiveModePower*

4. SIMULATION AND RESULTS

The scope of our simulator [3] was extended to study the response time and energy consumption of the retrieval scheme presented in Section 3.

4.1 Description of Simulator

Similar to our previous work, we employed a hierarchical indexing scheme to organize objects on parallel air channels [3]. The index structure can be transmitted in different manners including:

- A complete index is transmitted at the beginning of each broadcast in the first channel before the data, or
- A dedicated channel is used to exclusively and continuously transmit the index in a cyclic manner.

The simulator models a mobile unit retrieving objects from an indexed parallel broadcast channels utilizing the method proposed in Section 3. A NASDAQ [8] database was used as the source data for the objects in the broadcast.

The simulator utilizes a two-dimensional grid of N x M to represent the broadcast channels, where *N* and *M* represent the number of objects in a channel and the number of parallel air channels, respectively. The generation of the user requests was performed randomly representing a distribution of *K* objects in the broadcast. In various simulations runs, the value of *K* was varied form one to N x M – in a typical user query of public data, *K* is much less than N x M. Finally, to take future technological advances, parameters such as transmission rate and power consumption in different modes of operation were fed to the simulator as variable entities.

The simulator generates the average time spent on each of the three steps of the retrieval process. For each step, the unit switches to active mode to perform the action, and then returns to doze mode. Hence, the simulator calculates the average active time when the mobile unit is in active mode accessing data and the average idle time when the unit is in doze mode. It also estimates the average query response time. Along with the average times reported, the simulator determines the average number of broadcast passes required to retrieve objects, as well as the number of channel switches performed by the mobile unit while retrieving data. Finally, the simulator uses the collected information to determine the energy consumption of the retrieval process.

4.2 Parameters

For various simulation runs 4290 securities from the NASDAQ [8] database was used as our test-bed. As in our previous research, the hierarchical indexing scheme based on inheritance relationships was used. In inheritance relationships the lower level of the hierarchy inherits attributes and methods from the upper level. The size of the index structure is dependent on the number of distinct keys for objects within a class. For these simulations in particular, the size of the index was 13.52% of the size of the data objects (not including the index). The number of channels varied from 1 to 16 (2 to 17 when an independent channel was used for transmitting the index) to measure the relationship between the number of channels, switching frequency among channels, conflicts, power consumption, and response time. These parameters are summarized in Table 1.

4.3 Simulation Results

For each simulation run, a request of *K* objects was randomly generated. A set of input parameters including the number of parallel air channels, the broadcast transmission rate, and the power consumption in different modes of operation was passed to the simulator. The simulator was run 1000 times and the average of the designated performance metric – the total number of pages accessed, the query response time, the mobile unit active and idle times, the energy consumption, the number passes over the broadcast channels, and the number of channel switches – was calculated.

Table 1: Input Parameters

Parameter	Value (Default/Range)
Number of Objects on Broadcast	4290
Number of Channels	1-16
Size of Air-Channel Page	512 Bytes
Broadcast Data Rate	1 Mbit/sec
Power Consumption (Active Mode)	130 mW
Power Consumption (Doze Mode)	6.6 mW
Power Consumption (Switching channels)	13 mW

The results of the simulations where an indexing scheme was employed were compared against a broadcast without any indexing mechanism. Two indexing scenarios were simulated:

- **Case 1**: The index was transmitted with the data in the first channel (index with data broadcast) and,
- **Case 2**: The index was transmitted over a dedicated channel in a cyclic manner.

Finally, the results were generated for different percentages of requested objects for completeness of the research. However, it should be noted that a simple user query unlikely request more than 1% of objects in the broadcast.

To evaluate the effectiveness of the algorithm proposed in Section 3, a comparison between the proposed algorithm against the *Row Scan* algorithm [4] utilizing indexing on parallel air channels was performed. The index transmission was performed in a cyclic manner on an independent channel and the number of objects retrieved was varied between 5 and 50, which is a reasonable range of objects requested by a query. Finally, the number of parallel air channels utilized was varied among 2, 4, 8 and 16. The simulation results showed that, regardless of the number of parallel air channels, the proposed algorithm reduces both the number of passes and the response time compared to the *Row Scan* algorithm. Moreover, the energy consumption was also reduced, but only when the number of objects retrieved was approximately 15 or less (Table 2).

The simulation results showed that in the worst case, the overhead of the proposed algorithm was slightly less than the time required to transmit one data page.

4.3.1 Response Time

Figure 2 depicts the response times in terms of the number of requested objects, ranging between 5 and 50, and the number of broadcast channels. As noted before three cases were examined. From Figures 2(a) and 2(b) one can conclude that regardless of the indexing scheme employed, the response time decreases as the number of channels increases. However, our simulation results showed that after a certain point the response time increases as the number of channels increases. This is due to an increase in the number of conflicts and hence an increase in the number of passes over the broadcast channels. Figure 2(c) shows that the response time remains relatively constant regardless of the number of channels used. In this organization, the user must scan the same amount of data regardless of the user query and the number of parallel channels. It should be noted that the efficiency of

Table 2: Improvement of Proposed Algorithm vs. Row Scan (10 objects)

Channels	Passes	Response time	Energy
2	48.0%	28.0%	2.7%
4	68.0%	43.6%	3.1%
8	72.3%	46.5%	3.3%
16	71.8%	40.8%	3.4%

the proposed protocol varies based on the number of parallel channels employed. In terms of response time, Case 2 has a better response time compared to Case 3 as the number of channels increases.

