Sādhanā Vol. 40, Part 7, October 2015, pp. 2129–2142. © Indian Academy of Sciences

Partial index replicated and distributed scheme for full-text search on wireless broadcast

VIKAS GOEL[1,*], ANIL KUMAR AHLAWAT[2] and M N GUPTA[3]

[1]Department of Computer Science & Engineering, Ajay Kumar Garg Engineering College, Ghaziabad 201009, Uttar Pradesh, India
[2]Department of Computer Application, Krishna Institute of Engineering & Technology, Ghaziabad 201206, Uttar Pradesh, India
[3]Department of Computer Science & Engineering/Information Technology, Amity School of Engineering & Technology, Brijwasan 110061, New Delhi, India
e-mail: rvikasgoel@yahoo.com; a_anil2000@yahoo.com; mngupta@gmail.com

MS received 15 September 2014; revised 23 June 2015; accepted 2 July 2015

Abstract. Information indexing is an extremely useful solution for wireless broadcast channels, because of its energy efficiency. Full text search on the wireless broadcast stream is another popular type of information dissemination access. This research paper discusses indexing schemes for full text search information, using two levels of structure B+ tree and inverted list. The partial replication in the B+ tree indexing scheme is proposed to extend the existing full text search indexing scheme. In the proposed work, the index information of B+ tree has been cut to reduce access time and tuning time: metrics for evaluating indexing schemes. The replication in B+ tree may be done up to a certain level of the indexed informat+ion (Data). Evaluation and analysis of the proposed indexing scheme show an excellent improvement over existing indexing schemes.

Keywords. Energy efficient; full-text search; B^+ tree; wireless communication; inverted list; partial distributed indexing.

1. Introduction

Wireless data broadcast is one of the most efficient solutions for disseminating information to a vast number of mobile clients than asymmetric communication (Imielinski *et al* 1994). A server periodically broadcasts the data on the wireless channel. Hand held mobile devices tune into the channel to retrieve the data without sending any request to the server. These hand held mobile devices have limited energy resource, i.e. battery power. So, energy conservation is an important issue for mobile devices (Barbara 1999). In asymmetric communication, the mobile devices have to tune to the channel consistently in active mode (more energy consumption) to access the

[*]For correspondence

desired information. This is inefficient due to limited energy resource. Selective tuning is one solution to this problem. The mobile devices move into the doze mode (less energy consumption) to sleep and move into the active (high energy consumption) mode to listen the channel only when the desired data is available on the channel. By switching between active and doze modes, the power consumption of a mobile device is reduced significantly (Imielinski *et al* 1994). In order to identify the data frames which would qualify ahead of time, the control information about the content of the frames are added along with the data frames. The organization of the control frames with the data frames is called indexing. In literatures, various indexing techniques have been proposed to reduce the energy consumption of the mobile devices. Basically, these indexing techniques are divided into four categories: Tree based, Hash based, Signature based and Table based (Faloutsos & Christodoulakis 1984; Imielinski *et al* 1994; Zhong *et al* 2013b).

These techniques are further divided into two parts: uniform and non-uniform. The uniform broadcasting technique has been proposed in (Imielinski *et al* 1994, 1997; Seifert & Hung 2006; Xu *et al* 2006; Yang & Bouguettaya 2005) and the non-uniform broadcasting technique in (Seifert & Hung 2006; Yao *et al* 2006). Access time and tune time are the two metrics in all indexing schemes (Imielinski *et al* 1994):

(1) Access time: The time elapsed from the moment a client wants data to the time when the desired data is downloaded.
(2) Tuning time: The amount of time a client listens to the broadcast channel.

In uniform broadcasting, all data are broadcasted only once in each broadcast cycle and has the same average access time i.e. half of the length of a broadcast cycle. In non-uniform broadcasting, the most frequently accessed information is broadcasted more than once in broadcast cycles. Such a nonuniform broadcasting has proved to be superior in terms of average access time compared to a uniform broadcasting.

Today, full text searching is one of the most popular query types used in various information retrieval systems. The access methods for full text searching have been examined in Chung *et al* (2010). However, earlier these methods were developed for disk storage. The efficiency and cost of the access method parameters are calculated in terms of the number of disk accesses. The major difference is that the disk-based indexing allows random access to data, whereas the data must be accessed sequentially on a wireless broadcast channel. This property significantly changes the evaluation metrics for indexing schemes on wireless broadcast channel.

In wireless data broadcast, the research work on full-text query processing through B+ tree & inverted list is published by Chung *et al* (2010). In their work, full text searching has been performed in two steps; first, B+ tree is utilized and then inverted list in the processing of full-text queries. Two methods have been proposed: the inverted-list and the inverted-list + index-tree that have been extended to $(1, \alpha)$ and $(1, \alpha(1, \beta))$ methods respectively (Chung *et al* 2010).

However, Chung *et al* have not considered the partial replication of the index tree. In this paper, we are considering the partial replication in the B+ index tree. The partial replication of the index tree is taken into consideration for distributing the index. Considering the advantages of the partial replication to cut down the length of index information, a partial index replicated and distributed scheme is proposed. The performance of the proposed indexing scheme is evaluated and analyzed. The results demonstrate the effectiveness of the proposed work.

Our paper progresses in the form of the following sections viz., Section 2 describes the background of the indexing techniques. Section 3 describes and discusses the preliminary of the full text search indexing scheme. Section 4 presents a proposed partial index replicated and distributed scheme for full text search on a wireless broadcast. In section 5, mathematical analysis of the proposed scheme with the existing schemes is undertaken. In section 6, performance of the proposed technique is analyzed. In section 7 result analysis is carried out in detail and comparisons with the existing schemes are also shown. Finally, section 8 concludes with the results drawn through the research process.

