Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 2 (2024)

Study of Air Ordering Procedure for Information Broadcasting

[1]Ashutosh Sharma, [2]Nikita Joshi,[3]Archana Gupta,[4]Naveen Kumar,[5]Anita Rani,[6]Mridula Yadav

[1]Associate Professor

(Institute of Management Studies, Ghaziabad university Courses Campus)

[2]Assistant Professor

(Institute of Management Studies, Ghaziabad university Courses Campus)

[3]Assistant Professor

(Institute of Management Studies, Ghaziabad university Courses Campus)

[4]Assistant Professor (IPEM College, Ghaziabad)

[5]Assistant Professor (IPEM College, Ghaziabad)

[6]Assistant Professor (IPEM College, Ghaziabad)

Abstract

Wireless data broadcasting is a new way to share public information with many mobile users. To measure how well these systems work, we look at two main things: how long it takes to access the information (access latency) and how long the device has to stay active (tuning time). Indexing technology helps reduce tuning time by letting devices find information quickly and then switch to a low-power mode while waiting. Different indexing methods work better or worse, so comparing them can be hard. In this paper, I look at and compare some popular indexing methods for wireless data broadcasting, such as distributed index, exponential index, hash table, and Huffman tree index. I use probability theory to analyze how each method performs.

Introduction

Wireless data broadcasting has grown in popularity recently because it efficiently disseminates public information to a large number of mobile users with similar interests. This method can handle multiple requests for the same data through a single broadcast. Typically, a base station broadcasts a set of data items, called a program, as radio signals within a specific area at regular intervals. Users in this area can tune into the broadcast channel, locate the desired data, wait for its transmission, and then download it.

This broadcasting technique is used in various real-world applications, such as providing news, traffic updates, and weather information, especially to tourists and visitors in unfamiliar locations.

Given the limited battery life of most mobile devices, two main criteria are used to evaluate the performance of a data broadcasting system: access latency and tuning time. Access latency is the total time from initiating a query to completing the data download. Tuning time is the total period a device remains active during this process. Since active mode consumes significantly more energy than doze mode, reducing tuning time is essential for energy efficiency. Thus, access latency measures the speed of query responses, while tuning time measures energy efficiency.

Indexing technology plays a crucial role in reducing tuning time. An index is a data structure that indicates the location of data items. In data broadcasting, indices contain time offsets for target data items. Once a device retrieves this offset, it knows when the data will be broadcasted, allowing it to switch to doze mode to save energy and wake up just before the broadcast. Different indexing methods have varying search efficiencies. Including indices with data items can increase the broadcast program size, potentially lengthening access latency. Therefore, researchers strive to balance tuning time and access latency when designing indexing schemes.

Several traditional disk-based indexing techniques have been adapted for data broadcasting, such as distributed index, Huffman tree, spatial index, hash table, exponential index, and signature tree. These methods, developed under different conditions, present challenges in performance comparison. Furthermore, the same indexing method can yield different results in different scenarios. Consequently, a unified evaluation strategy is essential to assess the efficiency of various indices and guide the selection of the most suitable indexing scheme for specific systems.

Yang and Bouguettaya imposed strict constraints on data items by assuming each item has an equal size. However, a model with flexible data sizes is more practical for real-world applications. Given the rapid development of indexing technologies in recent years, it is essential to use the latest designs for accurate comparison. My aim is to compare the performance of various indexing techniques across all possible situations. To summarize, the communication environment varies in three key aspects: broadcast environment, data types, and broadcast scheduling.

Broadcast Environment:

Single-Channel Broadcasting: Data items in a program are broadcasted through a single channel, creating an interleaving structure of data and index packets.

Multi-Channel Broadcasting: A program is broadcasted simultaneously across multiple RF channels. Indexing and data scheduling techniques differ significantly in this environment.

Data Type:

In a broadcast program, data items are combined for dissemination, each identified by a primary key. Data types are described by the size and popularity of each item. Early research assumed data items have the same size and access probability , but this assumption is impractical. Recent index designs consider data items of different sizes and varying access probabilities to more accurately reflect real-world scenarios.

Broadcast Scheduling:

Flat Broadcast: In each broadcasting cycle, every data item is broadcasted once, and the whole program is repeated, ensuring equal repetition of all data items.