In general, employment of indexing scheme reduces the response time when retrieving a relatively small number of objects (smaller than 5% of total broadcast data). When the percentage of objects requested increases, the number of conflicts increases as well [4]. The retrieval scheme proposed in Section 3 tries primarily to reduce the conflicts in each pass of the broadcast; however, when the number of potential conflicts increases considerably, some conflicts become unavoidable causing an increase in the number of passes and hence an increase in the response time. However, our simulation results also showed that as the percentage of requested objects closes to 100%, the response time reduces. This proves the validity of the proposed scheduling algorithm since it generates the same retrieval sequence as the *Row Scan* method would [4].

4.2.2 Switching Frequency

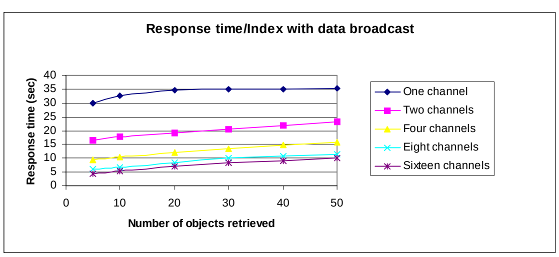
Figure 3 shows the switching frequency for Case 1 and Case 2 since switching pattern is not affected by the indexing policy employed. From this figure one can conclude that the switching frequency increases as the number of channels and number of objects increases. This can be explained by an increase in the number of conflicts; as the proposed method tries to reduce the number of conflicts, the switching frequency will increase. One can notice that when the percentage of objects requested exceeds 50%, the switching frequency begins to decrease. This is due to the fact that the proposed method does not attempt to switch channels as often to avoid the conflicts as the number of conflicts increases substantially. When no indexing technique is utilized, the *Row Scan* method is employed, producing a constant switching frequency independent of the number of objects requested. The switching frequency is at most equal to the number the total channels employed in the simulation.

4.2.3 Energy Consumption

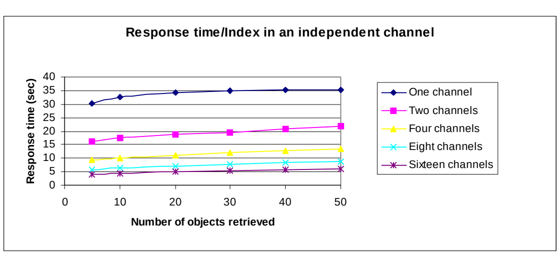
Figure 4 presents the energy consumption for Cases 1, 2, and 3, respectively. From figures 4(a) and 4(b) we can conclude that the energy consumption increases as the number of channels increases. This is because when the number of channels increases the number of conflicts

Figure 2: Response Time.

a) Case 1



b) Case 2



c) Case 3

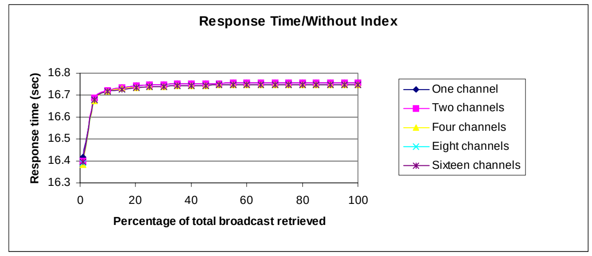
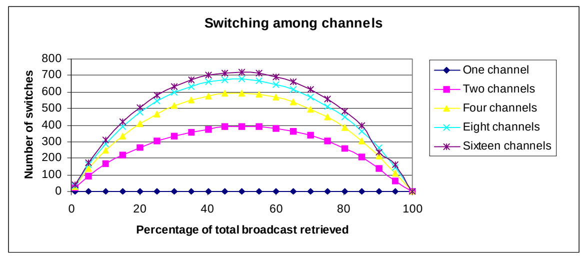


Figure 3: Switching Frequency (Case 1 and Case 2).



increases [4] and so does the number of channel switches. In addition, as can be noted, the energy consumption increases when up to 50% of the broadcast objects are requested, and then it decreases as the number of requested objects increases. It can also be concluded that the energy consumption for Case 1 and Case 2 is almost the same; however, Case 1 consumes more energy than Case 2 in doze mode. In Case 1, in many instances, the mobile unit must wait in doze mode while index is retransmitted. On the other hand, when no indexing technique is used (Figure 4(c)), the energy consumption varies only minimally, due to the nature of the *Row Scan* algorithm employed.