2. Background

Due to features like scalability, energy efficiency and bandwidth utilization, wireless data broadcast has become very popular among researchers over the past few years. In the wireless broadcast environment, the broadcast cycle consists of data units, called buckets same as the blocks on disks. So, the basic unit for accessing the data on the air is a bucket. The bucket is of two types: data bucket and index bucket. A number of buckets are used to calculate the average access time and the average tuning time (Imielinski *et al* 1994).

First, Imielinski *et al* (1994) gave an impression on air indexing i.e. wireless data broadcast system. They also proposed popular energy efficient air indexing schemes (1, m) indexing and B+ tree distributed indexing through the uniform broadcasting of data. Previously, Acharya *et al* proposed a broadcast disk (BD) to allocate more popular data to appear more often in a broadcast cycle than less popular ones, called non-uniform broadcast (Acharya *et al* 1995). Yao *et al* proposed another distributing indexing technique the exponential index, in which the index is distributed but in an exponential manner. In Yao *et al* (2006), the tradeoff between confidentiality and performance of the signature-based index has been discussed at length. A new approach hash-based index for uniform wireless data broadcast was also proposed by Xu *et al* (2006). However, all these indexing techniques focus only on structured data with predefined key attributes and these may not be applied directly to guide full-text queries with no predefined key attributes.

The inverted list is a famous data structure method for document indexing and retrieval systems which is also used as a technique for full-text search. Tomasic *et al* (1994) studied the incremental updates of inverted lists by dual-structure index. Scholer *et al* (2002) discussed the compression of inverted lists of document postings which contains the position and the frequency of indexed terms and developed two approaches to improve the document retrieval efficiency. Zobel & Moffat (2006) have done a survey on inverted files for text search engines. Mahapatra & Biswas (2011) discussed a number of inverted index technologies. A large number of researches on inverted list are based on disk-storage documents but for on-air documents, modifications are needed in order to adjust with on-air storage features.

Zhong *et al* (2011) worked on the huffman tree. They proposed a distributed indexing scheme based on the Huffman tree. Zhong *et al* (2013a) created a model that achieves high performance in a multichannel environment. They used hash-based indexing scheme in their model. Zhong *et al* (2013b) analyzed and compared the performance of a number of existing air indexing schemes. They created a unified environment for comparing air indexing schemes, namely tree based, exponential based, hash based and signature based.

Yang *et al* (2011) proposed a data streaming scheme (Basic-Hash) with hash-based indexing and inverted list techniques to facilitate energy and latency efficient full-text search in wireless data broadcast. The first research was done by Chung *et al* (2010) that proposed an inverted list index method for full text search. The authors proposed a full text searching algorithm and an index replication distributing method for locating the index structure over wireless channel. The index is intermixed with the data contents. They also combined traditional B+-tree indexing technique with an inverted list for full-text query on the wireless broadcast channel. They have proposed two indexing methods: Inverted-List and Inverted-List + Index-Tree that was further extended to $(1, \alpha)$ and $(1, \alpha(1, \beta))$. In both of these methods, either the index was fully replicated or not replicated. However, they did not consider the partial replication of the index tree. Throughout this research work partial replication of the index tree in the distributed manner is considered.

3. Preliminary

3.1 *The index tree*

Figure 1 depicts the traditional B+ tree to index the data. The index tree consists of index tree buckets. The B+ tree is split into two parts: the upper replicated part and the lower non-replicated part. The traditional B+ tree is used as an index tree except that the leaf node has a pointer to the corresponding inverted list bucket in spite of data by Imielinski *et al* (1994) and Imielinski *et al* (1997).

3.2 *The inverted list*

In the wireless data broadcast system, many researchers have used the inverted list technique to facilitate full-text search (Chung *et al* 2010; Mahapatra & Biswas 2011; Yang *et al* 2011). A single word can be linked to many documents and each document contains a number of words. As a result many-to-many relationship exists between the words and the documents in full-text search. In order, to decide many-to-many relationship between words and documents, the inverted list has been used to index the data in the data retrieval systems (Tomasic *et al* 1994; Zhang & Suel 2007; Zobel & Moffat 2006). An inverted list is constructed in figure 2 with the data set of table 1. For example, there are four documents Doc1, Doc2, Doc3 and Doc4. The documents have the text depicted in table 1. Various words are extracted from these documents by eliminating stop words or by stemming. The words and address of documents containing these words are linked together by an inverted list of documents in figure 2.

Figure 3 depicts the structure of the index list bucket and the data bucket. An index list bucket is assigned to many words and a list of pointers to the documents containing them. If there