Skew Broadcast: More popular data items are broadcasted multiple times within one cycle, allowing clients to download them more quickly. Broadcast scheduling methods also vary between single-channel and multi-channel environments.

This paper primarily evaluates index performance in the basic communication environment of single-channel broadcast, assuming data items of different sizes and access probabilities. The focus is on four widely used indexing techniques: distributed index, exponential index, hash table, and Huffman tree index. To ensure fair evaluation, the latest and most efficient indexing designs are used. An accurate formulation, based on probability theory, is constructed to evaluate each indexing scheme's performance.

The paper is organized as follows: Section 2 reviews recent literature on wireless data broadcasting, including various indexing technologies in different communication environments. Sections 3 and 4 analyze and compare the distributed index, exponential index, hash table, and Huffman tree index. Finally, Section 5 concludes with an overall performance summary of the indexing techniques

2. Literature Review

In wireless data broadcasting, the focus remains on how to design index structures and how to allocate data onto channels. Our concern is to reduce access latency and tuning time that improves efficiency of the system.

2.1 Traditional schemes

The primary focus is on how to schedule the data to decrease access latency. Acharya et al. proposed "broadcast disk", which allocates data with similar access frequencies onto different disks and broadcast data of these disks repeatedly according to their frequencies, in order to cope with non-uniform access distribution. But our problem arises for tuning time that is same as that of the access latency. This causes more poor consumption of mobile devices as the devices still not using the feature of doze mode.

2.2 Indexing schemes

The problem of tuning time is considered here, by using various indexing techniques. Indexing techniques provide information regarding that instant of time during which the data will be available on the medium. Till then this time device may switch to the doze mode that may improve the tuning time and conserve the poor as Ill. Some commonly used indexing techniques are listed below:

2.2.1 Hashing schemes

Hashing-based schemes utilize hash functions and store index information with in data buckets. Imielinski et al. Presented two hashing protocols, i.e., Hashing A and Hashing B. Hashing A calculates h(K) and then follows the Shift value to find data, whereas Hashing B provides a minor modification of the hashing functionto improve performance. Later, Yao et al. Proposed MHash, which considers a two argument hash functionH(k,l) to map each data to a number of slots, thus facilitates skewed access probabilities and reduces access latency. Recently, Zhong et al. gave the scheme and proposed HAMHash. By applying several hash functions to allocate data onto multiple channels, facilitating skewed broadcast according to non-uniform data access frequencies and allowing flexible number of data replications by introducing adjustable parameters. HAMHash achieves almost optimal tuning time and energy efficiency.

2.2.2 Tree-based schemes

Imielinski et al. proposed (1,m) index, which broadcasts the index part m times in front of each piece of the data file. They distributed the index, such that it behaves as a B^+-tree into replicated part and non- replicated part. B^+-tree distributed index (BTD) was extended by many other researchers to satisfy different system requirements. One work proposed an index allocation method named TMBT for multi-channel data broadcasting, which creates a virtual BTD for each data channel and multiplexesthemontheindexchannel.Hsuetal.Modified BTDtodealwithnon-uniform data access frequencies. Gao et al. built a complete multi-channel broadcasting system based on the variation of BTD for data set with non-uniform access probabilities and unequal data sizes. In addition, one paper [13] discussed a signature-based approach for information filtering, where the binary hashing code of each datum (as signature) forms a tree to assist searching, which may not perform under non-uniform access probabilities.

2.2.3 Hybrid Indexing schemes

Huet al. designed a hybrid indexing scheme combining BTD and signature-based index. It merges the features of both and covers up the Iakness of each other. Huffman tree is a skewed index tree which takes into account the data access probabilities, where more popular data have a shorter path from the root of the tree, thus the average tuning time is minimized. The construction of Huffman tree is similar to Huffman code construction, but it has a problem that the clients may fail to find the desired data through traversing that Huffman tree. The other algorithm for constructing skewed Huffman tree has the same problem. There is another kind of Huffman tree, Alphabetic Huffman Tree, which serves as a binary search tree.

2.2.4 Table-based schemes

Imielinski et al. presented the flexible index, which divides data file into a variable number of segments according to one adjustable parameter, and stores indices in the tables within data segments. Another work by Xu et al.gave an idea of exponential index that shares links in different search tables, which allows clients to start searching at an arbitrary index node. However, both methods may not perform Ill in non-uniform access probabilities.