From these figures we can observe that both Case 1 and Case 2 consume less power than Case 3 when a small percentage of objects is retrieved (around 1%). When the percentage of objects requested increases, the number of conflicts and consequently the switching frequency increases as well [4]. This increase in switching frequency results in an increase in the energy consumption.

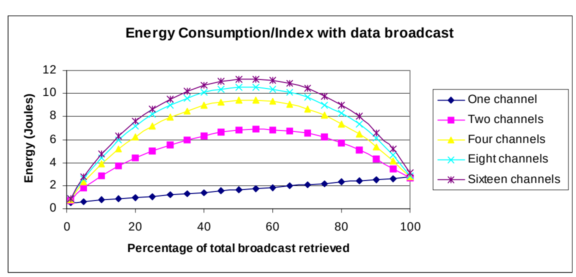
5. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we examined the problem of retrieving objects from a parallel broadcast channel environment. We proposed a retrieval scheme that utilizes indexing for minimizing the energy consumption while retrieving objects.

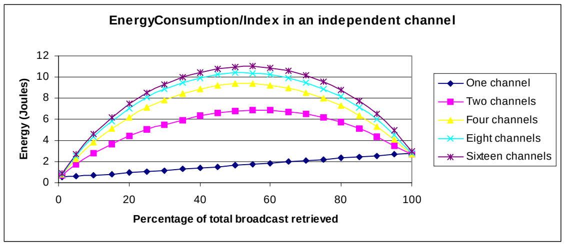
We found that the proposed scheme reduces the response time and the number of passes compared with the case in which *Row Scan* with

Figure 4: Energy consumption.

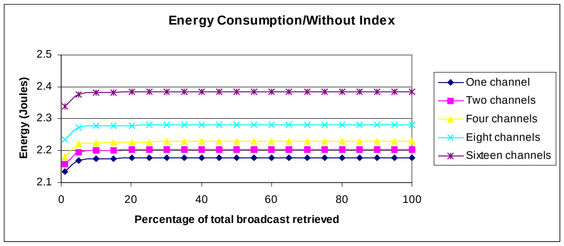
a) Case 1



b) Case 2



c) Case 3



index is used. Moreover the energy consumption was also reduced when reasonable number of objects was requested. The results also showed that the response time of a query decreases as the number of channels increases. Moreover, the results revealed that the index allocation in the broadcast channels has an effect in terms of both response time and energy consumption. The simulation results showed that allocation of index in a separate channel offers lower energy consumption and response time than the scheme where index is intermixed with the data objects.

The retrieval scheme proposed reduces the number of passes over the broadcast channels as the number of channels increases; however, this produced an increase in the number of conflicts and consequently, an increase in the switching frequency among channels. This results in an increase in the energy consumption. We concluded from our study that when a reasonable percentage of objects is requested, the proposed scheduling scheme significantly reduces the energy consumption and response time.

One future research could develop a new method that combines the strengths of the tree based indexing with the signature based indexing schemes.

ACKNOWLEDGMENT

The Office of the Naval Support under the contract N00014-02-1-0282 in part has supported this work.

REFERENCES

[1] Y. C. Chehadeh, A. R. Hurson, L. L. Miller, "Energy-Efficient Indexing on a Broadcast Channel in a Mobile Database Access System", *In The International Conference on Information Technology: Coding and Computing (ITCC'00)*, Las Vegas, Nevada. March 2000.

[2] J. B. Lim, A. R. Hurson, "Heterogenous Data Access in a Mobile Enviroment- Issues and Solutions", *Advances in Computers*, Vol. 48, pp.119-178.

[3] S. Boonsiriwattanakul, A. R. Hurson, N. Vijaykrishnan, Y. C. Chehadeh. "Energy-Efficient Indexing on Parallel Air Channel in a Mobile Database Access System", *World Multiconference on Systemics, Cybernetics and Informatics: Communication Systems, Internet and Mobile/Wireless Computing (SCI'99/ ISAS'99)*, Vol 4, pp. 30-38. Orlando, Florida.

[4] J. Juran, A. R. Hurson, N. Vijaykrishnan, S. Boonsiriwattanakul. "Data Organization and Retrieval on Parallel Air Channels", *High Performance Computing HiPC 2000*. Bangalore, India, December 2000.

[5] W. Stallings. "Data and Computer Communications", *6th Edition*, Macmillan Publishing Company, Englewood Cliffs, NJ, 2000.

[6] T. Imielinski, S. Viswanathan, and B. R. Badrinath, "Data on Air: Organization and Access," *IEEE Transactions on Computers*, Vol. 9, No. 3, May/June 1997, pp. 353-372.

[7] Q. L. Hu, and D. L. Lee, "Power Conservative Multi-Attribute Queries on data Broadcast", *In Proceedings of IEEE International Conference on Data Engineering (ICDE 2000)*, San Diego. Feb. 2000, pp. 157-166.

[8] NASDAQ World Wide Web Home Page, *http:\\www.nasdaq.com*, February 2002.