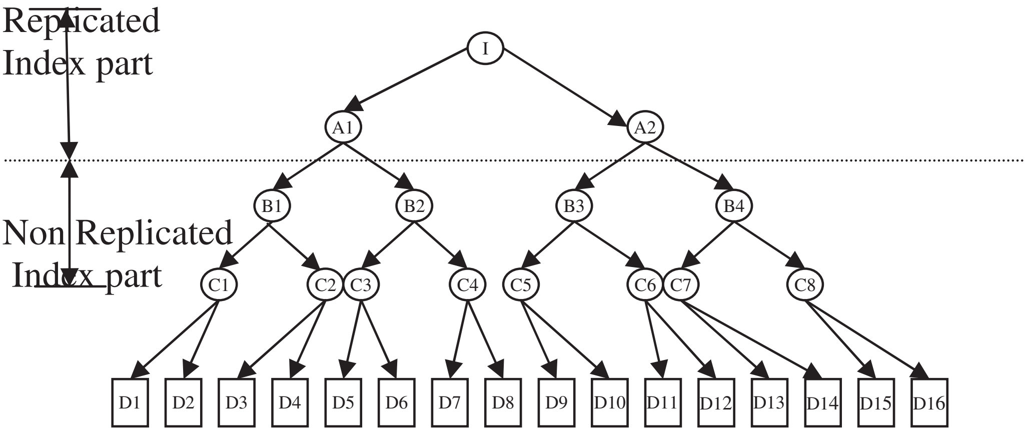
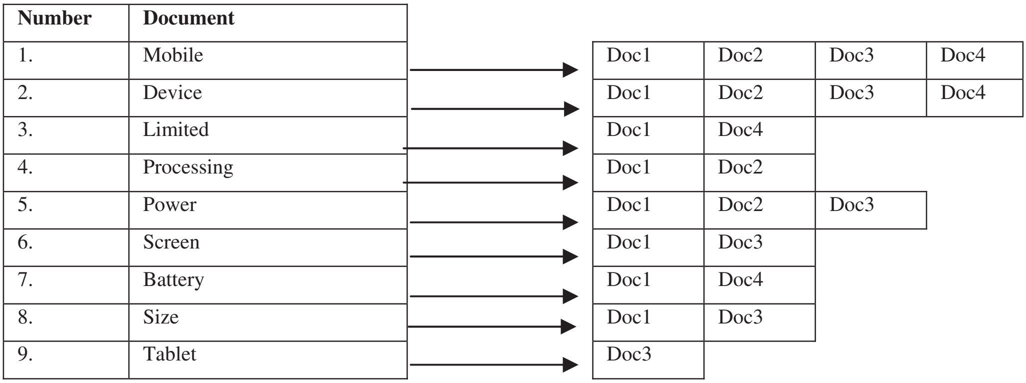


Figure 1. A sample B+ Tree.



Number	**Document**				
1.	Mobile	Doc1	Doc2	Doc3	Doc4
2.	Device	Doc1	Doc2	Doc3	Doc4
3.	Limited	Doc1	Doc4		
4.	Processing	Doc1	Doc2		
5.	Power	Doc1	Doc2	Doc3	
6.	Screen	Doc1	Doc3		
7.	Battery	Doc1	Doc4		
8.	Size	Doc1	Doc3		
9.	Tablet	Doc3			

Figure 2. An inverted list of document.

Table 1. Data record.

Document	Text
Doc1	Mobile devices have limited resources like processing power, battery and screen size
Doc2	Now, Mobile devices have the processing power in GHz
Doc3	Various mobile devices called Tablets have very big screens
Doc4	Rechargeable battery still has limited power for mobile devices

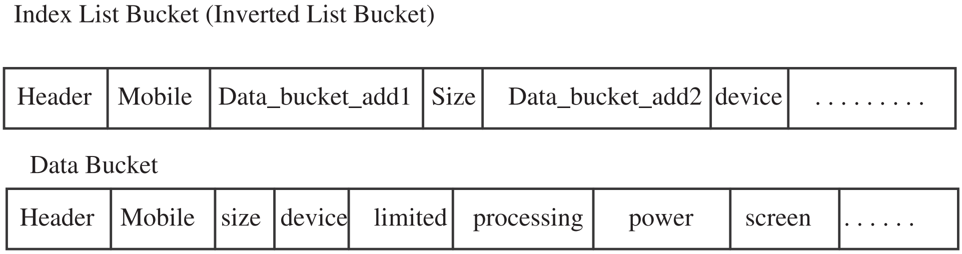


Figure 3. The structure of bucket in inverted list.

is a need to store many pointers in one bucket, then more buckets can be specified. In index distributed and replicated schemes, an inverted list is placed just before the data on the wireless broadcast stream. The server periodically broadcasts the index bucket and the data bucket of this inverted list on the wireless communication channel.

In case of non-replication of an index, a mobile user has to wait half the broadcast cycle if a mobile user accesses the broadcast stream in the middle. This wait increases the access time. To reduce this wait time, index replication may be a solution proposed by Imielinski *et al* (1994). The inverted list index may be replicated α times where α is the number of replications. The method is extended by Chung *et al* (2010) as $(1, \alpha)$ scheme. In this method the inverted list index is replicated α time in a broadcast cycle. The broadcast cycle is also called bcast. The bcast of an inverted list index method is represented by a combination <list index, data>.

The α time replicated inverted list index occupies a large space on the wireless channel. The tunning time for searching the desired index will be more. Chung *et al* have proposed a two-level index structure by adding an index tree structure with an inverted list. In this two-level hierarchy, the bcast is a combination of <tree index, list index, data>. The method is extended by Chung *et al* as $(1, \alpha(1, \beta))$ scheme. In this scheme, the inverted list index is replicated α times and the index tree is replicated β times for each inverted list (Chung *et al* 2010).

4. Proposed partial index replicated and distributed scheme

An improvement may be made on $(1, \alpha)$ and $(1, \alpha(1, \beta))$ indexing schemes by cutting down the index replication. It has been observed that there is no need to replicate the entire index path between successive data segments. It is sufficient to have only a portion of the index, which indexes the data segment that follows it. The index is partially replicated in distributed indexing. An inverted list is used to guide a full-text search and a tree-based index to locate the key word in that inverted list. In $(1, \alpha)$, $(1, \alpha(1, \beta))$ and the proposed distributed indexing scheme, the index is interleaved with data to construct a bcast. Only a relevant index is interleaved with data in the proposed distributed indexing scheme that opposes the interleaving of the whole index in $(1, \alpha)$ and $(1, \alpha(1, \beta))$ indexing. The proposed indexing scheme reduces the overall length of the index in bcast.