3. Analysis of various Indexing schemes:

3.1 B+ -tree-based distributed index: The B+-tree distributed indexing strategy is designed to accommodate non-uniform data access patterns and unequal data sizes, making it well-suited for real-world applications. This strategy combines the strengths of both the distributed index and B+-tree index .

In the BTD strategy, the B+-tree index is broadcasted on the channel using a depth-first traversal and is truncated at level ll. Nodes from level 1 to level ll form the replicated part, while the remaining nodes constitute the non-replicated part, which can be seen as multiple subtrees rooted at the indices at level l+1l+1. Each index in the replicated part acts as a control index, equipped with a control table that specifies the search ranges for different subtrees.

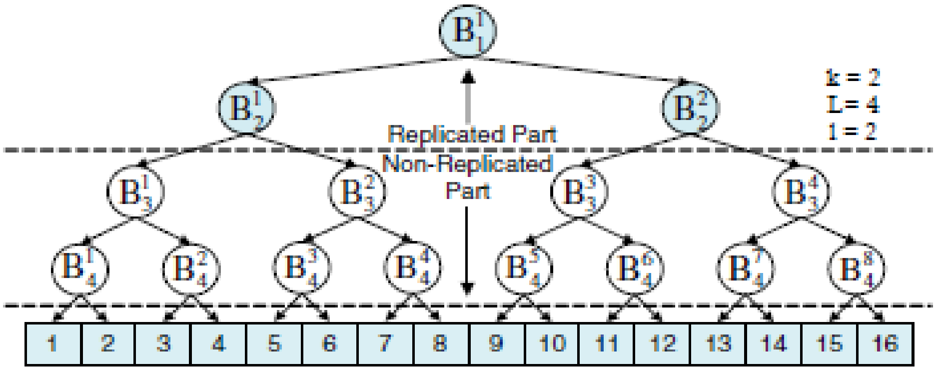

Fig. 1 An example of B$^+$-tree cut at the 2nd level

Figure 1 illustrates an example of a full binary B+-Tree based distributed index structure with k=2k=2, L=4L=4, and l=2l=2. The data set DD contains 16 data items, depicted as grey blocks at the bottom. Each index node $BijBij$ represents the jj-th index node at the ii-th level of the tree. Nodes from the 2nd level and above form the control indices of the replicated part, while the nodes below are the search indices of the non-replicated part. The tree is traversed according to specific rules, interleaving data buckets and index buckets on the same broadcast channel. The complete program BB is streamed on a broadcast channel, encompassing both index and data buckets.

3.2. Exponential Index

The exponential index is highly efficient because it shares links across different search trees, reducing storage overhead. This strategy is based on the generalized exponential index concept introduced in [6]. A key feature of the exponential index is its error resilience, making it well-suited for broadcasting environments with link errors. Its linear and distributed structure allows searches to start from any index and enables quick recovery from link errors.

3.3. Hash Scheme

The hash scheme is a well-known data access method in traditional database systems and is now also used in wireless data broadcasting environments. Hashing-based schemes do not require a separate directory to be broadcasted with the data [13]. Instead, this information is included in the data buckets. Each bucket consists of two parts: the Data part and the Control part. The Control part, used for searching the required data, includes:

-Hash function

-Shift: The pointer to a bucket that contains keys.

3.4. Huffman Tree-based Distributed Index

The Huffman tree index has been employed in wireless broadcast environments for decades due to its efficiency. It considers the access probability of data items when constructing the Huffman tree, placing popular data items with higher probabilities closer to the root. This reduces search time when traversing from the root.

Extending the distributed method to Huffman tree-based broadcast, particularly in the context of flat broadcast, is an innovative idea that hasn't been explored previously. In this section, I will discuss the construction of the Huffman Tree-based Distributed Index Scheme (HTD) and conduct a theoretical analysis of its efficiency. The structure of index and data buckets in HTD is almost identical to that in BTD.

The first step is to construct a k-ary Alphabetic Huffman tree:

Stage 1: Select data nodes didi and djdj as candidates to be merged when:

1. There are no leaves between them,
2. The sum of their frequencies is the minimum,
3. didi and djdj are the leftmost nodes among all candidates.