Figure 4 depicts the partial replication of index tree with an inverted list in a proposed partial index replicated and distributed scheme. The proposed indexing scheme improves the access time by providing a partial replication of the index tree. The distributed indexing scheme is not a new scheme for index construction, but the proposed indexing scheme uses this method effectively to allocate index and data on the wireless broadcast channel. The index tree is divided into two parts: a replicated part (upper part) and a non-replicated part (lower part). The top

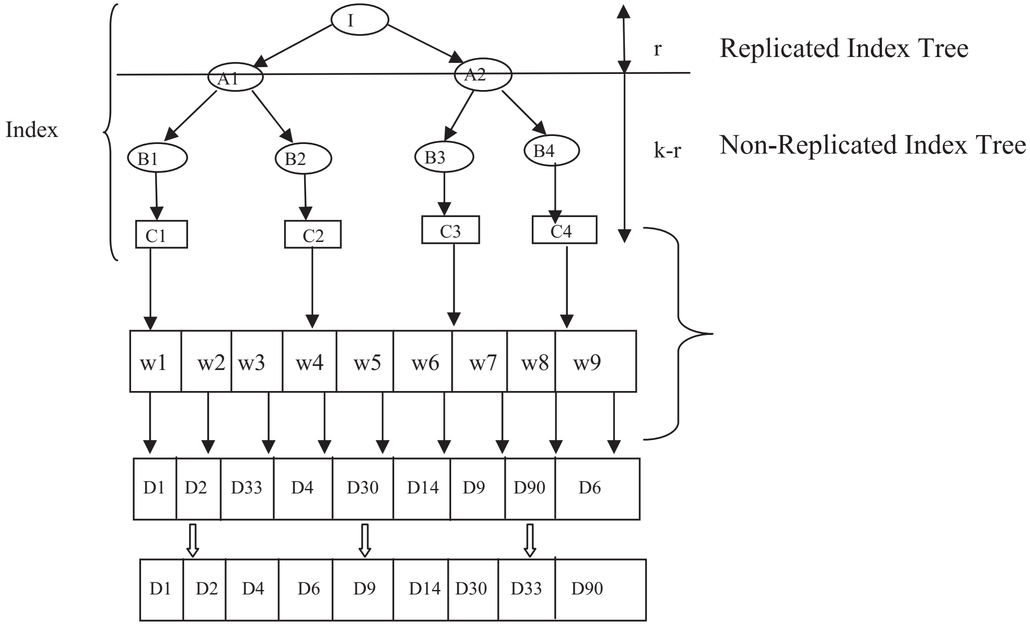


Figure 4. Inverted list + index tree in proposed partial index replicated and distributed scheme.

replicated levels are denoted by “r” and bottom “k-r” levels are denoted by non-replicated part of the index tree. The index buckets of the (r+1)$^{\text{th}}$ level are called non-replicated roots. These roots are denoted as non-replicated roots by ordering left to right collectively. In non-replicated root, each index sub-tree root will appear only once in the whole broadcast. It is placed just in front of the set of data segments pointing to indexes. Hence, each descendant node of a non-replicated root of the index will appear only once in the broadcast.

Above a non-replicated root, each node of the index tree is replicated as many times as the number of children it has. The leaf node of the index tree consists of index tree bucket pointers pointing to the inverted list bucket. The structure of the index tree bucket consists of a header as:

Start_word: when the word size is smallest or
End_word: when the word size is longest
PTR: pointer points to another index tree bucket only if it is an internal node, otherwise points to the inverted list bucket

Non-replicated root is represented by $\{B_1, B_2, \ldots B_t\}$ group of buckets. Each bcast of the broadcast will be a sequence of <Rep (B), Ind (B), List (B), Data (B)> in the order of left to right. Where,

Rep(B): refers to the replicated portion of the path from the beginning to the index segment B.
Ind(B): refers to the non-replicated portion of the index tree.
List(B): refers to the indexed data in the inverted list.
Data(B): refers to the set of data buckets indexed by B.

The inverted list, List(B) is α times replicated in a bcast and the index tree is replicated β times for each inverted list. The index tree is divided into two: replicated parts Rep(B) and Non-replicated parts Ind(B). Therefore, $\alpha\beta$ times Non-replicated part Ind(B) are placed on a single bcast.

5. Mathematical analysis

5.1 *The system model and parameters*

The system model is defined on the basis of parameters defined in table 2. The NumberofDocuments is the total number of documents on the server. Each document size is defined as DocumentSize. Data and index are transferred in the data bucket and the index bucket respectively. The size of the bucket is defined as BucketSize. The number of distinct words in the database is defined as NumberofDistinctWords identified by leaf node of B+ tree. The length of each word is WordSize. The bucket address on the wireless channel is identified by an AddressTupleSize. The height of B+ tree is H and the fanout of B+ tree is N. A word is framed by traversing B+ tree in pre-order at leaf node and linked to the inverted list. All leaf nodes are distinct words. We have assumed that B+ tree should be balanced i.e. words should be of the same length. A number of documents matched with this word are selectivity of word (S). The B+ tree index is replicated α times and the inverted list index is replicated β times. A bcast constitutes index and data. The total size of a broadcast stream is BroadcastSize. The average access time and the average tuning time are the two evaluation parameter. AvgAT is defined as average access time and AvgTT is defined as average tuning time.

Table 2. Parameters.

Parameter	Description
NumberofDocuments	Database on server
DocumentSize	Size of document
BucketSize	Size of bucket
NumberofDistinctWords	The no of distinct words in the database
WordSize	The length of the word
AddressTupleSize	The address of the bucket on the stream
H	The height of the tree
N	Fanout
S	Selectivity of a query word i.e. the number of documents matched to a query word
α, β	Index replication levels
BroadcastSize	The size of the broadcast stream
AvgAT	Average access time
AvgTT	Average tuning time

5.2 *Proposed partial index replicated and distributed scheme*

This section proposes expressions of the proposed distributed indexing scheme. The values of evaluation metrics: average access time, the average tuning time and the optimum value of replicated levels are evaluated. It is assumed that the initial probes by the clients are uniformly distributed over the period of the whole bcast.

Let,

"r" – denotes the number of replicated (top) levels
"n" – denotes the capacity of the bucket and finally
"k" – denotes the total number of levels in the index tree

The proposed scheme considers balanced index trees that have all the leaves at the same level and each node has the same number of children. In reality, the index tree may be radically different (unbalanced, varying fanout). This is the future work of the proposed scheme.

$$\text{NumberofDistinctWords} = (n^k - 1) \tag{1}$$

$$DBSize = \frac{\text{NumberofDocuments} * \text{DocumentSize}}{\text{BucketSize}} \tag{2}$$

$$SizeofIndexList = \frac{(n^k - 1)^*(\text{wordSize} + \text{AddressTupleSize}^*\text{S})}{\text{BucketSize}}. \tag{3}$$

5.2a *Proposed formula*: A partial replication changes the size of the index tree and the length of the bcast. In the proposed indexing scheme, the length of the index tree is split into two: (*r*) replicated part and (*k*–*r*) non-replicated part. The size of the index tree formula will be:

$$\text{SizeofIndexTree} = \frac{(n^{k-r} - 1)}{\text{n} - 1} \tag{4}$$

$$\text{SegmentSize} = \text{Size of IndexTree} + \frac{1}{\beta} * \text{SizeofIndexList} + \frac{1}{\alpha\beta} * \text{DBSize} \tag{5}$$

$$\text{BroadcastSize} = \alpha\beta * \text{SegmentSize} \tag{6}$$

$$\text{Average access time} = \frac{\text{SegmentSize}}{2} + \frac{\text{S} * \text{BroadcastSize}}{S+1}. \tag{7}$$

So, the average access time is

$$\begin{aligned} \text{Avg AT} \;&=\; \left(\frac{1}{2} + \frac{\alpha\beta * S}{S+1}\right) * \text{SizeofIndexTree} \\ &\quad + \left(\frac{1}{2\beta} + \frac{\alpha}{S+1}\right) * \text{SizeofIndexList} + \left(\frac{1}{2\alpha\beta} + \frac{S}{S+1}\right) * \text{DBSize} \end{aligned} \tag{8}$$

Initially, the tuning time depends on the height of the index tree. The first probe is to determine the occurrence of control index. The second probe is for accessing the control index. Control index directs the client to higher levels of index. Next, the control index directs the client to the inverted list index. At last, the required data is downloaded. The average tuning time is

$$\text{AvgTT} = 1 + \text{H} + 1 + \text{S}. \tag{9}$$

The average tuning time is not affected by both α and β but the average access time is affected by both. So, the identification of the optimum values of α and β for the minimum access time is to be undertaken. Values may be represented by following expressions:

$$\alpha = \sqrt{\frac{(S+1) * DBSize}{2 * S * \text{SizeofIndexList}}} \tag{10}$$

$$\beta = \sqrt{\frac{(S+1) * (Sizeof\,Index\text{List+DBSize})}{2 * S * \text{SizeofIndexTree}}}. \tag{11}$$

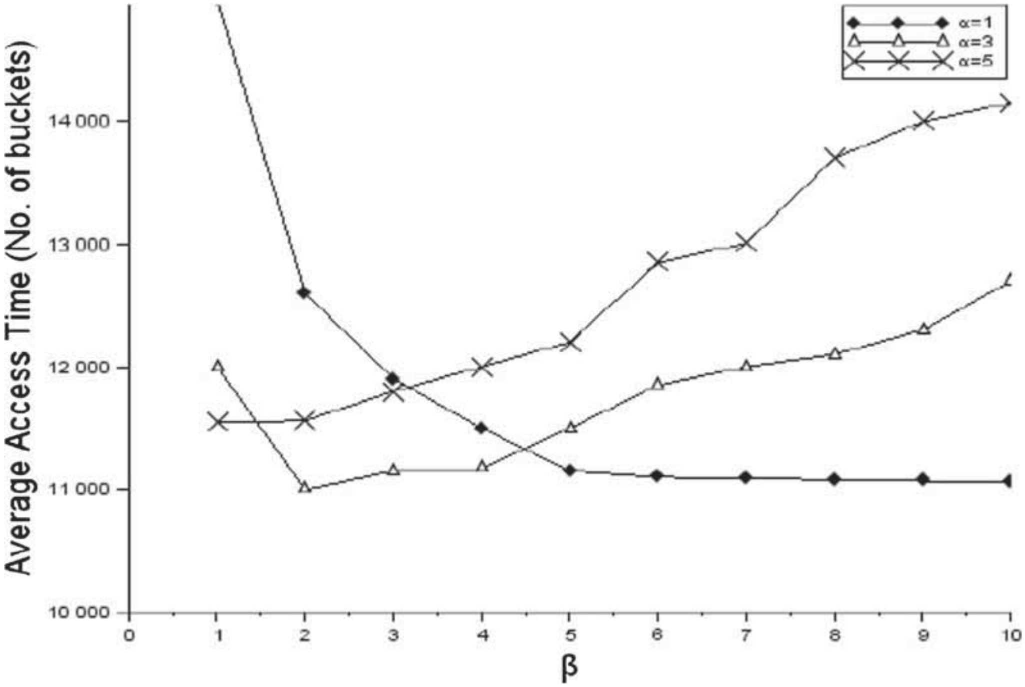
6. Performance

In this section, performance of the proposed indexing scheme is analyzed. All schemes are analyzed and compared on the basis of performance metrics average access time and average tuning time. The data sets and parameter values are kept same as in Chung *et al* (2010) for fair comparisons. For evaluating the indexing schemes, 10,000 number of documents each of size 1,024 are taken. The numbers of distinct word are 4,703 for creating an index tree and an inverted list. The total size of a word is 16 and 4 is the address tuple size. The number of repetitions of a word is kept non-uniform by varying the values from 1 to 5 for a document and the selectivity of the word is also varied accordingly. The size of the bucket is 1,024 and the number of index replication is 2 in the index tree. This means that the index tree is cut at level 2 and divided into two the upper replicated part and the lower non-replicated part.