If these conditions are met, a new index node didi is created with a frequency equal to the sum of didi's and djdj's frequencies, and didi is replaced by djdj in the construction sequence.

Stage 2: Record the level of each data node (leaf node). The root node's level is 1. From bottom to the root, rearrange pointers so that for each level, the leftmost two nodes have the same parent, followed by the next two, and so on

1. Comparative analysis of different indexing schemes:

A comparison among all the indexing schemes i.e. HASH, Huffman, B^+-tree and Exponential are presented below in the form of table, for the following measurement criteria: energy efficiency andtime efficiency, effect of skeId access probability and index bucket size, performance underlink errors, construction complexity and searching complexity, length of bcast, flexibility to tune between AAL and ATT, and clustered or non-clustered features.

In Table1, the performance of these schemes has been categorized into four grades, i.e.excellent, good, fair, and poor, based on the following given parameters. The measurement “Easeof Searching” mainly evaluates the complexity of the searching algorithms in terms of the average time to answer one query in each scheme.

Table 2 The comparison of various indexing schemes

Features	HASH	Huffman	B^+-tree	Exponential
Energy efficiency	Excellent	Good	Fair	Excellent
Time efficiency	Good	Excellent	Good	Fair
Efficient for skewed data	No	Yes	No	No
Better with smaller index	No	Yes	Yes	No
Resilient to link errors	Yes	Sometimes	Sometimes	Yes
Ease of construction	Fair	Poor	Good	Excellent
Ease of searching	Fair	Excellent	Good	Acceptable
Has short Bcast	Good	Excellent	Good	Fair
Flexibility	No	Yes	Yes	Yes
Clustered or non-clustered	Non-clustered	Non-clustered	Clustered	Clustered

From table 2 mentioned above, I obtain the following results in general:

B+-tree:

Easy construction and performs well in searching, particularly with smaller index buckets.

Short access time and flexible, sometimes resilient to link errors.

Suitable for systems with frequently updated data sets and where clients prioritize shorter response times over energy consumption.

Exponential:

Easiest to construct and resilient to link errors, with short broadcast time and low energy consumption.

Suitable for systems with frequent data set updates and frequent link errors, where clients prioritize energy efficiency over fast response times. Clients may tolerate longer waiting times for target data.

Hash:

Consumes minimal energy with short broadcast and access times, also resilient to link errors.

Ideal for systems where clients require both short response times and minimal energy consumption. Works well for systems that do not require frequent updates, although constructing broadcasting sequences and modifying hash functions for different data sets may take time.

Huffman Tree:

Performs well in searching, especially with skewed access probabilities and smaller index buckets.

Flexible, with short broadcast time, best time efficiency, and low energy consumption, sometimes resilient to link errors.

Recommended for systems where clients prioritize minimum response time and low energy consumption, especially with skewed access probabilities in the data set. However, not suitable for systems with frequent data set updates.

5. Conclusion

This paper has undertaken an analysis of various indexing schemes to compare their performance across different environments. Four widely used indexing schemes were selected for evaluation: distributed index, exponential index, hash scheme, and Huffman tree index. The aim was to assess their features, performance, and efficiencies using the same criteria.

In conclusion, considering a given dataset:

The hash scheme emerges as the most energy-efficient method.

The Huffman scheme stands out as the most time-efficient.

B+-tree scheme, while easy to construct, may not perform optimally.

Exponential scheme, though simple to construct, exhibits resilience to link errors.

Based on these findings, service providers can readily select the most suitable indexing scheme to meet the specific requirements of their data broadcasting systems.

References:

[1] Acharya S, Alonso R and FranklinMetal, Broadcast disks: data management for asymmetric communicationenvironments. In: Proceedings of the ACM SIGMOD internationalconference on management of data, San Jose, CA,1995, 24(2), pp. 199–210.

[2] Chen M, Yu P and Wu K, Indexed sequential data broadcasting in wireless mobilecomputing. In:Proceedings of the international conference on parallel and distributed systemsSeoul, Korea, 1997, pp. 124–131.

[3] Gao X, Shi Y, Zhong J, SAMBox: a smart asynchronous Multi-channel black box for wireless data broadcast. In: Proceedings of the 21st international conference on software engineering and data engineering, Los Angeles, CA, 2012, pp. 121-134.