Parameters and their values are given in table 3. Different indexing schemes: no replication, $(1, \alpha)$ and $(1, \alpha(1, \beta))$ indexing are analyzed and compared. Initially $\alpha = 1$ and $\beta = 1$ is

Table 3. Parameters with values.

Parameters	Value
Number of documents	10,000
Size of each document	1,024
Number of distinct words	4,703
Size of words	16
Address tuple size	4
Number of repetition	1–5
Average selectivity	51
Bucket size	1,024
Number of replication	2



Graph 1. Access time performance of a proposed partial index replicated and distributed scheme.

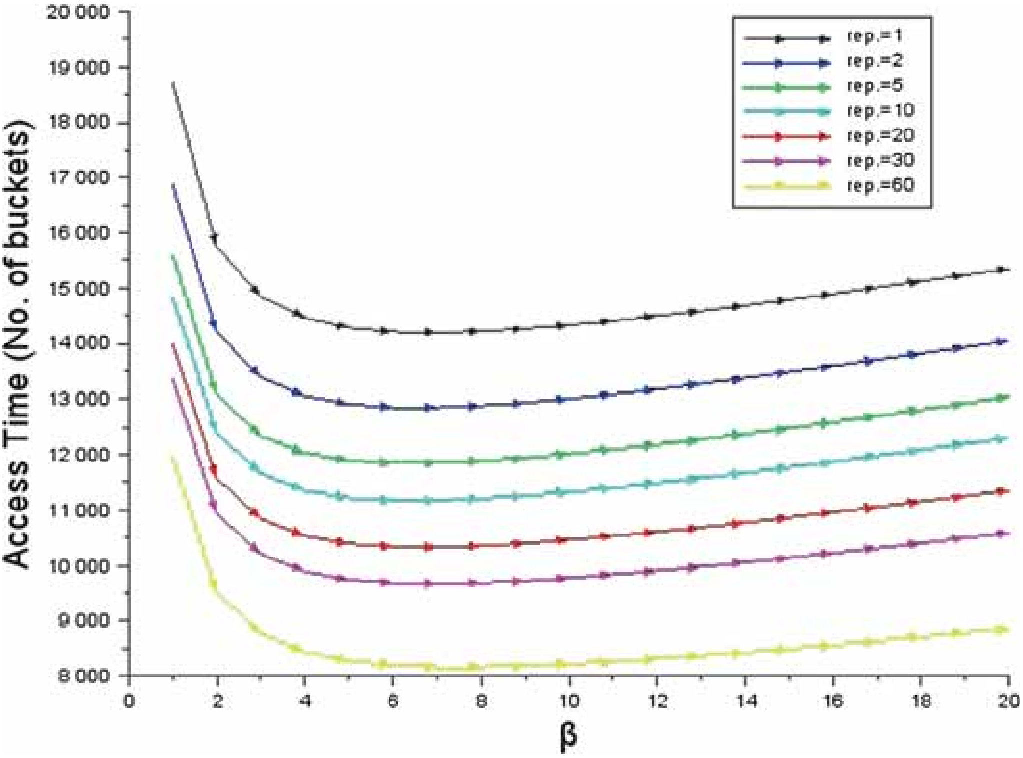
assumed, i.e. there is no replication in any of the indexing techniques. Various indexing schemes may be represented as (1,0), (1,1), (1,1 (1,1)) and proposed indexing scheme: partial index replicated and distributed scheme with (1,1 (1,1)). (1,0) means no index in a bcast. (1,1) and (1,1(1,1)) means one tree and one inverted list without replication in a bcast.

The average access time and the optimum values of α and β of the proposed indexing scheme are evaluated. The values of α and β show replicated levels. The average access time, performance metric depends on replicated values of B+ tree and inverted list as α and β respectively. In this experiment, the average access time of the proposed indexing scheme is evaluated by varying the values of α and β. It may be observed by graph 1 that at $\alpha = 3$ and $\beta = 2$ values, the average access time of the proposed scheme is least. So, $\alpha = 3$ and $\beta = 2$ are the optimal values.

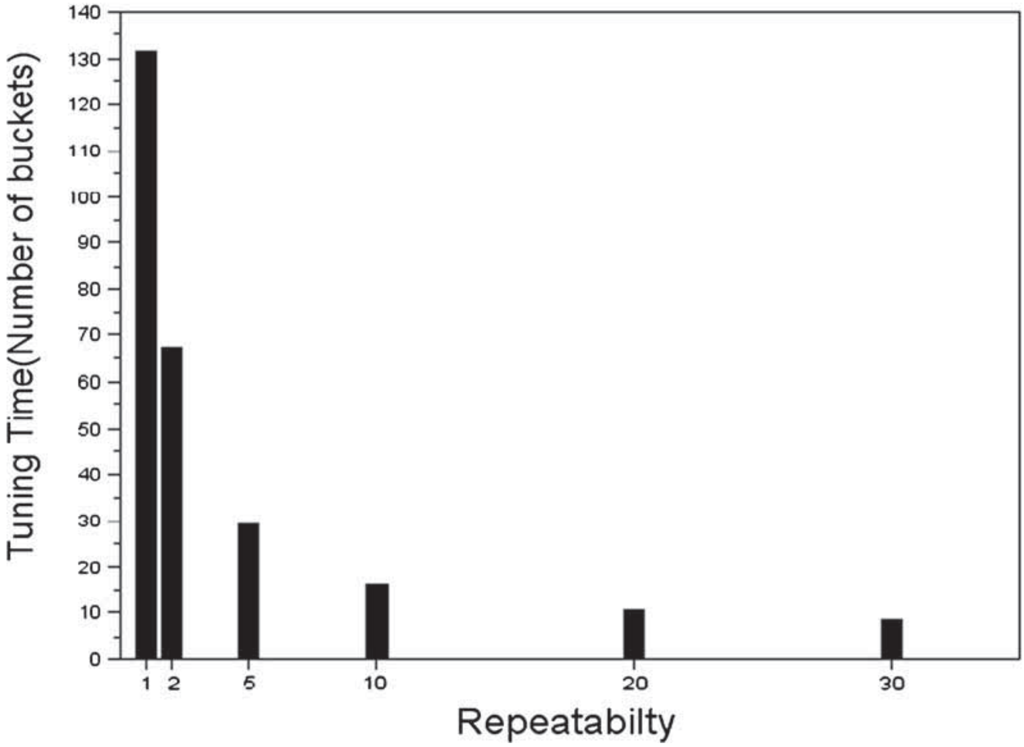
Next, the average access time and the average tuning time of the proposed indexing scheme are further evaluated by the varying repeatability of words. If the repeatability of words increases in a document, the number of distinct words decreases. The less number of distinct words reduces the selectivity as the document may not contain each word. Further, the low selectivity reduces the size of inverted list. Now, the inverted list contains less number of document pointers. The access time of the proposed indexing scheme depends upon the size of the B+ tree and the inverted list. Access time is further reduced because of less size of the inverted list.

Graph 2 depicts decrement of access time as repeatability increases by varying values β in the proposed indexing scheme. The access time decreases up to a certain value of β (data replication value). After that value it increases again. That value is the optimum value of data replication. It may be concluded from graph 2 and graph 3 that the number of distinct documents and the selectivity are inversely proportional to the repeatability. This reduces the access time of the proposed indexing too.

Graph 3 depicts decrement of tuning time as the repeatability increases in the proposed indexing scheme. It may be concluded from graph 2 that the number of distinct documents and selectivity are inversely proportional to the repeatability. This reduces tuning time of the proposed indexing too.



Graph 2. Access time performance of proposed partial index replicated and distributed scheme.



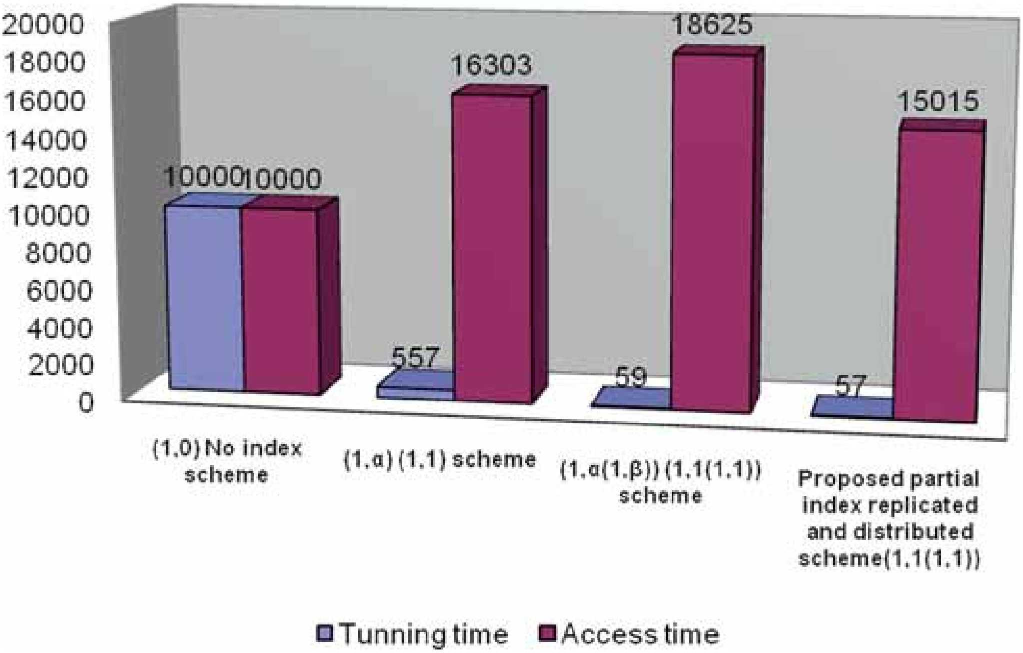
Graph 3. Tuning time performance of proposed partial index replicated and distributed scheme.

7. Result analysis

As discussed in earlier sections, the existing indexing schemes are no replication, $(1, \alpha)$ and $(1, \alpha(1, \beta))$. A significant improvement over the results in terms of access time and tuning time is evaluated, compared and depicted in graph 4. The analysis of tuning time and access time is performed for no replication, $(1, \alpha)$ and $(1, \alpha(1, \beta))$ indexing schemes and proposed indexing scheme through graph 4. The results depict that the average access time is significantly decreased in the proposed indexing scheme as compared to the existing indexing schemes. A decrease in the average access time is due to the partial replication of index in the index tree (B+ tree). This reduces the overall index length in a bcast. Due to a partial replication of the index tree, data overhead is reduced in the proposed indexing scheme.