[4] Hsu C, Lee G and Chen A, Index and data allocation on multiple broadcast channels considering data access frequencies, Proceedings of the third international conference on mobile data management,Singapore,2002, pp. 87– 93.

[5] Hu Q, LeeW and Lee D, A hybrid index technique for power efficient data broadcast, Distributed Parallel Databases,2004, 9(2), pp:151–177.

[6] Hu T and Tucker A, Optimal computer search trees and variable-length alphabetic codes, SIAM J ApplMath, 1971, 21(4), pp: 514–532.

[7] Imielinski T, Viswanathan S and Badrinath B, Power efficient filtering of data on air, In: Proceedings of the international conference on extending database technology, Cambridge, UK, 1994, pp. 245–258.

[8] Imielinski T, Viswanathan S and Badrinath BR, Data on air: organization and access, IEEE TransKnowl Data Eng, 1997, 9(3), pp:353–372.

[9] Jung S, Lee B, Pramanik S, Atree-structured index allocationmethod with replication overmultiplebroadcast channels in wireless environment. IEEE Trans Knowl Data Eng, 2005, 17(3), pp:311–325.

[10] Lee WC and Zheng B, A fully distributed spatial index for wireless data broadcast, Aberer K,Franklin M, Nishio S (eds) Proceedings of the international conference on data engineering, Tokyo,Japan,2005, pp. 417–418.

[11] Lee W and Lee D, Using signature techniques for information filtering in wireless and mobile environments, Distrib Parallel Databases,1996, 4(3), pp:205–227.

[12] Lu X, GaoX, YangY and Zhong J, SETMES: a scalable and efficient tree-basedmechanical scheme formulti-channel wireless data broadcast, Proceedings of theACMinternational conference on ubiquitousinformation management and communication, Kota Kinabalu, Malaysia,2013, pp:180-192.

[13] Shivakumar N and Venkatasubramanian S, Efficient indexing for broadcast based wireless systems. JMobile NetwAppl, 1996, 1(4), pp:433–446.

[14] Tsakiridis F, Bozanis P and Katsaros D, Interpolating the air for optimizing wireless data broadcast, In: Proceedings of the ACM international workshop on mobility managementand wireless access, Chania, Crete Island, Greece, October 2007, pp. 112–119.

[15] Wang J, Set-based broadcast scheduling for minimizing the worst access time of multiple dataitems in wireless environments. InfSci, 2012, 19(9), pp:93–108.

[16] Wang S and Chen H,Tmbt: an efficient index allocation method for multi-channel data broadcast, Proceedings of the international conference on advanced information networking and applicationsworkshops,2007, pp. 236-242.

[17] Xu J, Lee WC, Tang X, Gao Q and Li S, An error-resilient and tunable distributed indexing scheme for wirelessdata broadcast. IEEE Trans Knowl Data Eng, 2006, 18(3), pp:392–404.

[18] Yang X andBouguettaya A, Broadcast-based data access in wireless environments, Jensen C, Jeffery K, Pokorny J (eds) Proceedings of the international conference on extending databasetechnology. Czech Republic, Prague,2002, pp. 553–571.

[19] Yao Y, Tang X, Lim E and Sun A,An energy-efficient and access latency optimized indexing schemefor wireless data broadcast. IEEE Trans Knowl Data Eng,2006, 18(8), pp:1111–1124.

[20] Zheng B, Lee WC, Liu P, Lee DL and Ding X, Tuning on-air signatures for balancing performance and confidentiality.IEEE Trans Knowl Data Eng,2009, 21(12), pp:1783–1797.

[21] Zhong J, Wu W, Shi Y and Gao X, Energy-efficient tree-based indexing scheme for efficient retrievalunder mobile wireless data broadcasting environment, Yu J, Kim M, Unland R (eds) Proceedings ofthe 16th international conference on database systems for advanced applications, Hong Kong, China,April 2011, LNCS 6588, pp:335–351.

[22] Zhong J, Wu W, Shi Y and Gao X,Evaluation andcomparison of various indexing schemes in single-channel broadcast communication environment, Springer-Verlag London, 2013, 40, pp: 375-409.

[23] Zhong J, Gao Z, Wu W,Chen W, Gao X and Yue X, High performance energy efficient multi-channel wireless data broadcasting system, IEEE Wireless communications and networking conference, Shanghai, China,2013, pp. 4346- 4351.