Initially, the tuning time depends on the height of the index tree. An initial probe is to determine the occurrence of a control index. A second probe is to access a control index. A control index directs the client to higher levels of the index. Next, the control index directs the client to the inverted list index. Last, the required data is downloaded by the client. As depicted in graph 4, the average access time and the average tuning time are further reduced in the proposed indexing scheme compared to no replication, $(1, \alpha)$ and $(1, \alpha(1, \beta))$ schemes.

The values of evaluating parameters: the average access time and the average tuning time of proposed and existing indexing schemes are evaluated by varying the number of buckets. These values are tabulated in table 4 and depicted in graph 4.



Graph 4. Access time and tuning time performance of the proposed partial index replicated and distributed scheme.

Table 4. Evaluating parameters with values for indexing schemes.

Parameters\ schemes	(1,0) no replication scheme	$(1, \alpha)$ Scheme $\alpha = 1$	$(1, \alpha(1, \beta))$ Scheme $\alpha = 1$ and $\beta = 1$	Proposed partial index replicated and distributed Scheme $\alpha = 1$ and $\beta = 1$
Average access time	10,000	16,303	18,625	15,015
Average tuning time	10,000	557	59	57

8. Conclusion

Being energy efficient, the data indexing is an extremely useful solution for broadcasting on wireless channel. A full text search on wireless broadcast stream is another popular type of information access. In this paper, a full text search information indexing scheme is discussed, that uses two level structures: B+-tree and inverted list. The existing full text search indexing scheme is extended by proposing a new partial index replicated and distributed scheme. The index tree is divided into two parts: the upper replicated part and the lower non-replicated part. The replication of B+ tree may be at a certain fixed level to index the information. The access time and the tuning time of the proposed indexing scheme is evaluated, analyzed and compared with existing indexing schemes. The findings confirm that the average access time and the average tuning time are reduced significantly in confirmation to the research scheme. As per calculation, there are optimum values of replication i.e. values of α and β show minimal access time and tuning time.

The average access time and the average tuning time of the proposed indexing scheme are evaluated by the varying repeatability of words. As the repeatability of words increases, the number of distinct words decreases in a document. The less number of distinct words reduces the selectivity of the words because a document may not contain each word. It may also be concluded that the number of distinct documents and the selectivity are inversely proportional to the repeatability. It is found that our proposed indexing scheme is the most power efficient and fast data access indexing scheme for full-text search over wireless channels.

References

Acharya S, Franklin M, Zdonik S and Alongso R 1995 Broadcast disks: Data management for asymmetric communications Environments. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data* 199–210

Barbara D 1999 Mobile computing and database – A survey. *IEEE Trans. Knowl. Data Eng.* 11(1): 108–117

Chung Y D, Yoo S and Kim M H 2010 Energy and latency efficient processing of full text searches on a wireless broadcast stream. *IEEE Trans. Knowl. Data Eng.* 22(2): 207–218

Faloutsos C and Christodoulakis S 1984 Signature files: An access method for documents and its analytical performance evaluation. *ACM Trans. Office Inf. Syst.* 2(4): 267–288

Imielinski T, Viswanathan S and Badrinath B R 1994 Energy efficient indexing on air. In: *Proceedings of SIGMOD* 25–36

Imielinski T, Viswanathan S and Badrinath B R 1997 Data on air: Organization and access. *IEEE Trans. Knowl. Data Eng.* 9(3): 353–372

Mahapatra A K and Biswas S 2011 Inverted indexes: Types and techniques. *Int. J. Comput. Sci. Issues (IJCSI)* 8(4): 384–292

Scholer F, Williams H E, Yiannis J and Zobel J 2002 Compression of inverted indexes for fast query evaluation. *SIGIR* pp. 222–229

Seifert A and Hung J J 2006 FlexInd: A flexible and parameterizable air indexing scheme for data broadcast systems. In: *Proceedings of the 10th International Conference on Extending Database Technology*, Lecture notes in computer science, 3896: 902–920

Tomasic A, Garcıa-Molina H and Shoens K 1994 Incremental updates of inverted lists for text document retrieval. *SIGMOD* 23(2): 289–300

Xu J, Lee W C, Tang X, Gao Q and Li S 2006 An error-resilient and tunable distributed indexing scheme for wireless data Broadcast. *IEEE Trans. Knowl. Data Eng.* 18(3): 392–404

Yang X and Bouguettaya A 2005 Adaptive data access in broadcast-based wireless environments. *IEEE Trans. Knowl. Data Eng.* 17(3): 326–338

Yang K, Shi Y, Wu W, Gao X and Zhong J 2011 A novel hash-based streaming scheme for energy efficient full-text search in wireless data broadcast. DASFAA; Part-1. *Lecture notes in computer science* 6587: 372–388

Yao Y, Tang X, Lim E P and Sun A 2006 An energy-efficient and access latency optimized indexing scheme for wireless data broadcast. *IEEE Trans. Knowl. Data Eng.* 18(1): 1111–1124

Zhang J and Suel T 2007 Optimized inverted list assignment in distributed search engine architectures. In: *Parallel and distributed processing symposium* 41

Zhong J, Wu W, Shi Y and Gao X 2011 Energy-efficient tree-based indexing schemes for information retrieval in wireless data broadcast. DASFAA, Part-II, *Lecture notes in computer science* 6588, Springer-Verlag Berlin Heidalberg, 335–351

Zhong J, Gao Z, Wu W, Chen W, Gao X and Yue X 2013a High performance energy efficient multi-channel wireless data broadcasting system. 1–6

Zhong J, Wu W, Gao X, Shi Y and Yue X 2013b *Evaluation and comparison of various indexing schemes in single-channel broadcast communication environment*. Springer-Verlag, London, 1–35

Zobel J and Moffat A 2006 Inverted files for text search engines. *ACM Comp. Survey* 38(2): 6-es